SSW11

The 11th ISCA Speech Synthesis Workshop

Budapest, Hungary

SSW11 Proceedings
ISBN 978-615-01-2108-6

SSW11 Sponsors

Platinum Sponsor

Google

Gold Sponsors

科大讯飞 iFLYTEK

SAMSUNG

The International Speech Communication Association (ISCA)

MTA

HTEnet Innovációs Nonprofit Kft.

MŰEGYETEM 1782
BME VIK

Dear Participants,

At the time of submitting our bid for SSW11, I hoped that we can re-create the atmosphere of SSW1 in Autrans that was the first international speech conference I could attend in 1990. I enjoyed the single-track approach that allowed everyone to listen to all the presentations. At the age of 31 years, coming from a country where an ordinary citizen until then only once in 3 years was allowed to travel to Western Europe, that conference was a sort of "open door" to the speech synthesis research community. We could have a chat anytime with world-famous researchers (e.g. Ken Stevens). That was when we could start a friendship with one of the organizers, Christian Benoit, whose name is still well-known although he passed away 23 years ago. Fortunately, the other organizer of SSW1 – Gérard Bailly- will be with us on-site. That is one of the reasons why I encourage everyone to come to Budapest in person.

Another important reason is that Hungary was among the first TTS developers of the world. The HungaroVox formant-based TTS system was demonstrated in 1981 by the Hungarian Academy of Sciences. One of the HungaroVox developers - Gábor Olaszy - will be also on-site with us.

Unfortunately, due to the COVID pandemic, our original plans could not be fulfilled. Initially, we wanted to have everyone in the same hotel (just as it was at SSW1 in Autrans) by Lake Velence. Due to travel restrictions and uncertainty, we moved the conference to a hybrid setting, and the location is a wellness hotel in Budapest which is more flexible in terms of lecture room size and reservations. Although Hungary and Budapest are some of the safest places worldwide, it seems that most participants could only afford remote registration. Even with these limitations, we try to provide as much interaction possibility during the conference as possible.

All submitted papers were reviewed by three members of the Scientific Committee. The accepted 40 papers will be presented in oral sessions, where 24 minutes are devoted to the introduction, the talk, and Q&A activity. On each day, we start with an oral session. One-hour keynotes are planned for the middle of the day in Europe so that the audience spanning from Vancouver to Tokyo had a chance to be fully aware. Discussions may be mixed with lunch after the keynotes. In selecting the keynotes, we tried to follow the special topic proposed for SSW11: "*Speech uniqueness and deep learning*". The first plenary speaker is Lior Wolf, who will present the crossroads of speech, singing, and music. Our second keynote speaker, István Winkler, will introduce us into his basic research on the development of the communication

of infants. On the last day, Thomas Drugman will present the latest results on expressive TTS, which is an essential requirement for more extended human-machine speech dialogues.

Authors are encouraged to submit an extended version of the papers to a special issue on speech synthesis of the Infocommunications Journal (www.infocommunications.hu).

Please use the breaks and the session discussions for both private and group forms of exchanging ideas and opinions. The platform of the presentations will be Zoom Webinar. To facilitate discussions, we shall use the Spatial Chat infrastructure.

Those who come in person have the advantage of personal participation at the welcome reception and the social event besides the regular program.

I would like to thank members of the Organizing Committee for all the effort that made SSW11 possible. Both the past and current Synsig Board members have helped a lot during the preparation. Péter Nagy and Mária Tézsla from the Scientific Association for Infocommunications have taken up the burden of organizational and financial administration. Scientific Committee members spent several hours on thoroughly evaluating the papers and give helpful feedback to the authors.

Our sponsors, Google, Apple, Samsung, iFlytek, ISCA, and the Hungarian Academy of Sciences, allowed us to keep registration fees low while providing high-quality services to the audience. I hope that against all the difficulties caused by the COVID pandemic, SSW11 will provide a remarkable contribution to the development of speech synthesis. Although smaller in numbers, but hopefully through intensive interaction with each other and the two keynote speakers who will be in Budapest, on-site participants will have a unique chance. On behalf of our Speech Communication and Smart Interaction Labs of the Budapest University of Technology and Economics, you are all welcome anytime when you visit Budapest.

I hope that one of the young researchers participating at SSW11 will chair SSW26 30 years from now. That would be a remarkable result.

Looking forward to meeting you at SSW11.

Budapest, August 10, 2021.

Géza Németh, Chairman

Organizing committee

Géza Németh	BME TMIT, Hungary
Junichi Yamagishi	National Institute of Informatics Japan, University of Edinburgh, UK
Sébastien Le Maguer	ADAPT Centre/TCD, Ireland
Esther Klabbers	Readspeaker, Netherlands
Mátyás Bartalis	BME TMIT, Hungary
Tamás Gábor Csapó	BME TMIT, Hungary
Bálint Gyires-Tóth	BME TMIT, Hungary
Gábor Olaszy	BME TMIT, Hungary
Csaba Zainkó	BME TMIT, Hungary

Abstract book design

Csaba Zainkó	Budapest University of Technology and Economic
Sébastien Le Maguer	Trinity College Dublin / Adapt Centre

Scientific Committee

Nagaraj Adiga	University of Crete
Gerard Bailly	GIPSA-Lab
Pallavi Baljekar	Google
Timo Baumann	University of Hamburg
Antonio Bonafonte	Universitat Politècnica de Catalunya
Joao Cabral	Trinity College Dublin
Robert Clark	Google
Erica Cooper	National Institute of Informatics
Tamás Gabor Csapo	Budapest University of Technology and Economic
Daniel Erro	Cirrus Logic
Raul Fernandez	IBM
Philip N. Garner	Idiap Research Institute
Balint Gyires-Toth	Budapest University of Technology and Economic
Qiong Hu	Google
Esther Klabbers	ReadSpeaker
Zhen-Hua Ling	University of Science and Technology of China
Sébastien Le Maguer	Trinity College Dublin / Adapt Centre
Jindrich Matousek	University of West Bohemia
Thomas Merritt	Amazon
Bernd Möbius	Saarland University
Eva Navas	University of the Basque Country
Géza Németh	Budapest University of Technology and Economic
Yamato Ohtani	AI Inc.,
Michael Pucher	Acoustics Research Institute
Francesc Alias Pujol	La Salle - Universitat Ramon Llull
Tuomo Raitio	Apple Inc
Manuel Sam Ribeiro	The University of Edinburgh
Andrew Rosenberg	Google
Adriana Stan	Technical University of Cluj-Napoca
Eva Szekely	KTH Royal Institute of Technology
Tomoki Toda	Nagoya University
Markus Toman	Neuratec
Jaime Lorenzo Trueba	Universidad Politecnica de Madrid
Pirros Tsiakoulis	Samsung Electronics
Junichi Yamagishi	The University of Edinburgh
Csaba Zainkó	Budapest University of Technology and Economic
Heiga Zen	Google

Lior Wolf, Facebook AI Research and Tel Aviv University, Israel

Deep Audio Conversion Technologies and Their Applications in Speech, Singing, and Music

Lior Wolf is a research scientist at Facebook AI Research and a full professor in the School of Computer Science at Tel-Aviv University, Israel. He conducted postdoctoral research at prof. Poggio's lab at the Massachusetts Institute of Technology and received his PhD degree from the Hebrew University, under the supervision of Prof. Shashua. He is an ERC grantee and has won the ICCV 2001 and ICCV 2019 honorable mention, and the best paper awards at ECCV 2000 and ICANN 2016. His research focuses on computer vision, audio synthesis, and deep learning.

István Winkler, Research Centre for Natural Sciences, Hungary
Early Development of Infantile Communication by Sound

István Winkler, PhD, DSc, electrical engineer, psychologist. He received his PhD in 1993 at the University of Helsinki, studying auditory sensory memory by electroencephalographic measures. He defended his Doctor of Science thesis in 2005 at the Hungarian Academy of Sciences on auditory deviance detection. His current fields of interest are predictive processing in the auditory deviance detection, auditory scene analysis, communication by sound, and the development of these functions in infancy. During his career, he has authored/coauthored over 250 publications, which received over 11000 references. Currently he is the director of the Institute of Cognitive Neuroscience and Psychology, Research Centre for Natural Sciences, Budapest, Hungary and the head of the Sound and Speech Perception research group (http://www.ttk.hu/kpi/en/sound-and-speech-perception/).

Thomas Drugman, Amazon, Germany
Expressive Neural TTS

Thomas Drugman is a Science Manager in Amazon TTS Research team. He received his PhD in 2011 from the University of Mons, winning the IBM Belgium award for “Best Thesis in Computer Science”. His PhD thesis studied the use of glottal source analysis in Speech Processing. He then made a 3-year post-doc on speech/audio analysis for two biomedical applications: trachea-esophageal speech reconstruction and cough detection in chronic respiratory diseases. In 2014, he joined Amazon as a Scientist in the Alexa ASR team. He then transferred to the TTS team in 2016, where he is Science Manager since 2017. He has contributed in making Amazon’s Neural TTS more natural and expressive, notably by enriching Alexa’s experience with different speaking styles: emotions, newscaster, whispering, etc. His current research interests lie in improving the naturalness and flow of longer synthetic speech interactions. He has about 125 publications in the field of Speech Processing. He got the Interspeech Best Student Paper awards in 2009 and 2014 (as supervisor). He is also member of the IEEE Speech and Language Technical Committee since 2019.

PROGRAM

Thursday, August 26, 2021

SSW Opening

08:30	Welcome	

Session 1: Special synthesis problems

09:00	Sai Sirisha Rallabandi, Babak Naderi & Sebastian Möller: *Identifying the vocal cues of likeability, friendliness and skilfulness in synthetic speech*	1
	Tamás Gábor Csapó: *Extending Text-to-Speech Synthesis with Articulatory Movement Prediction using Ultrasound Tongue Imaging*	7
	Martin Lenglet, Olivier Perrotin & Gérard Bailly: *Impact of Segmentation and Annotation in French end-to-end Synthesis*	13
	Marc Illa, Bence Mark Halpern, Rob van Son, Laureano Moro-Velazquez & Odette Scharenborg: *Pathological voice adaptation with autoencoder-based voice conversion*	19
	Elijah Gutierrez, Pilar Oplustil-Gallegos & Catherine Lai: *Location, Location: Enhancing the Evaluation of Text-to-Speech synthesis using the Rapid Prosody Transcription Paradigm*	25
11:00	Coffee break	

Keynote 1: Expressive Neural TTS

11:10	Lior Wolf: *Deep Audio Conversion Technologies and Their Applications in Speech, Singing, and Music*	

Friday, August 27, 2021

Session 4: Articulation and Naturalness

Keynote 2: Early Development of Infantile Communication by Sound

Session 5: Emotion, singing and voice conversion

17:20 Afternoon Session discussion

Saturday, August 28, 2021

Session 7: Modeling and evaluation

Keynote 3: Deep Audio Conversion Technologies and Their Applications in Speech, Singing, and Music

Session 8: Synthesis and Context

SSW closing

15:10 Closing & SynSIG announcement

11th ISCA Speech Synthesis Workshop (SSW 11)
26-28 August 2021, Budapest, Hungary

Identifying the vocal cues of likeability, friendliness and skilfulness in synthetic speech

Sai Sirisha Rallabandi[1], *Babak Naderi* [1] *and Sebastian Möller*[1,2]

[1]Quality and Usability Lab, Technische Universität Berlin, Germany,
[2]Speech and Language Technology, Deutsches Forschungszentrum für Künstliche Intelligenz (DFKI), Berlin, Germany

{s.rallabandi, babak.naderi, sebastian.moeller}@tu-berlin.de

Abstract

The advent of neural Text-to-Speech (TTS) synthesizers has enhanced the expressivity of synthetic speech in the recent past. However, there is very little work on understanding the acoustic correlates of paralinguistic traits, emotions, speaker attributes and characteristics from synthetic speech. This paper investigates the acoustic correlates of the speaker attributes: likeability, friendliness, and skillfulness. Our study was carried out on the voices derived from the two commercial TTS systems, Amazon Polly (9 voices) and Google TTS engine (10 voices). In our previous study, we performed a crowd-sourcing-based evaluation to collect the subjective ratings for the desired speaker attributes. In this work, we perform the acoustic feature prediction using the backward elimination algorithm. We show that the level of loudness, spectral flux, fundamental frequency, its formant frequencies, and their combinations contribute to the desired speaker attributes. We further combine the ratings of friendliness and likeability to investigate the characteristic, warmth in synthetic speech and correspondingly, skilfullness represents the characteristic, competence.

Index Terms: Synthetic speech, acoustic correlates, linear regression, likeable, friendly, skillful

1. Introduction

Artificial speech generation is predominant through its applications such as navigation [1], language learning systems [2], customer service [3], personal assistants [4] and many more. The neural speech synthesizers have facilitated the expressivity in the generated speech in the recent past [5, 6, 7, 8, 9, 10]. The fidelity of these TTS systems can be further enhanced through the generation of paralinguistic traits, various emotions, and social speaker characteristics. In order to achieve this, there is a need for the investigation of the acoustic correlates of various speaker attributes in synthetic speech. In this paper, we identify the vocal cues responsible for the social speaker characteristics, warmth and competence in synthetic voices.

Perception of a person from their behavior has been researched extensively in 1940s and 50s [11, 12]. Various studies were carried out similar to the BIG FIVE personality traits to categorise and understand the human behavior [13, 14, 15, 16]. In [12], the sociology researchers stated that the perception of a person is done based on two criteria: social norms (warmth) and task accomplishment (competence). In [17] psychology researchers state that the characteristics, warmth and competence can be termed as the universal dimensions of social perception. This is because they include both interpersonal relationships and the social behavior of a person. The attributes that describe the characteristics, warmth and competence in humans were: likeability, friendliness, and skilfulness respectively [18]. Following [18], in our current work, we utilise these 3 dimensions (likeability, friendliness and skilfulness) to interpret warmth and competence in synthetic voices. Similar to BIG FIVE [13, 14] and [18, 19], we conducted a subjective evaluation with two commercial TTS systems (Google TTS engine, Amazon Polly) in [20]. The evaluation was carried out with 15 different adjectives describing various speaker attributes of TTS voices. As an extension, in the current work, we are interested in identifying the acoustic correlates of the speaker attributes that contribute to the social characteristics, warmth and competence in synthetic speech. Inspired by [18] we utilise a subset of the subjective ratings (likeability, friendliness, skilfulness) collected in [20].

The acoustic correlates of various emotions, moods, attitudes, personality traits have been researched previously in both natural and synthetic voices [21, 22, 22, 23, 24, 25]. Speech rate, intensity, speech pauses, pitch, duration and their combinations were commonly identified as the vocal cues responsible for various emotions, personality traits and speaker characteristics [21, 24, 25, 26, 27, 28]. Literature suggests that the acoustic correlates of various emotions and expressions can be divided into 3 categories: voice quality, timing and pitch parameters [21, 22, 28].

To the best of our knowledge, this is the first attempt to analyse the vocal cues of speaker attributes, friendliness, likeability and skilfulness in order to understand the social speaker characteristics, warmth and competence from synthetic speech. Inspired by [25], we perform an acoustic feature prediction over the OpenSMILE features [29] extracted for the synthetic voices. Through our work, we present that the spectral flux, formants (F1, F2, F3), slope of the voiced segment are responsible for warmth in female voices. While, first and second formants (F1, F2), slope of Unvoiced segment, and loudness contribute to warmth in male voices. Competence in female voices is perceived through slope of voiced segment, spectral flux and mfcc. While, the competence in male voices is due to fundamental frequency (F0) and voiced segment length. Later, we perform an automatic prediction of warmth and competence using linear regressor and Support Vector Regressor (SVR).

This paper is organised as follows: In Section 2, we describe the evaluation setup followed by the system performance in Section 3. In Section 4 we present the acoustic feature prediction. Automatic prediction of warmth and competence is presented in Section 5 followed by a discussion in Section 6.

Throughout this work, we will be using the terms voices/speakers/systems to refer the TTS voices and participants/raters/listeners for the subjects who participated in the evaluation. The terms items/attributes/adjectives/questionnaire refer to the questions we used in the subjective evaluation. We

10.21437/SSW.2021-1

use the terms dimensions/speaker attributes to refer likeability/friendliness/skilfulness.

2. Evaluation setup

2.1. Speech data preparation

The commercial TTS systems, Amazon Polly [1] (Neural) and Google TTS engine [2] (Wavenet) have been explored for the study of speaker attributes. There were 4 male and 5 female voices from Amazon Polly and 5 male, 5 female from Google TTS engine. In total, we had 19 different US native speakers' voices. The speech samples generated for each of these voices were from the Harvard database [3]. The number of sentences generated were 32. We have combined the individual speech samples and created speech segments each of length 20 seconds (approx.). Finally, there were 4 speech segments for each of the TTS voices (4 speech segments * 19 voices).

2.2. Subjective evaluation

The social perceptions of synthetic voices were studied in [20]. In [20], we performed a 15-item semantic differential scaling test in a crowd-sourcing subjective test setup. The 15 items were: relaxed, confident, enthusiastic, energetic, friendly, arrogant, pleasant, likeable, responsible, reliable, accessible, sympathetic, skilful, kind, extrovert. For our test, we have included the speaker attribution task in the P.808 toolkit [30]. We have utilised the continuous 100-point scale with the adjective-antonym pairs at its extreme ends. We used The Fragebogen [31] implementation for presenting the questionnaire during the subjective tests. The test was conducted on Amazon Mechanical Turk (AMT). We have included the eligibility and the environment suitability tests in out evaluation setup. We have recruited only US native speakers for the task. The participants were instructed to use headphones throughout the test without fail. The following are the instructions provided to the participants during the test.

For each question, please listen to the audio sample and give your opinion about the voice you hear on the following scales. You will find the positive adjective at the extreme left and a negative adjective at the extreme right of the scale. You can listen to each audio sample multiple times during the test. There is no right or wrong answer as long as you listen to the audio files and give your opinion.

In each session, participants were provided with 4 speech clips and 15 attributes. Additionally, we have repeated one attribute randomly for every question in a session. We have used this repeated attribute as the hidden quality control mechanism [32, 33]. Each speech clip played in a loop until the participant rated all the adjectives.

2.3. Data processing

We performed a pre-processing of the subjective data to remove the participants whose ratings were not reliable. Based on the environment suitability tests, we have rejected 41 responses. Later on, we calculated the Pearson correlation coefficient for the repeated attributes in the test and the original attribute. The ratings were rejected if the correlation coefficient was below 0.5 (5 participants were removed). In total, we have accepted 90% of the subjective data. The number of participants that were retained after the pre-processing were 43 female and 44 male (87 participants). Their ages ranged between 19 and 77 (mean = 40.31 and std = 12.57). On the retained subjective data, we have calculated the intraclass correlation coefficient ICC(1,k) for inter-rater reliability. The average raters absolute value was 0.974 with a 95% confidence interval in the range of 0.95 to 0.99. For our current study, we have utilised the subjective ratings of the scales, friendliness, likeability and skilfulness.

[1] https://aws.amazon.com/polly/

[2] https://cloud.google.com/text-to-speech/

[3] https://www.cs.columbia.edu/ hgs/audio/harvard.html

3. TTS performance

Figures 1, 2, 3, 4 display the perceived speaker attributes: friendliness, likeability and skilfulness in both the TTS systems. We have calculated the mean of the subjective ratings for each of these speaker attributes.

*3.0.1. **Google's female voices***

Figure 1 displays the mean subjective ratings calculated over the three desired attributes for Google's female voices along with the 95% confidence intervals. Among the Google's female voices, H displays lowest mean ratings on friendliness (37), likeability (37.6) and skilfulness (32.13). Speaker E displays highest rating on friendliness (59.9) and likeability (54.3). Speaker C displays the highest rating on skilfulness (41.64).

*3.0.2. **Google's male voices***

Figure 2 displays the mean subjective ratings calculated over the three desired attributes for Google's male voices along with the 95% confidence intervals. Among the Google's male voices, J displays lowest mean ratings on friendliness (31), likeability (29.25) and skilfulness (25.53). Speaker B displays highest rating on friendliness (47) and likeability (46.89). Speaker A displays the highest rating on skilfulness (35.66).

*3.0.3. **Amazon Polly's female voices***

Figure 3 displays the mean subjective ratings calculated over the three desired attributes for Amazon Polly's female voices along with the 95% confidence intervals. Among the Amazon Polly's female voices, Ivy displays lowest mean ratings on friendliness (36.9). Joanna display lowest ratings on likeability (35.93) and skilfulness (31.17). Speaker Kendra displays highest rating on friendliness (54.28) and likeability (51.39). Speaker Ivy displays the highest rating on skilfulness (45.6).

*3.0.4. **Amazon Polly's male voices***

Figure 4 displays the mean subjective ratings calculated over the three desired attributes for Amazon Polly's male voices along with the 95% confidence intervals. Among the Amazon Polly's male voices, Justin displays lowest mean ratings on friendliness (30.6), likeability (32.53). Matthew displays the lowest mean ratings on skilfulness (31.6). Speaker Joey displays highest rating on friendliness (46.2) and likeability (48.03). Speaker Kevin displays the highest rating on skilfulness (45.03).

4. Prediction of acoustic correlates

In order to predict the acoustic correlates of the desired characteristics, we initially downsample the speech segments to 16 kHz and derive the OpenSMILE [29] features. We have employed the Geneva Minimalistic Acoustic Parameter Set (eGeMAPS) configuration [34], since this was built to capture affective speaker characteristics. We have therefore derived

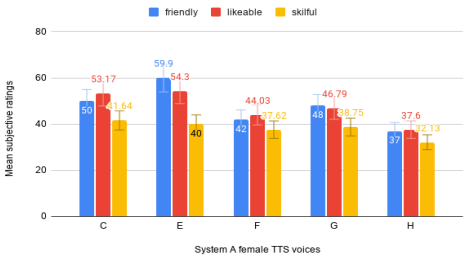


Figure 1: *Mean subjective ratings calculated over friendliness, likeability and skilfulness for Google's female voices.*

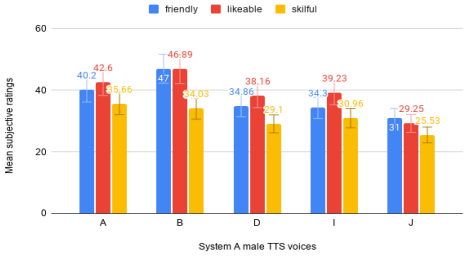


Figure 2: *Mean subjective ratings calculated over friendliness, likeability and skilfulness for Google's male voices.*

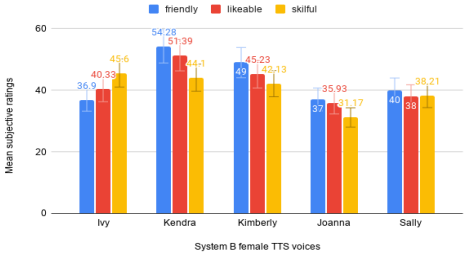


Figure 3: *Mean subjective ratings calculated over friendliness, likeability and skilfulness for Amazon Polly's female voices.*

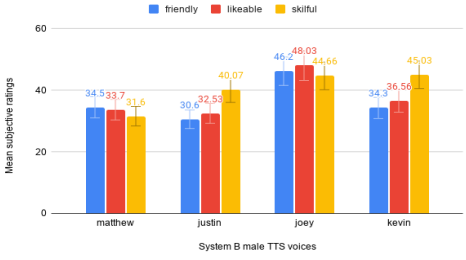


Figure 4: *Mean subjective ratings calculated over friendliness, likeability and skilfulness for Amazon Polly's male voices.*

88 acoustic features corresponding to each speech segment of the TTS voices. Later, we have employed a linear regression based backward elimination algorithm for each of the speaker attributes, friendliness, likeability and skilfulness. Tables, 1, 2, 3 display the derived acoustic features (for each of friendliness/likeability/skilfulness respectively), their corresponding coefficients along with their variances for female voices. We derived these vocal cues using the linear regression on 4 speech segments per each female voice (4 speech segments * 5 Google female voices + 4 speech segments * 5 Amazon Polly female voices). Tables, 4, 5, 6 display the derived acoustic features (for each of friendliness/likeability/skilfulness respectively), their corresponding coefficients along with their variances for male voices. We derived these vocal cues using the linear regression on 4 speech segments per each male voice (4 speech segments * 5 Google male voices + 4 speech segments * 4 Amazon Polly male voices).

Here, the derived acoustic features are the independent variables, the speaker attributes: friendliness, likeability and skilfulness are the dependent variables. A positive coefficient indicates that for a 1-unit change in the acoustic feature (independent variable), there will be an increase in the perception of the speaker attribute (friendliness/likeability/skilfulness) from that voice (increase in the mean of the dependent variable by that coefficient value) and vice versa.

4.1. Friendliness in female voices

Table 1 displays the acoustic correlates of friendliness in female voices (Google and Amazon Polly female voices). We observed that the attribute, friendliness was dependent on the acoustic features, spectral flux, and formants F1, F2 and F3 in female voices. We have also presented the explained variance (R squared = 0.829) obtained during the acoustic feature prediction. We observed that with the change in the spectral flux and the second formant (F2), the friendliness in female voices decreases by the value of 8.7546 and 0.0052 respectively. Accordingly, with the change in the first (F1) and third formants (F3) the friendliness in female voices increase by 0.0174 and 0.0037 respectively.

Table 1: *Acoustic correlates of friendliness in Amazon and Google female voices. The explained variance for female friendliness is 82.9%.*

Acoustic features	Coefficients
Spectral Flux	-8.7546
F1 mean	0.0174
F2 mean	-0.0052
F3 mean	0.0037

4.2. Likeability in female voices

Table 2 displays the acoustic correlates of likeability of female voices (Google and Amazon Polly female voices). We observed that the attribute, likeability was dependent on the acoustic features, spectral flux, formants F1, F2 and Voiced segment Slope (500-1500) in female voices. We have also presented the explained variance (R squared = 0.812) obtained during the acoustic feature prediction. We observed that with the change in the spectral flux, the second formant (F2), and the slope the likeability in female voices decreases by the value of 9.0631, 0.0086 and 57.3852 respectively. Accordingly, with the change in the

first (F1), the likeability of female voices increase by a factor of 0.0241.

Table 2: *Acoustic correlates of likeability in Amazon and Google female voices. The explained variance for female likeability is 81.2%.*

Acoustic features	Coefficients
Spectral Flux	-9.0631
F1 mean	0.0241
F2 mean	-0.0086
SlopeV500-1500	-57.3852

4.3. Skilfulness in female voices

Table 3 displays the acoustic correlates of skilfulness in female voices (Google and Amazon Polly female voices). We observed that the attribute, skilful was dependent on the acoustic features, Voiced segment Slope (0-500),spectral flux, voiced segment mfcc3 in female voices. We have also presented the explained variance (R squared = 0.581) obtained during the acoustic feature prediction. We observed that with the change in the spectral flux, slope and Mel Frequency Cepstral Coefficients (mfcc3), the perception of skilfulness in female voices decreases by the value of 6.4559, 0.1868 and 0.2858 respectively.

Table 3: *Acoustic correlates of skilfulness in Amazon and Google female voices. The explained variance for female skilfulness is 58.1%.*

Acoustic features	Coefficients
SlopeV0-500	-0.1868
SpectralFlux	-6.4559
mfcc3V	-0.2858

4.4. Friendliness in male voices

Table 4 displays the acoustic correlates of friendliness in male voices (Google and Amazon Polly male voices). We observed that the attribute, friendliness was dependent on the acoustic features, first formant (F1), Unvoiced segment Slope (500-1500) and loudness in male voices. We have also presented the explained variance (R squared = 0.685) obtained during the acoustic feature prediction. We observed that with the change in the first formant (F1), Voiced segment Slope (500-1500) and loudness the friendliness in male voices decreases by the value of 0.0117, 176.8888 and 1.1870 respectively.

Table 4: *Acoustic correlates of friendliness in Amazon and Google male voices. The explained variance for male friendliness is 68.5%.*

Acoustic features	Coefficients
F1 mean	-0.0117
SlopeUV500-1500	-176.8888
loudness	-1.1870

4.5. Likeability in male voices

Table 5 displays the acoustic correlates of likeability of male voices (Google and Amazon Polly male voices). We observed that the attribute, likeability was dependent on the acoustic features, loudness, loudness rising slope, formant F1, and unvoiced segment Slope (500-1500) in male voices. We have also presented the explained variance (R squared = 0.731) obtained during the acoustic feature prediction. We observed that with the change in the loudness rising slope, first formant (F1), second formant (F2), and unvoiced slope the likeability of male voices decreases by the value of 0.6164, 0.0101 and 169.6958 respectively. Accordingly, with the change in the loudness, the likeability of male voices increase by a factor of 6.7662.

Table 5: *Acoustic correlates of likeability in Amazon and Google male voices. The explained variance for male likeability is 73.1%.*

Acoustic features	Coefficients
loudness	6.7662
loudness rising slope	-0.6164
F1 mean	-0.0101
SlopeUV500-1500	-169.6958

4.6. Skilfulness in male voices

Table 6 displays the acoustic correlates of skilfulness in male voices (Google and Amazon Polly male voices). We observed that the attribute, skilful was dependent on the acoustic features, fundamental frequency (F0) and voiced segment length in male voices. We have also presented the explained variance (R squared = 0.698) obtained during the acoustic feature prediction. We observed that with the change in the fundamental frequency (F0), the perception of skilfulness in male voices decreases by the value of 8.7332. Accordingly, with the change in the voiced segment length, the perception of skilfulness in male voices increase by a factor of 6.1338.

Table 6: *Acoustic correlates of skilfulness in Amazon and Google male voices. The explained variance for male skilfulness is 69.8%.*

Acoustic features	Coefficients
F0 semitone	-8.7332
Voiced Segment Length	6.1338

5. Automatic prediction of warmth and competence

In this section, we present the automatic prediction of warmth and competence using the regression algorithms, linear regressor, and Support Vector Machine (SVM). For prediction of warmth, we have combined the subjective ratings of the scales, friendliness and likeability. For competence, we use the subjective ratings of skilfulness. The number of training examples we had were 40 for female and 36 for male voices. Hence, we perform a Leave-one-speaker-out cross validation. Table 7 shows the results of automatic prediction of warmth and competence. The first block consists of the prediction of warmth in male and female TTS voices. In the second block, we present the prediction performance for the characteristic, competence. The number of input features fed to the model in case of male and female voices and the characteristic predicted is presented. The performance of the models is evaluated with the metric, mean

Table 7: *Results of automatic prediction of warmth and competence from synthetic speech. AFs= number of acoustic features fed to the model, Ch. = characteristic, warmth (W) or competence (C) (2 attributes (likeability, friendliness) representing warmth and 1 attribute, skilfulness representing Competence) , LR = Linear Regression, SVR = Support Vector Regressor, MSE = mean squared error*

Model	Female			Male		
	AFs	Ch	MSE	AFs	Ch	MSE
LR	5	W	0.21	5	W	0.32
SVR	5	W	0.20	5	W	0.33
LR	3	C	0.47	2	C	0.35
SVR	3	C	0.45	2	C	0.34

squared error (MSE). We observe that the MSE score for female warmth is lower compared to that of MSE of male warmth with the same number of input features. In case of competence, eventhough the female input features are more than that of the male input features, the MSE scores of female are much higher than that of the male voices. The MSE scores of male warmth and competence display similar MSE scores with different number of input features.

6. Discussion

Table 8 presents the acoustic correlates of warmth in female voices. The vocal cues responsible for both the speaker attributes, friendliness and likeability are spectral Flux, first and the second formant (F1, F2). Additionally, third formant (F3) contributes to friendliness and Voiced slope contributes to likeability in female voices.

Table 8: *Warmth in female*

Friendliness	**Likeability**
Spectral Flux	*Spectral Flux*
F1 mean	*F1 mean*
F2 mean	*F2 mean*
F3 mean	SlopeV500-1500

Table 9 presents the acoustic correlates of warmth in male voices. The vocal cues responsible for both the speaker attributes, friendliness and likeability are loudness, first formant (F1) and unvoiced slope. Additionally, loudness rising slope contributes to likeability in male voices.

Table 9: *Warmth in male*

Friendliness	**Likeability**
F1 mean	*loudness*
SlopeUV500-1500	*F1 mean*
loudness	loudness rising slope
-	SlopeUV500-1500

Table 10 presents the acoustic correlates of competence in male and female voices. The vocal cues responsible for competence in male voices were fundamental frequency (F0) and voiced segment length. The acoustic correlates of competence in female voices were voiced slope, spectral flux and mfcc.

Table 10: *Competence in female and male voices*

Female	**Male**
Voiced Slope	F0
Spectral Flux	Voiced length
mfcc	-

From our analysis, we observed that the acoustic features intensity/loudness, spectral flux, fundamental frequency and its formants are the common acoustic features in both natural and synthetic voices contributing to various emotions and speaker characteristics [22, 25, 27]. We observe that the acoustic correlates of social speaker characteristics in synthetic speech can also be categorised into vocal quality (spectral parameters), timing (voiced segment length) and pitch (frequency parameters) as in natural speech [21, 22, 28].

The TTS voices, E (Google female voice) and Kendra (Amazon Polly female) display highest warmth among other TTS voices. The voices, Ivy (Amazon Female) and Kevin (Amazon male) display highest competence over the considered TTS voices.

The acoustic correlates predicted for each of the 3 attributes were obtained from the subjective evaluation conducted on a 15-item semantic differential scaling test. The subjective responses when requested for 3 scales (likeability, friendliness and skilfulness) alone might be different. Additionally, the models were trained on the 20 second long speech segments. Thus, we might have averaged the acoustic information present in the speech samples. Analysing the subjective ratings of individual speech samples could be interesting. Also, collection of the subjective ratings for a larger database and also different speech corpora (conversations, news reading, Semantically Unpredictable Sentences) is another future work. In the current work, the input dimensions (88) were higher than that of the number of training examples (40 for female, 36 for male) during automatic feature prediction. We have thus followed a recursive feature elimination approach for acoustic feature prediction. Therefore, as an extension to this work, we would perform an analysis with a larger dataset and unaveraged acoustic information.

7. Acknowledgements

Authors would like to thank Benjamin Weiss for his valuable time and feedback. This work is being supported by the German Research Foundation (DFG), under funding MO 1038/29-1, TU PSP-Element: 1-50001062-01-EF. We also thank all the participants of our subjective tests.

8. References

[1] K. C. Raghavi, S. K. Rallabandi, S. Sitaram, and A. W. Black, "Speech synthesis for mixed-language navigation instructions," in *Proc. INTERSPEECH*, 2017.

[2] Y.-C. Huang and L.-C. Liao, "A study of text-to-speech (tts) in children's english learning," *Teaching English with Technology*, vol. 15, pp. 14–30, 01 2015.

[3] A. Wilkinson, A. Parlikar, S. Sitaram, T. White, A. W. Black, and S. Bazaj, "Open-Source Consumer-Grade Indic Text To Speech," in *Proc. SSW*, 2016.

[4] Yaniv, Leviathan and Yossi, Matias, "Google Duplex: An AI System for Accomplishing Real-World Tasks Over the Phone," in *Google AI Blog*, 2018.

[5] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *arXiv preprint:1609.03499*, 2016.

[6] Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio, and et al., "Tacotron: Towards End-to-End Speech Synthesis," *in Proc. Interspeech*, 2017.

[7] F. Biadsy, R. J. Weiss, P. J. Moreno, D. Kanevsky, and Y. Jia, "Parrotron: An end-to-end speech-to-speech conversion model and its applications to hearing-impaired speech and speech separation," *arXiv preprint:1904.04169*, 2019.

[8] Y. Wang, D. Stanton, Y. Zhang, R. Skerry-Ryan, E. Battenberg, J. Shor, Y. Xiao, F. Ren, Y. Jia, and R. A. Saurous, "Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis," *arXiv preprint:1803.09017*, 2018.

[9] D. Stanton, Y. Wang, and R. Skerry-Ryan, "Predicting expressive speaking style from text in end-to-end speech synthesis," *2018 IEEE Spoken Language Technology Workshop (SLT)*, pp. 595–602, 2018.

[10] Y.-J. Zhang, S. Pan, L. He, and Z. Ling, "Learning latent representations for style control and transfer in end-to-end speech synthesis," *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6945–6949, 2019.

[11] S. E. Asch, "Forming impressions of personality." *The Journal of Abnormal and Social Psychology*, vol. 41, no. 3, p. 258, 1946.

[12] R. F. Bales, "A set of categories for the analysis of small group interaction," *American Sociological Review*, vol. 15, no. 2, pp. 257–263, 1950.

[13] J. S. Wiggins, P. Trapnell, and N. Phillips, "Psychometric and Geometric Characteristics of the Revised Interpersonal Adjective Scales (IAS-R)," *Multivariate Behavioral Research*, 1988.

[14] R. McCrae and O. John, "An Introduction to the Five-Factor Model and its Applications," *Journal of Personality*, 1992.

[15] V. P. Rosenberg S, Nelson C, "A multidimensional approach to the structure of personality impressions," *Journal of Personality and Social Psychology*, 1968.

[16] S. E. Asch, "Forming impressions of personality," *The Journal of Abnormal and Social Psychology*, 1946.

[17] S. T. Fiske, A. J. Cuddy, and P. Glick, "Universal dimensions of social cognition: Warmth and competence," *Trends in cognitive sciences*, vol. 11, no. 2, pp. 77–83, 2007.

[18] S. T. Fiske, "Stereotype Content: Warmth and Competence Endure," *Current Directions in Psychological Science*, 2018.

[19] A. Abele, N. Hauke, K. Peters, E. Louvet, A. Szymkow, and Y. Duan, "Facets of the fundamental content dimensions: Agency with competence and assertiveness—communion with warmth and morality," *Frontiers in Psychology*, vol. 7, 2016.

[20] Sai Sirisha Rallabandi, Abhinav Bharadwaj, Babak Naderi, Sebastian Möller, "Perception of social speaker characteristics in synthetic speech," *in Proc. Interspeech*, 2021.

[21] K. R. Scherer, "Vocal affect expression: A review and a model for future research." *Psychological bulletin*, vol. 99, no. 2, p. 143, 1986.

[22] C. Gobl and A. N. Chasaide, "The role of voice quality in communicating emotion, mood and attitude," *Speech Commun.*, vol. 40, pp. 189–212, 2003.

[23] C. Nass, U. Foehr, and M. Somoza, "The effects of emotion of voice in synthesized and recorded speech," 2001.

[24] J. Cahn, "The generation of affect in synthesized speech," *Journal of the American Voice I/O Society*, vol. 8, pp. 1–19, 1990.

[25] L. F. Gallardo and B. Weiss, "Perceived interpersonal speaker attributes and their acoustic features," *Proc. Phonetik & Phonologie*, 2017.

[26] P. Laukka, P. Juslin, and R. Bresin, "A dimensional approach to vocal expression of emotion," *Cognition and Emotion*, vol. 19(5), pp. 633–653, 08 2005.

[27] M. SCHROEDER, "Speech and emotion research : An overview of research frameworks and a dimensional approach to emotional speech synthesis," *Doctoral thesis, Phonus 7, Research Report of the Institute of Phonetics, Saarland University*, 2004.

[28] I. Murray and J. Arnott, "Toward the simulation of emotion in synthetic speech: a review of the literature on human vocal emotion." *The Journal of the Acoustical Society of America*, 1993.

[29] F. Eyben, F. Weninger, F. Gross, and B. Schuller, "Recent developments in opensmile, the munich open-source multimedia feature extractor," in *Proceedings of the 21st ACM International Conference on Multimedia*, 2013.

[30] B. Naderi and R. Cutler, "An Open source Implementation of ITU-T Recommendation P.808 with Validation," to appear in INTERSPEECH. ISCA, 2020.

[31] D. Guse, H. R. Orefice, G. Reimers, and O. Hohlfeld, "TheFragebogen: A Web Browser-based Questionnaire Framework for Scientific Research," in *2019 Eleventh International Conference on Quality of Multimedia Experience (QoMEX)*. IEEE, 2019, pp. 1–3.

[32] B. Naderi, I. Wechsung, and S. Möller, "Effect of being observed on the reliability of responses in crowdsourcing micro-task platforms," in *2015 Seventh International Workshop on Quality of Multimedia Experience (QoMEX)*. IEEE, 2015, pp. 1–2.

[33] B. Naderi, *Motivation of workers on microtask crowdsourcing platforms*. Springer, 2018.

[34] F. Eyben, K. R. Scherer, B. W. Schuller, J. Sundberg, E. André, C. Busso, L. Y. Devillers, J. Epps, P. Laukka, S. S. Narayanan *et al.*, "The Geneva minimalistic acoustic parameter set (GeMAPS) for voice research and affective computing," *IEEE transactions on affective computing*, 2015.

11th ISCA Speech Synthesis Workshop (SSW 11)
26-28 August 2021, Budapest, Hungary

Extending Text-to-Speech Synthesis with Articulatory Movement Prediction using Ultrasound Tongue Imaging

Tamás Gábor Csapó[1,2]

[1]Department of Telecommunications and Media Informatics,
Budapest University of Technology and Economics, Budapest, Hungary
[2]MTA-ELTE Lendület Lingual Articulation Research Group, Budapest, Hungary
csapot@tmit.bme.hu

Abstract

In this paper, we present our first experiments in text-to-articulation prediction, using ultrasound tongue image targets. We extend a traditional (vocoder-based) DNN-TTS framework with predicting PCA-compressed ultrasound images, of which the continuous tongue motion can be reconstructed in synchrony with synthesized speech. We use the data of eight speakers, train fully connected and recurrent neural networks, and show that FC-DNNs are more suitable for the prediction of sequential data than LSTMs, in case of limited training data. Objective experiments and visualized predictions show that the proposed solution is feasible and the generated ultrasound videos are close to natural tongue movement. Articulatory movement prediction from text input can be useful for audiovisual speech synthesis or computer-assisted pronunciation training.
Index Terms: DNN-TTS, audiovisual synthesis, ultrasound

1. Introduction

Statistical parametric methods are frequently used in text-to-speech (TTS) synthesis, with two main machine learning techniques: hidden Markov-models (HMM, [1]) and deep neural networks (DNN, [2]). Recently, the focus of TTS research has moved to end-to-end solutions, applying neural vocoders (e.g. WaveNet [3] and WaveGlow [4]) and sequence-to-sequence models using attention (e.g. Tacotron2 [5]). Still, traditional (non-end-to-end, vocoder-based) DNN-TTS systems are useful in limited data scenarios, when there is few training data available, for example with speech and biosignal recordings, or in case of audiovisual speech synthesis.

1.1. Audiovisual speech synthesis

Audiovisual speech synthesis is a subfield of the more general areas of speech synthesis and computer facial animation [6]. The goal of the visible speech synthesis is typically to obtain a mask with realistic motions, not to duplicate the musculature of the face to control this mask.

The field of visual speech synthesis is fairly well established and a variety of approaches have been developed (including rule-based [7] and data-driven methods [8]). Rule-based systems include models for speech sequence planning, for muscle mechanisms and for the physical speech production apparatus. Within the biomechanical model of the vocal tract, the tongue can be represented as a finite element mesh [7] and complex biomechanical simulations are necessary to estimate the internal muscle stresses during the movement of human articulators [9]. In the context of data-driven approaches and HMM-based synthesis, there are two main categories: image-based systems are supposed to look like a video of a real person, while motion capture based systems derive features from facial points tracked over time [8]. For HMM-based audiovisual synthesis, a synchronous corpus of parametrized facial motion data and acoustic speech data is necessary. Schabus et al. showed that in combined HMM-based text-to-speech synthesis and facial animation, joint audiovisual models perform better than training separate acoustic and visual models [8].

1.2. Predicting articulatory movement from text

Another type of TTS extension is when the target is to predict articulatory motion (e.g. lip or tongue movement) and not just the face of the speaker, besides the speech output. This requires special biosignals to be recorded, which can track the movement of the articulatory organs (e.g. EMA, X-ray, vocal tract MRI, and ultrasound tongue imaging). With such a system, by giving an arbitrary input text, one is able to hear the speech and, in synchrony with it, see how to move the tongue in 3D to produce target speech sounds. This visual feedback can make a big difference for pronunciation training in L2 learning, especially when the target language contains speech sounds which are difficult to articulate.

Most earlier studies in this context were using point-tracking devices, like electromagnetic articulography (EMA) [10, 11, 12, 13, 14, 15]. Ling and his colleagues proposed a HMM-based text-to-articulatory movement prediction system, i.e. which can synthesize the speaker's mouth from text [10]. Here, the durations were not modeled, but in a subsequent study, they also investigated the timing aspects and analyzed the critical articulators [11]. Wei et al. used DNNs for the text-to-EMA prediction and confirmed that stacked bottleneck features are effective for this purpose [12]. Steiner and his colleagues experimented similarly with text-to-EMA prediction using HMMs (with synchronous text-to-speech), and they also included a geometric 3D tongue model as the target [13]. Next, they compared HMMs and DNNs for the text-to-tongue model prediction [14]. It was found that with less than 2 hours of data, DNNs outperformed HMMs. Yu and her colleagues predicted 3D articulatory movement, from text and audio inputs, therefore combining the text-to-speech and acoustic-to-articulatory inversion fields [15]. For the machine learning approach, they used a bottleneck long-term recurrent convolutional neural network. They showed that the text information complements well the acoustic features during the prediction of EMA-based articulation. The final output of the system is speech synchronized with 3D articulatory animation, using a facial mesh model [15].

As shown above, there have been several studies investigating text-to-articulatory motion with HMMs or DNNs, but all of

10.21437/SSW.2021-2

them are using point-tracking equipment (electromagnetic articulography). Medical imaging target, like ultrasound or MRI, have not been used before in this context.

1.3. Ultrasound tongue imaging

Ultrasound tongue imaging (UTI) is a technique suitable for the acquisition of articulatory data. Phonetic research has employed 2D ultrasound for a number of years for investigating tongue movements during speech [16]. Stone summarized the typical methodology of investigating speech production using ultrasound [17]. Usually, when the subject is speaking, the ultrasound transducer is placed below the chin, resulting in midsagittal images of the tongue movement. The typical result of 2D ultrasound recordings is a series of gray-scale images in which the tongue surface contour has a greater brightness than the surrounding tissue and air. Compared to other articulatory acquisition methods (e.g. EMA, X-ray, XRMB, and vocal tract MRI), UTI has the advantage that the tongue surface is fully visible, and ultrasound can be recorded in a non-invasive way [17, 18, 19]. An ultrasound device is easy to handle and move, since it is small and light, and thus it is suitable for fieldworks, as well. Besides, it is a significantly less expensive piece of equipment than the above mentioned devices.

In our earlier studies, we have shown that ultrasound is a feasible solution for articulatory-to-acoustic mapping [18, 20] and acoustic-to-articulatory inversion [21]. However, text-to-ultrasound synthesis has not been investigated before.

1.4. Contributions of this paper

The goal of this paper is to extend DNN-TTS with articulatory movement prediction, using ultrasound images of the tongue. We show on the data of several speakers that the combined TTS and synthesized articulatory motion is feasible and can result in acceptable articulatory movement video. Text-to-articulatory movement prediction might be useful for computer-assisted pronunciation training (CAPT) applications and articulatory visualization.

2. Methods

2.1. Data

For the data, we used the UltraSuite-TaL80 database [22] (`https://ultrasuite.github.io/data/tal_corpus/`). We chose four English male (03mn, 04me, 05ms, 07me) and four female speakers (01fi, 02fe, 06fe, and 09fe). In parallel with speech, the tongue movement was recorded in midsagittal orientation using the "Micro" ultrasound system of Articulate Instruments Ltd. at 81.5 fps. Lip video was also recorded in UltraSuite-TaL80, but we did not use that information in the current study. The ultrasound data and the audio signals were synchronized using the tools provided by Articulate Instruments Ltd. Each speaker read roughly 200 sentences – the duration of the recordings was about 15 minutes, which was partitioned into training, validation and test sets in a 85-10-5 ratio.

2.2. Processing the ultrasound data

In our experiments, articulatory features estimated from the raw scanline data of the ultrasound were used as the additional target of the networks (see Fig. 1). The 64×842 pixel images were resized to 64×128 pixels using bicubic interpolation, and we calculated PCA coefficients, similarly to EigenTongues [23]. While calculating the PCA, we aimed at keeping the 70% of the variance of the original images, thus having 128 coefficients. An example for the PCA eigenvectors can be seen in Fig. 2, and the result of PCA is presented in Fig. 4. To be in synchrony with the acoustic features (frame shift of 5 ms), the ultrasound data was resampled to 200 Hz.

2.3. DNN-TTS framework

Fig. 1 illustrates the proposed approach, i.e. the combined acoustic and articulatory feature prediction using a DNN from text input. The experiments were conducted in the Merlin DNN-TTS framework [24] (`https://github.com/CSTR-Edinburgh/merlin`). Textual / phonetic parameters are first converted to a sequence of linguistic features as input (based on a decision tree). Next, neural networks are employed to predict acoustic (60-dimensional MGC, 5-dimensional BAP, and 1-dimensional LF0, with delta and delta-delta features) and articulatory features (128-dimensional ULT-PCA, with delta and delta-delta) as output for synthesizing speech, at a 5 ms frame step with the WORLD vocoder. From the predicted 128-dimensional articulatory features, the 64×128 image is reconstructed using the PCA matrix, and bicubic interpolation is applied to resize the image to 64×842 pixels, to be comparable with the original data. For visualization purposes, we transformed this raw scanline data to wedge format, which shows how the real aspect ratio of the tongue surface (for an example, see Fig. 4). The transformation was done with 'ultrasuite-tools' (`https://github.com/UltraSuite/ultrasuite-tools`)

2.3.1. FC-DNN

The DNN used here is a fully-connected feed-forward multilayer perceptron architecture (FC-DNN, six hidden layers, 1024 neurons in each). We applied tangent hyperbolic activation function, SGD optimizer, and a batch size of 256. The input features had min-max normalization, while output acoustic features had mean-variance normalization. We trained the networks for 25 epochs with a warm-up of 10 epochs, applying early stopping, and a learning rate of 0.002 after that with exponential decay. We only trained both a duration model and an acoustic model, the latter also containing the articulatory features.

2.3.2. LSTM

Recurrent networks are typically more suitable to process sequential data. Therefore, we also trained an LSTM network following the Merlin recipe (four FF layers with 1024 neurons each, and one LSTM layer with 512 neurons). To ensure a longer training with the recurrent network, we used ADAM optimizer, and a warm-up of 30 epochs with early stopping. The other parameters were the same as for the FF-DDN. We trained both duration and acoustic models.

All neural network trainings were done individually with each speakers' data, without average voice training or adaptation.

3. Experimental results

After training the above models, we synthesized sentences from the test parts of the ultrasound datasets. To measure the validation and test error, we calculated both spectral prediction error (Mel-Cepstral Distortion, MCD), and an articulatory feature

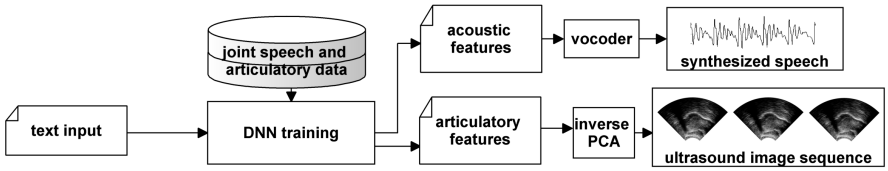


Figure 1: *Block diagram of the proposed approach.*

related error (ULT-PCA/RMSE, calculated on the normalized PCA values). We trained both duration and acoustic models, but for the error calculations, we synthesized the test sentences with their original timing. This way, warping the features in time was not necessary for calculating the error measures.

3.1. Demonstration samples

An example for the PCA eigenvectors are in Fig. 2, and the predicted articulatory feature sequence can be seen in Fig. 3 (speaker '01fi', sentence '201_aud'). As lower dimensional PCA vectors contain more information, the visualization was done in an exponential way and only 8 dimensions are plotted out of 128. Clearly, both the FC-DNN and the LSTM are following the tendencies found in the original data (e.g. in case of PCA-1, PCA-2, PCA-4), but the fine details are not modeled well. This type of oversmoothing is a frequent phenomena in statistical parametric speech synthesis. The higher dimensions (e.g. PCA-64 and PCA-128) are almost constant; i.e. they could not be modeled well with neural networks.

To visualize the individual ultrasound images, we plotted several ultrasound frames from the original videos and from the predicted ones, as a function of time, in Fig. 4. The reason why we are plotting every third frame is that for the 5 ms frame step of the Merlin toolkit, the 81.5 fps ultrasound video was interpolated to 200 Hz, and therefore, in the predicted data, roughly every 3rd frame contains visible tongue motion. In case of speaker '01fi', we can see in the top row (original ultrasound images after PCA) that there is a significant tongue movement, i.e. the tongue tip (on the right) goes higher, as the time passes. Both the predictions with the DNN and LSTM network follow the articulatory movement, but the images are smoothed – again, resulting from the statistical training. For speaker '03mn', similar tendencies can be observed: the movement of the tongue is changing its curvature as a function of time, but in the DNN-predicted and LSTM-predicted images, the tongue surface is not as clear as in the original data.

As the synthesized motion of the tongue is more visible in real-time videos, we made available several samples at `http://smartlab.tmit.bme.hu/ssw11_txt2ult`.

3.2. Objective measures

Table 1 summarizes the MCD results. The MCD values of the test sentences with the FC-DNN are between 5.8–7.0 dB (average: 6.228 dB), whereas with LSTM they are between 6.0–7.5 dB (average: 6.593 dB), indicating that the recurrent neural network was not helpful in estimating the acoustic features. The reason for this might be that we have limited articulatory-acoustic databases (roughly 200 sentences for each speaker), which is too small for training an LSTM model.

The results of the RMSE calculated on the articulatory feature are summarized in Table 2. The lowest error was achieved with the data of speaker 09fe: with FC-DNN, the test error is 2.9, while with LSTM, the test error is 3.1. The tendency is similar to the case of MCD: the LSTM network was not helpful in predicting the articulatory features, probably due to the small size of the data.

Table 1: *MCD errors on the dev/test set.*

	MCD	
Speaker	FC-DNN	LSTM
01fi	6.995 / 6.971	6.647 / 6.588
02fe	6.095 / 5.803	6.486 / 6.259
03mn	5.781 / 5.785	5.977 / 5.948
04me	5.896 / 6.024	6.318 / 6.312
05ms	6.244 / 6.256	7.235 / 7.083
06fe	5.758 / 5.582	6.444 / 6.330
07me	6.589 / 6.562	6.831 / 6.749
09fe	6.516 / 6.844	7.197 / 7.472
average	6.234 / 6.228	6.642 / 6.593

Table 2: *ULTPCA/RMSE errors on the dev/test set.*

	ULTPCA128/RMSE	
Speaker	FC-DNN	LSTM
01fi	3.292 / 3.223	3.319 / 3.208
02fe	3.533 / 3.732	3.753 / 3.904
03mn	3.147 / 3.660	3.289 / 3.680
04me	3.849 / 3.985	4.031 / 4.033
05ms	3.133 / 3.233	3.249 / 3.405
06fe	3.439 / 3.250	3.743 / 3.451
07me	3.544 / 3.595	3.498 / 3.461
09fe	3.022 / 2.864	3.234 / 3.133
average	3.370 / 3.443	3.515 / 3.534

4. Discussion and conclusions

We have shown above that text-to-ultrasound video prediction is feasible as an extension to traditional DNN-based text-to-speech synthesis, despite the relatively small amount of training data. Although the synchrony between visual and speech output is not enforced by the model, the tied acoustic and articulatory features during the DNN training ensure that the audio and visual features are in synchrony, i.e. that in the generated ultrasound videos, the tongue is moving according to the synthesized speech. To objectively check this, SyncNet, part of Wav2Lip could be applied to assess synchrony [25]. Our paper found that the joint learning of both acoustic and articulatory features has advantages, but this is not substantiated – a comparison of the joint model against two separate models remains

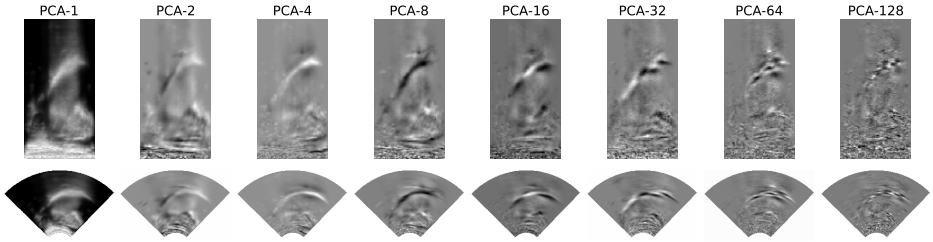


Figure 2: *PCA eigenvectors, from speaker '01fi'. Top: raw, scanline data (resized to 64×128 pixels). Bottom: wedge orientation.*

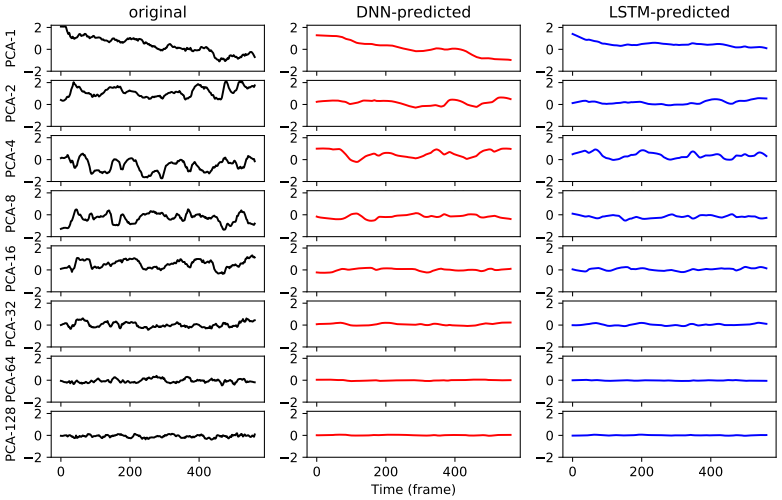


Figure 3: *Original and predicted articulatory features, from speaker '01fi'. Sentence: '"I leave it to nobody," said Shakespeare, sulkily.'*

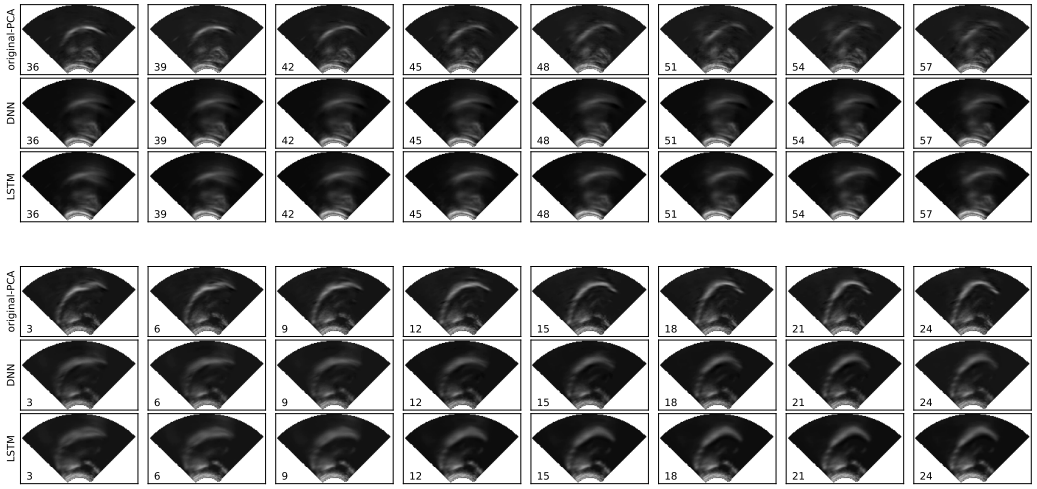


Figure 4: *Original and predicted articulatory feature sequences. Top: speaker '01fi', bottom: speaker '03mn'. The numbers at the bottom-left correspond to the frame number of the video.*

future work.

Although there have been several earlier attempts for extending text-to-speech synthesis with articulatory data, all of these studies were using EMA, being a point tracking equipment [10, 11, 12, 13, 14, 15], and containing less spatial information about the tongue than ultrasound. The advantage of ultrasound in this context is that the resulting video shows a larger portion of the tongue, compared to EMA.

Articulatory movement prediction from text input can be useful for audiovisual speech synthesis. A specific application is computer-assisted pronunciation training / computer-aided language learning [26, 27, 28], which can be beneficial for learners of second languages. With such a combined TTS and text-to-articulatory prediction system, by giving an arbitrary input text, one is able to hear the speech and, in synchrony with it, see how to move the tongue in 2D or 3D to produce target speech sounds. This visual feedback can be helpful for pronunciation training in L2 learning, especially when the target language contains speech sounds which are difficult to articulate.

In the future, we plan to investigate speaker adaptation and speaker-independent training. For this, a common articulatory space has to be defined, as the currently used PCA representation is specific for each individual speaker. Also, multi-task learning might be useful in this context: a system could potentially be pre-trained on speech-only material which is easier to acquire, and the UTI be trained only in addition. Besides, we plan to investigate the effect of the misalignments in the ultrasound transducer position [29, 30] on the text-to-ultrasound prediction results.

The code is accessible at `https://github.com/BME-SmartLab/txt2ult`.

5. Acknowledgements

The author was partly funded by the National Research, Development and Innovation Office of Hungary (FK 124584 and PD 127915 grants). We would like to thank CSTR for providing the Merlin toolkit and the UltraSuite-TaL articulatory database.

6. References

[1] H. Zen, T. Nose, J. Yamagishi, S. Sako, T. Masuko, and A. Black, "The HMM-based speech synthesis system version 2.0," in *Proc. ISCA SSW6*, Bonn, Germany, 2007, pp. 294–299.

[2] H. Zen, A. Senior, and M. Schuster, "Statistical parametric speech synthesis using deep neural networks," in *Proc. ICASSP*, Vancouver, Canada, 2013, pp. 7962–7966.

[3] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior, and K. Kavukcuoglu, "WaveNet: A Generative Model for Raw Audio," *CoRR*, vol. abs/1609.0, 2016.

[4] R. Prenger, R. Valle, and B. Catanzaro, "Waveglow: A Flow-based Generative Network for Speech Synthesis," in *Proc. ICASSP*, Brighton, UK, 2019, pp. 3617–3621.

[5] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan, R. A. Saurous, Y. Agiomvrgiannakis, and Y. Wu, "Natural TTS Synthesis by Conditioning Wavenet on MEL Spectrogram Predictions," in *Proc. ICASSP*, Calgary, Canada, 2018, pp. 4779–4783.

[6] D. W. Massaro, M. M. Cohen, M. Tabain, J. Beskow, and R. Clark, "Animated speech: research progress and applications," in *Audiovisual Speech Processing*, G. Bailly, P. Perrier, and E. Vatikiotis, Eds. Camridge, UK: Cambridge University Press, 2012, pp. 309–345.

[7] P. Perrier, ""GEPPETO": A target-based model of speech production including optimal planning and physical modeling," in *Adventures in Speech Science*, Tokyo, Japan, 2014.

[8] D. Schabus, M. Pucher, and G. Hofer, "Joint audiovisual Hidden Semi-Markov Model-based speech synthesis," *IEEE Journal on Selected Topics in Signal Processing*, vol. 8, no. 2, pp. 336–347, 2014.

[9] I. Stavness, M. A. Nazari, F. Cormac, P. Perrier, Y. Payan, J. Lloyd, and S. Fels, "Coupled Biomechanical Modeling of the Face, Jaw, Skull, Tongue, and Hyoid Bone," in *3D Multiscale Physiological Human*, R. O. C. H. F. E. Magnenat-Thalmann Nadia, Ed. Springer London, 2014, pp. 253–274.

[10] Z.-H. Ling, K. Richmond, and J. Yamagishi, "An Analysis of HMM-based prediction of articulatory movements," *Speech Communication*, vol. 52, no. 10, pp. 834–846, oct 2010.

[11] ——, "HMM-Based Text-to-Articulatory-Movement Prediction and Analysis of Critical Articulators," in *Proc. Interspeech*, Makuhari, Japan, 2010, pp. 2194–2197.

[12] Z. Wei, Z. Wu, and L. Xie, "Predicting articulatory movement from text using deep architecture with stacked bottleneck features," in *Proc. APSIPA*, Jeju, South Korea, 2016, pp. 1–6.

[13] I. Steiner, S. Le Maguer, and A. Hewer, "Synthesis of Tongue Motion and Acoustics from Text Using a Multimodal Articulatory Database," *IEEE/ACM Transactions on Audio Speech and Language Processing*, vol. 25, no. 12, pp. 2351–2361, 2017.

[14] S. Le Maguer, I. Steiner, and A. Hewer, "An HMM/DNN comparison for synchronized text-to-speech and tongue motion synthesis," in *Proc. Interspeech*. Stockholm, Sweden: International Speech Communication Association, 2017, pp. 239–243.

[15] L. Yu, J. Yu, and Q. Ling, "BLTRCNN Based 3D Articulatory Movement Prediction: Learning Articulatory Synchronicity From Both Text and Audio Inputs," *IEEE Transactions on Multimedia*, vol. PP, no. c, p. 1, 2018.

[16] M. Stone, B. Sonies, T. Shawker, G. Weiss, and L. Nadel, "Analysis of real-time ultrasound images of tongue configuration using a grid-digitizing system," *Journal of Phonetics*, vol. 11, pp. 207–218, 1983.

[17] M. Stone, "A guide to analysing tongue motion from ultrasound images," *Clinical Linguistics and Phonetics*, vol. 19, no. 6-7, pp. 455–501, jan 2005.

[18] T. G. Csapó, T. Grósz, G. Gosztolya, L. Tóth, and A. Markó, "DNN-Based Ultrasound-to-Speech Conversion for a Silent Speech Interface," in *Proc. Interspeech*, Stockholm, Sweden, 2017, pp. 3672–3676.

[19] V. Ramanarayanan, S. Tilsen, M. Proctor, J. Töger, L. Goldstein, K. S. Nayak, and S. Narayanan, "Analysis of speech production real-time MRI," *Computer Speech and Language*, vol. 52, pp. 1–22, 2018.

[20] T. G. Csapó, C. Zainkó, L. Tóth, G. Gosztolya, and A. Markó, "Ultrasound-based Articulatory-to-Acoustic Mapping with WaveGlow Speech Synthesis," in *Proc. Interspeech*, 2020, pp. 2727–2731.

[21] D. Porras, A. Sepúlveda-Sepúlveda, and T. G. Csapó, "DNN-based Acoustic-to-Articulatory Inversion using Ultrasound Tongue Imaging," in *International Joint Conference on Neural Networks*, Budapest, Hungary, 2019, pp. N–19 221.

[22] M. S. Ribeiro, J. Sanger, J.-X. X. Zhang, A. Eshky, A. Wrench, K. Richmond, and S. Renals, "TaL: a synchronised multi-speaker corpus of ultrasound tongue imaging, audio, and lip videos," in *2021 IEEE Spoken Language Technology Workshop (SLT)*, Shenzhen, China, 2021, pp. 1109–1116.

[23] T. Hueber, G. Aversano, G. Chollet, B. Denby, G. Dreyfus, Y. Oussar, P. Roussel, and M. Stone, "Eigentongue feature extraction for an ultrasound-based silent speech interface," in *Proc. ICASSP*, Honolulu, HI, USA, 2007, pp. 1245–1248.

[24] Z. Wu, O. Watts, and S. King, "Merlin: An Open Source Neural Network Speech Synthesis System," in *9th ISCA Speech Synthesis Workshop*. Sunnyvale, CA, USA: ISCA, sep 2016, pp. 202–207.

[25] K. R. Prajwal, R. Mukhopadhyay, V. P. Namboodiri, and C. V. Jawahar, "A Lip Sync Expert Is All You Need for Speech to Lip Generation In the Wild," in *Proceedings of the 28th ACM International Conference on Multimedia*, New York, NY, USA, 2020, pp. 484–492.

[26] W. Katz, T. Campbell, J. Wang, E. Farrar, J. Eubanks, A. Balasubramanian, B. Prabhakaran, and R. Rennaker, "Opti-speech: A real-time, 3D visual feedback system for speech training," in *Proc. Interspeech*, Singapore, Singapore, 2014, pp. 1174–1178.

[27] D. Jones, "Development of Kinematic Templates for Automatic Pronunciation Assessment Using Acoustic-to-Articulatory Inversion," *Master's Thesis*, jul 2017.

[28] C. Agarwal and P. Chakraborty, "A review of tools and techniques for computer aided pronunciation training (CAPT) in English," *Education and Information Technologies*, vol. 24, no. 6, pp. 3731–3743, nov 2019.

[29] T. G. Csapó and K. Xu, "Quantification of Transducer Misalignment in Ultrasound Tongue Imaging," in *Proc. Interspeech*, online, 2020, pp. 3735–3739.

[30] T. G. Csapó, K. Xu, A. Deme, T. E. Gráczi, and A. Markó, "Transducer Misalignment in Ultrasound Tongue Imaging," in *12th International Seminar on Speech Production*, 2020.

11th ISCA Speech Synthesis Workshop (SSW 11)
26-28 August 2021, Budapest, Hungary

Impact of Segmentation and Annotation in French end-to-end Synthesis

Martin Lenglet, Olivier Perrotin, Gérard Bailly

Univ. Grenoble Alpes, CNRS, Grenoble-INP, GIPSA-lab, France
{martin.lenglet,olivier.perrotin,gerard.bailly}@grenoble-inp.fr

Abstract

Audio books are commonly used to train text-to-speech models (TTS), as they offer large phonetic content with rather expressive pronunciation, but number and sizes of publicly available audio books corpora differ between languages. Moreover, the quality and accuracy of the available utterance segmentations are debatable. Yet, the impact of segmentation on the output synthesis is not well established. Additionally, utterances are generally used individually, without taking advantage of text level structuring information, even though they influence speaker reading. In this paper, we conduct a multidimensional evaluation of Tacotron2 trained on different segmentations and text level annotations of the same French corpus. We show that both spectrum accuracy and expressiveness depend on the segmentation used. In particular, a shorter segmentation, in addition with the annotation of paragraphs, benefits to spectrum reconstruction at the detriment of phrasing. Multidimensional analysis of mean opinion scores obtained with a MUSHRA-experiment revealed that phrasing was relatively more important than spectrum accuracy in perceptual judgement. This work serves as evidence that particular attention must be given to models evaluation, as well as how to use the training corpus to maximize synthesis characteristics of interest.

Index Terms: Speech Synthesis, French TTS, mixed-inputs TTS, French dataset

1. Introduction

In recent years, deep learning met huge success in language-related applications. In particular, state-of-the-art text-to-speech (TTS) models [1, 2, 3] coupled with neural vocoders [4, 5] achieve synthesis quality close to natural speech. As always with deep learning, the quality of the output heavily depends on the dataset used for training. The common approach of neural TTS, seen in events like Blizzard Challenge [6], is to compare multiple models on the same corpus to evaluate the resulting synthesis quality. This process minimizes the importance of input data structuring, which ultimately shapes the output of any deep learning model. One complementary work is to evaluate multiple segmentations of data structuring on the same TTS model. This paper adopts this approach.

Publicly available corpora designed to train TTS [7, 8, 9] are generally composed of audio book extracts read by one or more speakers, segmented in thousands of utterances. Utterances' lengths vary between 1 to 20 seconds, with boundaries often matching sentences, but not always. Even if these databases have been used to train state-of-the-art speech models [3, 10], long utterances may not be the best candidates to train TTS: (i) Large batch size with long utterances rely on high computation memory. (ii) Learning long-term dependencies is a challenging task for sequential models [11]. (iii) Style control, which is an increasing demand of the field, massively uses utterance level style embeddings [12, 13], which means that the shorter the utterances, the finer it is possible to tune speech style at inference time. These reasons made us consider a shorter segmentation may be better suited to train TTS efficiently.

Proposing a new segmentation gives us the opportunity to integrate specific annotations in the input data to give models relevant context information regarding the corresponding speech to produce: (i) End of paragraph are generally associated with specific phrasing modifications from the speaker, and are then worth noticing during training. (ii) In French, silent letters and optional liaisons are common, which are additional difficulties to train a TTS model on orthographic inputs alone. The addition of phonetic annotations contributes to alleviate this issue, and has shown to benefit to both transcriptions [14].

This paper presents a multidimensional comparison between the proposed segmentation and annotation of the LibriVox French corpus [15] and the original segmentation from M-AILABS [7], used to train the same Tacotron2 [1]. We evaluate the phrasing and spectral accuracy of each model. These objective measurements are paired with mean opinion scores evaluated through a MUSHRA-like experiment [16].

2. Related Work

To our knowledge, there is no publicly available French Tacotron2. Recent studies published on French synthesis focus on concatenation based TTS [17] or use Deep Convolutional TTS (DCTTS) [18]. DCTTS is a fully convolutional neural TTS, whose initial purpose was to alleviate the need for high computational power, while enabling quick training on smaller database. Although synthesis reaches acceptable standards, the overall quality does not match more recent models [1, 2, 3].

The later TTS explore the well established encoder-decoder architecture: the encoder converts the input sequence into a hidden representation that the decoder uses to generate mel-spectrogram frames. As an interface between the two, Tacotron2 [1] employs a location-sensitive attention [19] module which computes a fixed length vector for each decoder step. The encoder adopts an approach that is similar to the classical language model processing pipeline: the input sequence is passed through three convolutional layers that compute local pattern, followed by bidirectional LSTM. Alternatively, Transformer TTS [2] and Fastspeech [3] introduce self-attention and multi-head attention layers as a replacement for recurrent units. These three models produce synthetic speech of similar quality [3]. We chose Tacotron2 for its relative ease to implement and straight training process. Additionally, Tacotron2 shows promising results for expressive control [12, 13], which is also one of our short term goal.

Although mean opinion scores are generally used to assess the global quality of TTS, this evaluation takes multiple aspects of speech into account: phonetic correctness and intelligibility,

10.21437/SSW.2021-3

spectral smoothness, expressiveness, etc. These clues may not vary conjointly, which means that the use of a single metric may not be sufficient. [20, 21] employ multidimensional scaling (MDS) [22] to extend the quality analysis of TTS models. This paper prolongs this perspective.

3. Proposed Method

This section presents the original baseline and the new segmentation proposed from the French LibriVox dataset, and the modifications added to the Tacotron2 implementation shared by NVIDIA[1]. Our implementation[2] and database[3] are available online.

3.1. Segmentation and Annotation

3.1.1. Original Database

We used the M-AILABS French dataset [7] as a starting point. This corpus includes more than 190h of recorded speech, segmented in utterances from 1s to 20s, given with corresponding orthographic transcripts. Recordings come from the free public domain audio books LibriVox database [15]. We selected a subset of the recordings made by Nadine Eckert-Boulet (NEB), for a total duration of 34h. Each book duration and corresponding number of utterances are given in Table 1. Audio files are originally sampled at 16000Hz, but we re-sampled them at 22050Hz.

Table 1: *Books duration (and number of utterances) for original and new segmentation of the M-AILABS French corpus.*

Book	Original	New segmentation
Les Mystères de Paris	22:31:27 (12285)	21:37:21 (25458)
Mme Bovary	11:39:50 (5775)	11:08:55 (12781)
Total	34:11:17 (18060)	32:46:16 (38239)

The orthographic transcript is given by the Gutenberg Project[4]. It is worth mentioning that NEB does not always strictly follow the original text. Some miss-spelling remain (for example: "precepteur" is said instead of "percepteur"), as well as some omissions. These miss-alignments correspond to 0.1% of the original corpus. We did not correct any of those transcripts for the baseline. Though, we spelled out all texts, including frequently used abbreviations in French ("M.": "Monsieur", "Mlle": "Mademoiselle", "n°": "numéro" and "etc": "et cetera"), and numbers ("1838": "dix-huit cent trente-huit"). Two punctuation marks were also replaced to stand as a single unique character: "..." was replaced by "…", "–" by "¬".

Each clip was originally bounded with 500ms of silence (zeros in the waveform) at the beginning and the end. These silences do not correspond to the recordings, but have been artificially added to each audio clip after segmentation. To limit the duration of initial and final silences in the synthesis, we truncated these silences at 130ms. This duration matches the initial and final silence lengths found in other speech databases such as LJspeech [8].

3.1.2. Re-segmentation

To reduce the average duration of utterances, we first restore the initial audio books chapters structure by aligning the original text from the Gutenberg Project with the recordings from LibriVox. As for the original segmentation, all texts are spelled out, but previously mentioned miss-spelling and omissions are now manually corrected. In addition, end of paragraphs are annotated with the punctuation mark "§", which is introduced after the last punctuation mark preceding each carriage return. Ends of paragraphs are accompanied by phrasing patterns of NEB, that are worth highlighting in the training corpus. For instance, Table 2 shows F0 and elongation of the final syllable before ends of paragraph vs. paragraph-internal periods, as well as their values for the following syllable. The last syllable is generally longer before the end of paragraph, and the F0 gap across the boundary is increased (6.45 vs. 5.11 semitones respectively).

Table 2: *Comparison of F0 and elongation of syllable [23] around ends of paragraph (.§) and intermediate periods (.).*

		Syllable Previous	Syllable Following
Elongation (%)	.	+184	+21
	.§	+218	+24
F0 (semitone)	.	1.96	7.01
	.§	0.96	7.41

Chapters are then segmented based on silences of at least 400ms. This duration usually corresponds to pauses made between speaking turns in conversations [24]. 94.56% of silences coincide with punctuation marks. For the others, a comma is added at the end of the utterance. 130ms of ambient silence from the recording are kept at the beginning and the end of each utterance. Timestamps were hand-checked for each utterance to ensure optimal segmentation. Table 1 shows duration and number of utterances of the obtained segmentation. Note that the proposed segmentation is 01:25:01 shorter than the original, due to the reduction of intra-utterance silences, but that reduction does not impact either the text read nor the speaking rate.

Fig.1 gives the distribution of utterances length of the original and the proposed segmentation. Median utterance length (resp. first and third quartiles) are reduced from $6.44s$ ($3.88s$ and $9.26s$) to $2.77s$ ($1.89s$ and $3.95s$). 82.5% of utterances of the new segmentation last between $1s$ and $5s$, and 0.25% of utterances last more than $10s$. 1336 utterances are unchanged, which corresponds to 7.4% and 3.5% of the original and new segmentation respectively.

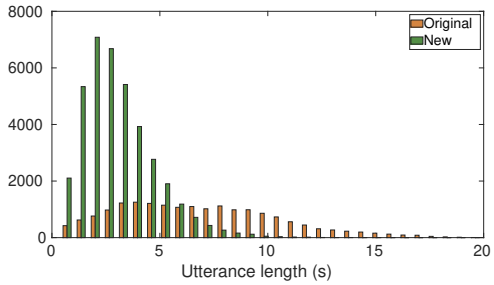


Figure 1: *Distribution of utterances length of original and new segmentation.*

[1] https://github.com/NVIDIA/tacotron2
[2] https://github.com/MartinLenglet/Tacotron2
[3] https://zenodo.org/record/4580406#.YI_qIyaxXmE
[4] https://www.gutenberg.org/

3.1.3. Phonetic Annotation

Training of both orthographic and phonetic transcripts, called representation mixing, enables to use both input types in the same utterance at inference time, and thus remove some ambiguities on particular issues, without the need for the whole phonetic transcript of the speech to synthesize. For instance, NEB performs numerous optional liaisons (22999 liaisons in the corpus of which 9597 [z], 9029 [t] and 3412 [n]), in particular bridging 844 infinitives and prepositions with [ʁ/]. Yet, these liaisons are not systematic, and adding the possibility to choose if the liaison is being made at inference time (as part of a style component) would be interesting. To study the impact of phonetic annotation, hand-crafted phonetic alignment is performed on the whole new segmentation.

3.2. Modifications of Tacotron2

3.2.1. Representation mixing

We introduce the mixed embedding matrices described in [14] in our model to give the possibility to train with both types of inputs. Contrary to [14], when the training includes phonetic inputs, input types are not mixed within the same utterance. The number of utterances is simply doubled, with the same audio file corresponding to both the orthographic and the phonetic input.

3.2.2. Gate loss correction

Synthesizing short utterances, typically one or two words, has been shown to be a challenging task for TTS models [25]. Recurrent artifacts are repetition of the last syllable, or unintelligible words. With our proposed segmentation, 5% of utterances last less than 1s, which might cause some issues during inference. To avoid this, we fine tune the training of each model with 2 modifications: (i) 9 frames of recorded ambient silence are added at the end of each utterance, in which the end-of-sequence probability is set to 1. This silence originates from the pause following each utterance. (ii) a multiplying factor is added to the gate loss error before back-propagation. We empirically found that these modifications correct previously mentioned artifacts, and improve the overall synthesis quality. The benefits of these modifications are evaluated in section 4.

4. Experiments and Results

4.1. Experimental Setup

The 6 models trained for this experiment are presented below:

- O **and** O_g are trained on the original segmentation from M-AILABS for 200 epochs.
- N **and** N_g are trained on the new segmentation proposed in section 3.1.2, with only orthographic inputs for 200 epochs.
- P **and** P_g are trained on the new segmentation proposed in section 3.1.2, with both orthographic and phonetic inputs for 100 epochs, since each epoch corresponds to twice the number of utterances of the orthographic models.

Models annotated $_g$ are fine-tuned with the gate loss correction. The multiplying factor is set to 10 for these models. This correction is introduced for the last quarter of the training epochs. Before that separation, only one model is trained using warm-start from the English model trained on LJSpeech shared by NVIDIA. The postnet is bypassed during the first 10 epochs, and the learning rate is fixed at 10^{-3}. This phase enables the model to initiate a coarse transition from English to French. Then the postnet is reactivated and the learning rate decreases exponentially until reaching 10^{-5} at 90 epochs. The batch size is limited to 32, due to memory limitations with long utterances of the original segmentation, and thus is set to 32 for all models. Batches are randomly picked among utterances of approximate same length.

We pick 5% of the original corpus as test set. To ensure a fair comparison between models, these 903 utterances are randomly selected among the 1336 common utterances between the original and the new segmentation. Thus, the amount of speech seen by each model during training is rigorously the same. Only the orthographic transcript of the test set is used in this section, even for models P and P_g. Note that this test set does not favor the new segmentation: phonetic inputs and paragraphs markers are not used.

The vocoder used is WaveRNN [5]. WaveRNN is faster and demands less resources than the original WaveNet [4] used by [1], and still provides a good voice quality [26]. We trained WaveRNN from scratch for 1000 epochs on the new segmentation from Table 1 with a learning rate of 10^{-4}. Then we fine-tuned the model with 520 more epochs at a learning rate of 10^{-5}.

4.2. Objective measurements

4.2.1. Accuracy

We evaluate the spectral accuracy of each model through the proximity of the generated spectra with the *vocoded* ground truth (GT). Since syntheses differ in length, mel-spectrograms are first aligned by dynamic time warping (DTW) [27]. Mean squared error (MSE) on aligned spectrograms are then computed and averaged on the test set; results are shown in Fig. 2.

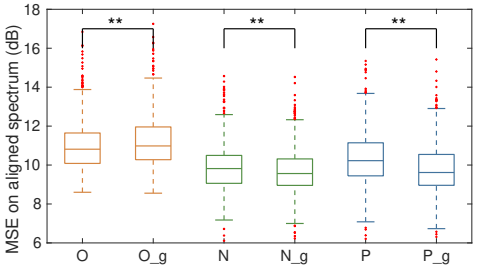


Figure 2: *Mean squared error between models and ground truth, calculated on mel-spectrograms aligned by dynamic time warping. ** indicates a significant effect of the gate loss correction according to Tukey-Kramer test ($p < 0.05$).*

The model has a statistical effect on the computed distances according to a one-way ANOVA ($F = 246.5$, $p < 0.001$). Tukey-Kramer multiple comparisons show that all pairs are statistically different, except P_g/N and P_g/N_g. The gate loss correction has a significant impact on all models. The new segmentation decreases the spectral distortion, with a beneficial contribution of the gate loss correction in this case. On the other hand, this correction decreases the spectral accuracy of the model trained on the original segmentation.

4.2.2. Phrasing

Pauses position and duration contribute to the expressiveness of speech [28]. We computed mean speech and silence duration across the whole synthesised test set for each model and for

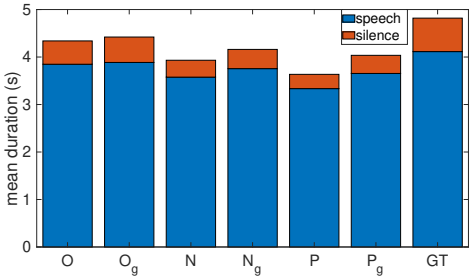


Figure 3: *Mean utterance duration on the whole test set for each model.*

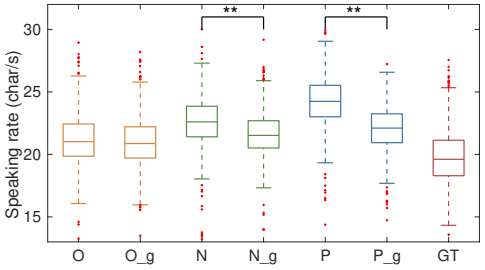


Figure 4: *Speaking rate of each model, calculated on each utterance of the test set. Speaking rate is estimated in characters per second, pause durations are not taken into account. ** indicates a significant effect of the gate loss correction ($p < 0.05$).*

GT. By extension, this calculation also enables us to estimate the speaking rate of each model on the test set. Mean utterance duration and speaking rate are shown in Fig. 3 and Fig. 4 respectively.

Models trained on the new segmentation do not exhibit the same temporal behavior than models trained on the original segmentation. Utterances mean duration is smaller with the new segmentation ($3.93s$ and $3.64s$ compared to $4.44s$ for N, P and O respectively). Silences duration are also proportionally smaller: 9.2%, 8.2% and 11.3% for N, P and O respectively. As a result, the speaking rate increases with the new segmentation. Note that the speaking rate of all models is significantly higher than GT. The gate loss correction tends to reduce the differences observed compared to GT. Not only silences duration are increased, but also speech duration, resulting in a lower speaking rate. This decrease is statistically significant for the new segmentation, but not for the original. All other pairs are significantly different according to Tukey-Kramer multiple comparisons.

Longer pauses observed with O and O_g may result from the intra-utterance pauses frequency and duration in the original segmentation provided by M-AILABS. In that case, models are trained on audio clips that sometimes contain pauses longer than 1s, and thus reproduce that behavior during inference. On the contrary, the re-segmentation processing avoids intra-silences longer than 400ms, resulting in a more straight-forward synthesis.

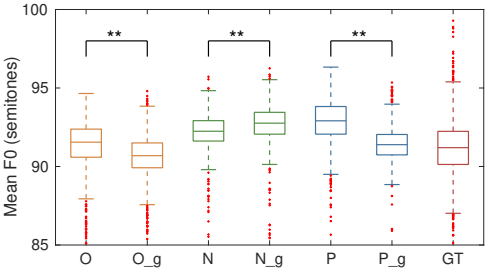


Figure 5: *Mean fundamental frequency calculated on voiced sections of each utterance of the test set. ** indicates a significant effect of the gate loss correction ($p < 0.05$).*

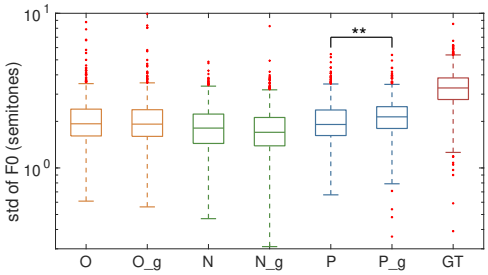


Figure 6: *Standard deviation of fundamental frequency calculated on each utterance of the test set. ** indicates a significant effect of the gate loss correction ($p < 0.05$).*

4.2.3. Pitch

As additional prosody measurements, we evaluate the pitch of each model using the Praat software [29]. The mean fundamental frequency (F0) and standard deviation of F0 is measured on voiced sections for every utterance of the test set. Results are given in Fig. 5 and Fig. 6 respectively.

One-way ANOVA shows a statistical effect of the model on both mean F0 and standard deviation of F0. Regarding mean F0, Tukey-Kramer multiple comparisons show that all pairs differ significantly, except O/P_g, O/GT and N_g/P. As to standard deviation of F0, only phonetic models P and P_g exhibit a significant effect of the gate loss correction, while both P and P_g are not statistically different from O and O_g. N and N_g have significantly lower standard deviation than all other models.

The new segmentation increases mean F0, but this effect is partially compensated when training the model on mixed inputs with gate loss correction. Similarly, the gate loss correction induces a lower mean F0 when training on the original segmentation. None of the presented models show standard deviation of F0 similar to GT, which might lead to less expressive synthetic voices.

4.3. Subjective evaluation

In accordance with objective measurements presented in section 4.2, 3 models were selected to evaluate the mean opinion scores through a MUSHRA-like experiment [16]. We keep only models that have been fine-tuned with gate loss correction, as they generally exhibit the closest proximity with GT behavior.

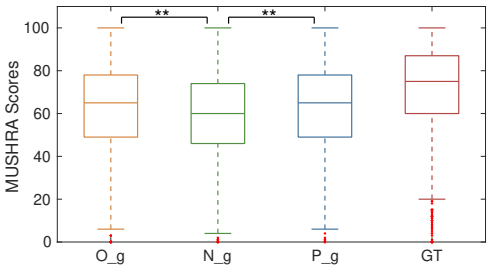


Figure 7: *MUSHRA results. ** indicates a significant difference between models ($p < 0.05$).*

GT is added as high anchor for the MUSHRA. This perceptive test was performed online using the webMUSHRA framework [30]. Utterances containing less than 7 words and more than 23 words were excluded from this test to keep only the central 90% of the test set length distribution. 60 utterances were randomly selected in the remaining test set, with equivalent representation of utterance lengths in the selection. 13 of the selected utterances contained one phonetic mistake (5 in O_g, 3 in P_g, and 5 in all models), and were replaced before the experiment. Participants were separated in 2 groups, each group listened to 30 out of the 60 selected utterances. For each utterance, participants were given the original text input, and were asked to evaluate the 4 given conditions (3 models + GT) according to the voice quality. No explicit reference was given during the listening. The experiment began with 5 minutes of training during which participants listened to a variety of synthesis that they were about to hear during the experiment and learned how to use the webMUSHRA interface. Audio examples are available online [5]. 44 participants recruited on Prolific [31] and aged 18-65 took part in the experiment. Participants were French native speakers, and had little or no previous experience with listening tests. Results of the MUSHRA are given in Fig.7

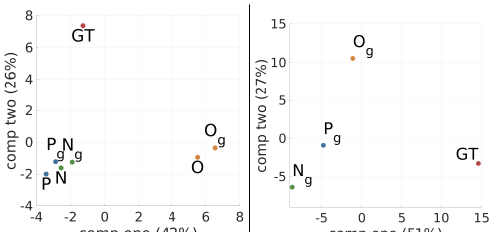
We compared the median score of each model using a Wilcoxon rank sum test. Differences are significant if $p < 0.05$. GT exhibits a significantly higher score than the 3 evaluated models. N_g scores significantly lower than all other models. No statistical differences are shown between O_g and P_g.

4.4. Multidimensional analysis

Despite the differences on specific expressiveness clues measured in section 4.2, subjective evaluation performed in section 4.3 does not exhibit a clear perceptive preference for one of the models O_g or P_g. To explore implicit dimensions of the evaluation of the models, we use a multidimensional analysis of the distances computed between each model and GT. These distances are evaluated on both objective and subjective measurements:

- **Subjective distances:** absolute score differences between all possible condition pairs evaluated in the MUSHRA, averaged across all participants and all utterances.
- **Objective distances:** MSE between all possible conditions pairs computed on mel-spectrograms aligned by DTW [27]. Objective distances are averaged across all 903 utterances of the test corpus.

[5] http://www.gipsa-lab.fr/~martin.lenglet/segmentation_impact/index.html



(a) *Objective MDS: MDS_O* (b) *Subjective MDS: MDS_S*

Figure 8: *Multidimensional scaling of distances between pairs conditions. Left and right graphs show objective and subjective distances respectively. Proportions of variance explained are given for each component.*

Table 3: *Correlation coefficients between objective measurements and components of MDS. * and ** indicate $p < 0.1$ and $p < 0.05$ respectively. ASE: aligned spectrum error, SR: Speaking rate, PD: pauses duration.*

MDS	Dim	objective measurements				
		ASE	SR	PD	mean F0	std F0
Obj	1	**0.90****	-0.47	0.44	**-0.71***	-0.06
	2	0.63	**-0.74***	**0.89****	-0.43	**0.97****
Subj	1	0.89	**-0.93***	**0.96****	-0.50	**0.98****
	2	0.97	-0.02	0.13	-0.83	-0.17

Then, we projected the two obtained distances matrices in two independent 2-dimensions space using classical Multidimensional scaling (MDS) [22]. To give a better idea of the impact of the gate loss correction, both corrected and non-corrected models were included in the objective MDS. Subjective and objective MDS (named MDS_S and MDS_O respectively in the following) are given in Fig.8.

Correlations between objective measurements computed in section 4.2 and the components of both MDS are estimated. Correlations coefficients are given in Table 3. Note that GT is not considered for correlation with aligned spectrum error (ASE). Correlation coefficients indicate that prosodic clues like pauses duration and standard deviation of F0 are closely related to the second component of MDS_O, but to the first component of MDS_S. On the other hand, spectral accuracy measurements ASE and mean F0 are correlated to the first component of MDS_O, and similarly for the second component of MDS_S, even if this tendency is not significant. Two main dimensions emerge in both evaluations: spectrum accuracy and expressiveness. The axis inversion (and associated portion of variance explained) tends to show these dimensions are not given the same importance in the perceptive judgement than in the objective measurement. As a result, the proximity of spectrum quality observed between GT and models trained on new segmentation on the first component of Fig.8a is downgraded to the second component of Fig.8b. Respectively, expressiveness is given more importance in the perceptive test than it is in the objective measurements, resulting in O_g being closer to GT in the first component of Fig.8b. Fig.8a emphasizes the benefits of the proposed gate loss correction, as all models annotated $_g$ are closer to GT on the expressiveness dimension.

5. Conclusions and Discussion

We have proposed a shorter segmentation of the French M-AILABS corpus and compared the training of Tacotron2 on both original and new datasets. Through multi dimensional evaluation, we have shown that the way speech data are segmented impacts both quality and expressiveness factors in opposite directions. Future works should elaborate on how to combine the advantages of both segmentation with curriculum training. An important contribution of this work is the addition of the gate loss correction as a fine tuning of the model, which contributes to improve prosodic aspects of the synthesized speech. The use of multidimensional analysis of mean opinions scores introduces relevant nuances to the MUSHRA results. The structuring of the subjective notation latent space, as well as the prediction of positions in this space thanks to objective measurements should be the focus of future works.

6. Acknowledgments

This research has received funding from the BPI project THERADIA and MIAI@Grenoble-Alpes (ANR-19-P3IA-0003). This work was granted access to HPC/IDRIS under the allocation 2021-AD011011542R1 made by GENCI.

7. References

[1] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan *et al.*, "Natural tts synthesis by conditioning wavenet on mel spectrogram predictions," in *ICASSP*. IEEE, 2018, pp. 4779–4783.

[2] N. Li, S. Liu, Y. Liu, S. Zhao, and M. Liu, "Neural speech synthesis with transformer network," in *AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 6706–6713.

[3] Y. Ren, Y. Ruan, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, "Fastspeech: Fast, robust and controllable text to speech," in *Advances in Neural Information Processing Systems*, vol. 32. Curran Associates, Inc., 2019.

[4] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[5] N. Kalchbrenner, E. Elsen, K. Simonyan, S. Noury, N. Casagrande, E. Lockhart, F. Stimberg, A. Oord, S. Dieleman, and K. Kavukcuoglu, "Efficient neural audio synthesis," in *International Conference on Machine Learning*. PMLR, 2018, pp. 2410–2419.

[6] X. Zhou, Z.-H. Ling, and S. King, "The blizzard challenge 2020," in *Proc. Joint Workshop for the Blizzard Challenge and Voice Conversion Challenge 2020*, 2020, pp. 1–18.

[7] I. Solak, "The M-AILABS speech dataset," https://www.caito.de/2019/01/the-m-ailabs-speech-dataset/, 2019.

[8] K. Ito and L. Johnson, "The lj speech dataset," https://keithito.com/LJ-Speech-Dataset/, 2017.

[9] P.-E. Honnet, A. Lazaridis, P. N. Garner, and J. Yamagishi, "The siwis french speech synthesis database. design and recording of a high quality french database for speech synthesis," Idiap, Tech. Rep., 2017.

[10] S. Karita, N. Chen, T. Hayashi, T. Hori, H. Inaguma, Z. Jiang, M. Someki, N. E. Y. Soplin, R. Yamamoto, X. Wang *et al.*, "A comparative study on transformer vs rnn in speech applications," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2019, pp. 449–456.

[11] S. Hochreiter, Y. Bengio, P. Frasconi, J. Schmidhuber *et al.*, "Gradient flow in recurrent nets: the difficulty of learning long-term dependencies," in *A field guide to dynamical recurrent neural networks*, Y. Hochreiter, Sepp Bengio, P. Frasconi, J. Kolen, and S. Kremer, Eds. IEEE Press, 2001.

[12] Y. Wang, D. Stanton, Y. Zhang, R.-S. Ryan, E. Battenberg, J. Shor, Y. Xiao, Y. Jia, F. Ren, and R. A. Saurous, "Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis," in *International Conference on Machine Learning*. PMLR, 2018, pp. 5180–5189.

[13] Y.-J. Zhang, S. Pan, L. He, and Z.-H. Ling, "Learning latent representations for style control and transfer in end-to-end speech synthesis," in *ICASSP*. IEEE, 2019, pp. 6945–6949.

[14] K. Kastner, J. F. Santos, Y. Bengio, and A. Courville, "Representation mixing for tts synthesis," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 5906–5910.

[15] J. Kearns, "Librivox: Free public domain audiobooks," *Reference Reviews*, 2014.

[16] I. BS, "1534-1, method for the subjective assessment of intermediate quality level of coding systems," *International Telecommunications Union, Geneva, Switzerland*, vol. 14, 2003.

[17] M. Shamsi, J. Chevelu, N. Barbot, and D. Lolive, "Corpus design for expressive speech: impact of the utterance length," in *Speech Prosody*. ISCA, 2020, pp. 955–959.

[18] H. Tachibana, K. Uenoyama, and S. Aihara, "Efficiently trainable text-to-speech system based on deep convolutional networks with guided attention," in *ICASSP*. IEEE, 2018, pp. 4784–4788.

[19] J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," *arXiv preprint arXiv:1506.07503*, 2015.

[20] C. Mayo, R. A. Clark, and S. King, "Multidimensional scaling of listener responses to synthetic speech," in *Interspeech*. ISCA, 2005, pp. 1725–1728.

[21] F. Hinterleitner, S. Möller, C. Norrenbrock, and U. Heute, "Perceptual quality dimensions of text-to-speech systems," in *Interspeech*, 2011.

[22] J. B. Kruskal, *Multidimensional scaling*. Sage, 1978, no. 11.

[23] P. Barbosa and G. Bailly, "Characterisation of rhythmic patterns for text-to-speech synthesis," *Speech Communication*, vol. 15, no. 1, pp. 127–137, 1994.

[24] G. Bailly and C. Gouvernayre, "Pauses and respiratory markers of the structure of book reading," in *Interspeech*, 2012, pp. 2218–2221.

[25] W. Ping, K. Peng, A. Gibiansky, S. O. Arik, A. Kannan, S. Narang, J. Raiman, and J. Miller, "Deep voice 3: Scaling text-to-speech with convolutional sequence learning," *arXiv preprint arXiv:1710.07654*, 2017.

[26] P. Govalkar, J. Fischer, F. Zalkow, and C. Dittmar, "A comparison of recent neural vocoders for speech signal reconstruction," in *Proc. 10th ISCA Speech Synthesis Workshop*, 2019, pp. 7–12.

[27] R. Kubichek, "Mel-cepstral distance measure for objective speech quality assessment," in *Pacific Rim Conference on Communications Computers and Signal Processing*, vol. 1. IEEE, 1993, pp. 125–128.

[28] E. Godde, G. Bailly, D. Escudero, M.-L. Bosse, and E. Gillet-Perret, "Evaluation of reading performance of primary school children: Objective measurements vs. subjective ratings," in *International workshop on child computer interaction (WOCCI)*, 2017.

[29] P. Boersma and D. Weenink, "Praat, a system for doing phonetics by computer," *Glot international*, vol. 5, pp. 341–345, 01 2001.

[30] M. Schoeffler, S. Bartoschek, F.-R. Stöter, M. Roess, S. Westphal, B. Edler, and J. Herre, "webmushra—a comprehensive framework for web-based listening tests," *Journal of Open Research Software*, vol. 6, no. 1, 2018.

[31] S. Palan and C. Schitter, "Prolific. ac—a subject pool for online experiments," *Journal of Behavioral and Experimental Finance*, vol. 17, pp. 22–27, 2018.

11th ISCA Speech Synthesis Workshop (SSW 11)
26-28 August 2021, Budapest, Hungary

Pathological voice adaptation with autoencoder-based voice conversion

Marc Illa[*1,2], *Bence Mark Halpern*[*1,3,4], *Rob van Son*[3,4], *Laureano Moro-Velázquez*[5], *Odette Scharenborg*[1]

[1]Multimedia Computing Group, Delft University of Technology, Delft, The Netherlands
[2]Universitat Politecnica de Catalunya, Barcelona, Spain
[3]University of Amsterdam, Amsterdam, The Netherlands
[4]Netherlands Cancer Institute, Amsterdam, The Netherlands
[5]Johns Hopkins University, Baltimore, USA

{m.illa,o.e.scharenborg}@tudelft.nl, {b.halpern,r.v.son}@nki.nl, laureano@jhu.edu

Abstract

In this paper, we propose a new approach to pathological speech synthesis. Instead of using healthy speech as a source, we customise an existing pathological speech sample to a new speaker's voice characteristics. This approach alleviates the evaluation problem one normally has when converting typical speech to pathological speech, as in our approach, the voice conversion (VC) model does not need to be optimised for speech degradation but only for the speaker change. This change in the optimisation ensures that any degradation found in naturalness is due to the conversion process and not due to the model exaggerating characteristics of a speech pathology. To show a proof of concept of this method, we convert dysarthric speech using the UASpeech database and an autoencoder-based VC technique. Subjective evaluation results show reasonable naturalness for high intelligibility dysarthric speakers, though lower intelligibility seems to introduce a marginal degradation in naturalness scores for mid and low intelligibility speakers compared to ground truth. Conversion of speaker characteristics for low and high intelligibility speakers is successful, but not for mid. Whether the differences in the results for the different intelligibility levels is due to the intelligibility levels or due to the speakers needs to be further investigated.

Index Terms: voice conversion, pathological speech, variational autoencoder

1. Introduction

Data-driven speech synthesis has recently been reaching new heights with the introduction of deep neural networks (DNNs). However, the success of these techniques is subject to high quality data and a large quantity of data, either of which is not available for many applications. Pathological speech synthesis, where the goal is to synthesise natural, but pathologically sounding samples, is such an application. Pathological speech synthesis has several motivations, the most notable being the data augmentation for automatic speech recognisers (ASRs), where the goal is to generate more data in order to improve recognition of pathological speech [1, 2, 3]. The second motivation for the development of pathological speech synthesis is that it could assist in informed decision making for the medical conditions at the root of the pathology. For instance, oral cancer surgery results in changes to a speaker's voice. The availability of a synthesis model that can generate how the voice could sound after surgery could help the patients and clinicians to make informed decisions about the surgery and alleviate the stress of the patients [4, 5].

While there are many speech synthesis techniques for typical speech, not many of these are applicable if we wish to synthesise highly natural pathological speech. Formant [6] and articulatory synthesis [7] are lacking in naturalness compared to DNN-based speech synthesis. Text-to-speech techniques (TTS) lack both linguistic resources (i.e a pronunciation lexicon) and the amount of data needed for these problems. The only promising method to synthesise pathological speech seems to be voice conversion (VC), which only needs a relatively small amount of data, compared to neural TTS.

However, synthesising pathological speech via VC is not without challenges. Existing pathological speech corpora [8, 9, 5, 10] provide healthy control speakers, but healthy speech recordings from the same pathological speaker are rarely available. This means that a successful pathological voice conversion system needs to learn conversion of both, the voice and pathological characteristics simultaneously, as suggested in previous work [4]. However, evaluation of such a setup is difficult. This is because the VC system is directly optimised for speech degradation in terms of the pathology, which would need the listeners (the evaluators of these systems) to be able to rate the success of generating the pathological characteristics and the synthetic/natural aspects of the speech separately. As we will show later in this paper, listeners struggle differentiating between speech severity and synthetic aspects of the speech. This can result in two, counter-intuitive scenarios from the viewpoint of typical VC: (1) a pathological VC system that is not able to properly capture the characteristics of the pathological speech could still receive better naturalness scores than the reference pathological speech; (2) Conversely, a VC system that is able to mimic the pathology, albeit exaggeratedly, could produce a naturalness score that is a lot lower than that of the reference.

Therefore, we propose a new approach where instead of using healthy speech as source for the VC, we use dysarthric speech, which is already pathological, and the VC system only has to customise it to a new (healthy/dysarthric) speaker's voice characteristics, i.e by using some representation of the speaker (speaker embedding). This synthesis approach alleviates the problem with naturalness ratings as the dysarthric-to-dysarthric VC is not optimised directly for speech degradation, therefore degradation is only due to the synthetic aspects compared to the source pathological utterance. Our first goal is to assess whether we can convert the voice characteristics of the pathological speakers in this setup in a natural way, while simultaneously assessing how natural real pathological speech is per-

[*]Equal contribution.

10.21437/SSW.2021-4

ceived.

In order to perform the VC, an autoencoder-based method will be used [11]. Autoencoder-based methods are of special interest in clinical scenarios as they are non-parallel, thus allow for incomplete data collection situations, while also being easier to train than GAN-based methods due to well-defined convergence criteria because they have only a single loss [12, 13, 14]. In this paper, we use HL-VQ-VAE-3 which is a type of variational autoencoder (VAE) using discrete representations. This hierarchical design has recently shown to give better results for VC [15] than the original VQ-VAE. Furthermore, by conditioning on speaker labels, the model allows to converting to/from multiple speakers within one single model.

An important additional goal of this work is to investigate whether standard VC techniques can be used for non-standard speech. It is well known from other domains of speech technology such as automatic speech recognition (ASR) that standard ASR systems perform poorly on atypical speech [16, 17, 18, 19, 20], making standard speech technology techniques less accessible to people with atypical speech. Our paper is thus also a preliminary investigation of a VQ-VAE-based VC technique's performance on converting a pathological source utterance instead of a typical utterance from a non-dysarthric speaker.

To summarise, in this paper we train a dysarthric-to-dysarthric VC system to answer the following research questions: (**RQ1**) *Can we convert the voice characteristics of a pathological speaker to another pathological speaker of the same severity with reasonable naturalness (where reasonable means comparable to non-parallel VC methods on typical speech)? In other words, is VC technology accessible to people with pathological speech?* (**RQ2**) *How does (real) pathological speech affect the mean opinion score (MOS)? In other words, what is the maximum attainable naturalness of synthetic pathological speech?*

Section 2 will start with the discussion of the used UASpeech dataset and the used VQ-VAE methods for the task, and finally concluded by the experimental design to test the approach. The perceptual evaluation results are presented in Section 3, followed by a discussion of the limitations of the proposed method, and further comments on the accessibility of VC to pathological speakers. Some of the samples are available at *https://pathologicalvc.github.io* .

2. Design and methods

2.1. Description of the dataset and preprocessing

In this study we use the UASpeech corpus [8], which contains isolated-word recordings of 15 speakers with dysarthria. These recordings consist of 449 words which are divided into 3 blocks of equal length (B1, B2 and B3). The speakers are divided into four groups based on their intelligibility: very low, low, mid and high, which correspond to 0-25%, 25-50%, 50-75% and 75-100% human transcription word error rate (WER) of the recordings, respectively. The transcriptions were done by 5 American English native speakers, who are non-expert listeners.

The vocoder used (see Section 2.2) is trained using the VCTK dataset [21], which contains speech of 108 native English speakers with different accents. The preprocessing consists of downsampling the tracks from 48 kHz to 24 kHz, which is done with librosa [22].

The UASpeech data is preprocessed following [2]: stationary noise is removed using Noisereduce [23] and the silence

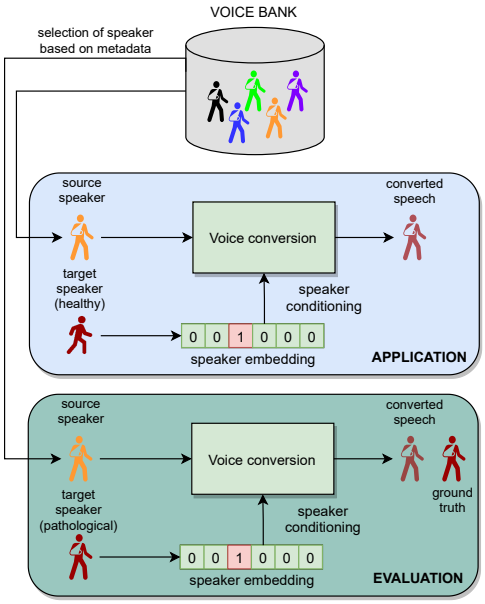


Figure 1: *Outline of our approach: the speech from a model pathological speaker is converted into speech with the characteristics of another pathological speaker. Red/orange colours denote the identity of the speaker. The figure is further explained in Section 2.*

from the beginning and end of the clips is cut. Then, the audio is resampled from 16 kHz to 24 kHz and normalised. Finally, 80-dimension mel-spectrograms (similar to [24]) are extracted from the audio files and used to compute the mel-cepstrum, which serves as input to our model.

2.2. Voice conversion model

The model is a 3-stage VQ-VAE. In the first stage, the input x to the model is a mel-cepstrum that goes through the convolutional encoder resulting in a hidden variable u_1 and a latent variable z_1. The second stage is identical to the first stage, except instead of x, now u_1 is fed into another convolutional encoder, resulting in u_2 and z_2. This is repeated for the third stage, feeding u_2 to obtain z_3 and u_3. This successive encoding serves to model the features in the speech that are present on successively longer temporal scales.

The variables z_n are all quantised using a nearest neighbour classifier with respect to the codebook's codewords of the corresponding stage. Then, we perform the decoding of the quantised variables q_n at each stage. The decoder is also convolutional which is additionally conditioned on a speaker label. During training, a speaker embedding table is learned from the training speakers, and during conversion/inference, this embedding will correspond to the target speaker of the conversion, which we can get by a table lookup. The decoding starts at the third stage and goes back to the first stage. The input of the third stage decoder is q_3 while for the second and first level the q_n signal is concatenated with the output v_n of the previous stage

(the output v_2 of the 3rd stage is fed to the 2nd and the output v_1 of the 2nd is fed to the 1st).

For the conversion, the trained model receives the input mel-cepstrum from a source speaker which is encoded and quantised in the same way as it is during training. Then, the speaker embedding is used to condition the decoder on a target speaker, so the source speaker quantised latent variables q_n are decoded conditioned on the target speaker embedding, which results in the decoded mel-cepstrum. Finally, the mel-cepstrum is resynthesised to the speech waveform using a Parallel WaveGAN vocoder[1] [25].

2.3. Details of the experimental design

As a reminder, in this study, we customise pathological speech to a different pathological speaker's voice characteristics. However, the clinical application would need customisation to a healthy speaker's characteristics. In the top panel of Figure 1, the application scenario is visualised, i.e., how the system could be used in a clinical setting. In the bottom part, our proposed evaluation scenario - the experiments that we do in the paper - is illustrated.

Looking at the top panel, a source pathological speaker is first selected from a large voice bank consisting of many samples of pathological speakers. Based on metadata, a clinical team could decide the kind of pathological speech degradation which is most likely for a patient. In this paper, we pair up by severity, but in actual practice an appropriate source speaker could be found matched by age, region, and type of treatment. This leads to a selection of a source pathological speaker. Using a small amount of a new patient's voice (target speaker), a speaker embedding can be extracted using the VQ-VAE based technique. Finally, we obtain the converted speech, which is expected to be pathological, but with the new patient's voice characteristics. The problem is that for the UASpeech, we don't have parallel pre-pathology and post-pathology voices. Therefore, a separate evaluation scheme has to be setup where we assume that the pathological and the healthy speaker embeddings should be unchanged for the same speaker, which is not always true, we refer to further discussion about this in Section 3.3.

The evaluation scheme is explained in the bottom panel. To circumvent the problem with the pre-pathology and post-pathology, we change the conversion process for the evaluation as follows. Instead of a new healthy speaker, we enroll a new dysarthric speaker with a matched intelligibility of the speech pathology from the UASpeech dataset because a ground truth (GT) is available there. The converted speech can then be compared to this GT to provide a proof of concept for the system.

Table 1: *Speaker pairs used for the VC experiments and their subjective WER differences.*

Speaker A (WER%)	Speaker B (WER%)	ΔWER (%)
M04 (2%)	M12 (7.4%)	5.4%
M05 (58%)	M11 (62%)	4%
M08 (93%)	M10 (93%)	0%

In our experiments, we convert the speech of three speaker pairs in both directions The setup for the experiments is the following. We train the VC model with all B1 and B3 sets of words of every dysarthric speaker to stay consistent with the standard UASpeech train-test partitioning.

[1]https://github.com/kan-bayashi/ParallelWaveGAN

We perform VC on the speech from B2 between speakers with a similar level of dysarthria. The selected dysarthric speaker pairs along with their corresponding human transcription error rates from UASpeech are summarised in Table 1. Unfortunately, it has not been possible to include females speakers because all female speakers had a different severity in the UASpeech dataset. We also refrained from controlling for the type of dysarthria in our experimental design, as that would have led to certain speaker pairs having excessive difference in their intelligibility, which would contrive the aim of the paper.

2.4. Subjective evaluation experiments

In order to answer our research questions, we performed subjective evaluation experiments. For RQ1 a subjective speaker similarity experiment was carried out, while for RQ2 a subjective naturalness experiment was carried out. The design of these experiments (including the composition of different stimuli) closely follow those of the VCC challenge standards [26, 27]. These experiments were run on the Qualtrics platform, and the participants (10 American English native listeners) were recruited through Prolific. All participants were remunerated justly (7.80 GBP per hour).

For the naturalness experiment, we used a mean opinion score (MOS) naturalness test. We hypothesised that listeners will not be able to distinguish between the distortions in the audio and the pathological characteristics of the speech. In order to account for this, we included GT stimuli in the naturalness test, which allows direct comparison of naturalness with real samples. The GT shows the maximum attainable naturalness (second part of RQ2) and the differences of the GT and VC scores show the reduction due to the synthetic aspects. To answer the first part of RQ2, we included healthy, natural stimuli, which allows us to measure the reduction in naturalness due to the reduction intelligibility. Nevertheless, we encouraged listeners to ignore the atypical aspects of the speech by adopting the naturalness question from the VCC2020 [26], which was proposed for cross-lingual VC, where pronunciation errors could appear, similar to pathological speech. For the speaker similarity test, we used an AB test in which listeners were asked to listen to two stimuli, and indicate if they thought they came from the same speaker, and rate their confidence in this decision. The question for the speaker similarity was directly adopted from the VCC2016 challenge [27].

3. Results and discussion

3.1. Naturalness

The results of the naturalness experiments are presented in Figure 2, which shows the MOS score for each of the seven types of speech tested, grouped by intelligibility, and with their 95% confidence intervals indicated. For clarity, the actual MOS scores are indicated on top of each bar.

We first focus on the question how GT pathological speech affects the naturalness perceived by listeners which is measured by the MOS score (our RQ2). Figure 2 shows that healthy speech and GT high intelligibility dysarthric speech have a similar MOS score. However, as intelligibility decreases, so does the MOS score, indicating that the MOS score not only captures naturalness but is influenced by the intelligibility of the speech. These results show that naive listeners cannot separate severity of a pathology and unnaturalness when asked to judge the naturalness of a speech sample. This also means that the GT MOS results are an upper bound on the achievable naturalness

of synthetic pathological samples.

Regarding the synthetic pathological speech, the performance on the high (VC) samples is somewhat lower than the performance of the HL-VQ-VAE-3 model on the VCC2020 challenge and identical to the performance of autoencoder-based models (2.1) [15]. However, the type of stimuli is different, so the differences in MOS are not directly comparable. The difference is most likely due to channel differences, the decreased intelligibility of the speech, and the different sampling frequency (UASpeech is 16 kHz, while VCC2020 is 24 kHz). When we compare the MOS scores for the converted speech of the different intelligibility speakers, we observe a slight degradation in naturalness with decreasing intelligibility. Comparing the VC and GT results, however, we observe a large degradation for the converted high intelligibility speech (Wilcoxon signed-rank test: $p \leq 0.05$). The difference in VC and GT MOS scores for the mid and low intelligibility speakers is much smaller (Wilcoxon signed-rank test: mid $p \leq 0.05$, low $p \geq 0.05$). It is possible that the standard 5-point MOS does not allow to express the nuances between mid and low samples appropriately. Therefore, for future studies concerning naturalness of pathological speech, we would recommend using a slightly wider, 7-point scale. Returning to RQ1, we can conclude that the synthetic speech of mid and low intelligibility pathological speakers have a naturalness that is perceived similar to that of real pathological speech, while synthetic high intelligibility pathological speech is not perceived as being as natural as real high intelligibility pathological speech.

To summarise, pathological speech is not perceived natural according to the MOS scale by naive listeners. In the case of mid and low intelligibility pathological speech, the perceived naturalness is similar between that of synthetic and real pathological speech. This is, however, not the case for high intelligibility synthesised pathological speech which is rated as being far less natural than real pathological speech. The performance of the VC approach is comparable to the one observed with typical speakers, therefore the current method is accessible to typical speakers, however this does not mean that VC is accessible to typical speakers (see Section 3.4).

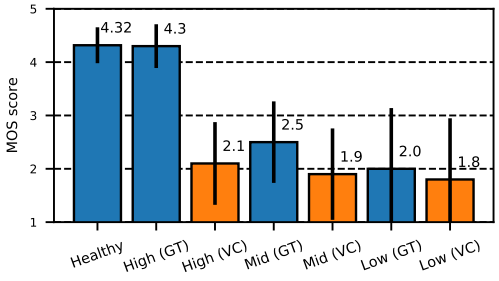


Figure 2: *Mean opinion scores for naturalness grouped by intelligibility with 95% confidence intervals. Blue denotes original, while orange denotes VC samples.*

3.2. Similarity

This section presents and discusses the results of the similarity experiments in order to answer the question whether it is possible to convert voice characteristics of pathological speakers. The results are presented in Figure 3. In each of the 12 panels, we visualise the results of comparing a voice converted (VC-D / VC-S) sample with the GT source (S) (Similarity to source) or the GT target (Similarity to target). Also, the GT samples are compared between them: S samples are compared to S samples to know how recognisable the source speaker is, T samples are compared to T samples to know how recognisable the target speaker is and S samples are compared to T samples in order to know how distinguishable is the source from the target speaker. Note that for each speaker pair in the top panel the source speaker is the target one in the bottom panel and vice versa, so this information appears repeated in Figure 3. Additionally letters in the case of the VC comparisons are used to help interpretation of the figures: VC-D stands for VC-different (i.e when converting M04 to M12, the converted should be different from M04), VC-S stands for VC-same (similarly, when converting M04 to M12, the converted should be same as M12).

For the low intelligibility pair (left 2 columns of Figure 3), the speakers seem reasonably distinguishable when looking at the GT as there is a 100% of agreement that M04 samples are produced by M04 and 90% for M12. For the speech samples of speaker M04 converted to speaker M12 (top panels), 73.33% of the converted samples were indicated as being from speaker M12 (VC-S), meaning that the conversion is fairly successful for that pair. For the speech samples of speaker M12 converted to speaker M04 (bottom panels), 56.33% of the converted M12-M04 samples (VC-S) were indicated as being from speaker M04. The results show that for the M12-M04 conversion the model is able to remove some of the source speaker (M12) characteristics and add some of the target (M04) ones, although to a lesser extent than in the M04-M12 conversion. Therefore, we conclude that the voice characteristic conversions for the low intelligibility speakers are successful.

For the mid intelligibility pair (middle four panels), the M11 seems to be clearly recognisable as there is a 90% of agreement that M11 samples are produced by M11, however listeners have difficulties recognising the voice characteristics of M05, i.e., only 20% of the trials where both samples were from speaker M05 were judged as both being from M05. For M05-M11 the VC performs poorly, which is indicated by 90% perceiving it different from the target (VC-S result). For M05-M11 the VC-S reaches a 20% of absolutely sure agreement. Notice that although it is a low score, it is the same that the GT samples exhibit. The voice characteristic conversions for the mid intelligibility speakers are thus inconclusive: while in one case the VC fails, in the other participants fail to recognise the speaker even from the GT samples. Further experimentation with more speaker pairs is needed.

For the high intelligibility pairs (right 2 columns of Figure 3), the speakers seem reasonably distinguishable. We can see that there is a 70% of agreement that M08 samples are produced by M08 and an 80% for M10. For M08-M10, there is a 46.66% of agreement that the converted samples sound like M10. For M10 to M08 VC, 75% of the listeners indicate that the converted samples sound like M08. We can see that some of the voice characteristics are successfully transferred for the high intelligibility samples, however while on the conversions M10 to M08 the result is similar to the GT samples, on the other direction (M08 to M10) there is a gap of 33.33% with respect to the GT. This behaviour is the same that we observed with low intelligibility pair conversions: although the speakers from the same pair are recognised with a similar agreement (100% and 90% for low intelligibility and 80% and 70% for the high intelligibility) the conversions are more successful in one direction than on the other.

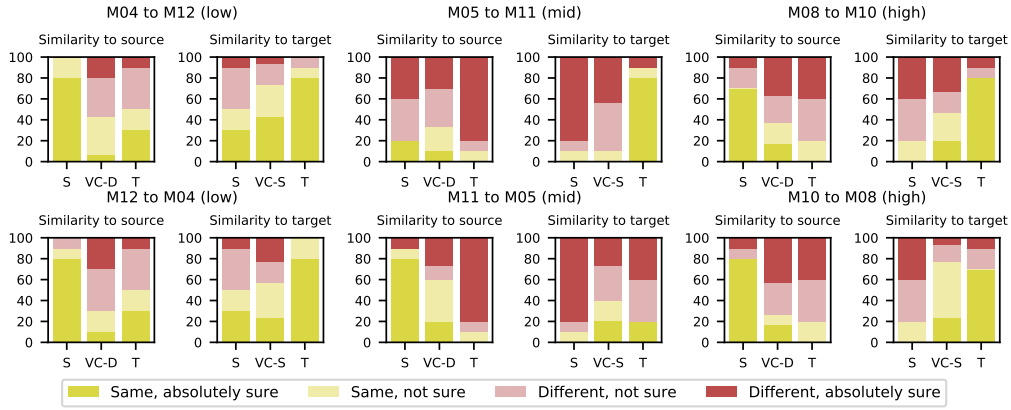


Figure 3: *Results of the speaker similarity experiments grouped by intelligibility pairs. S stands for source, T for target, VC-D for voice conversion different (VC samples should be different from source) and VC-S for voice conversion same (VC samples should be same as target).*

3.3. Limitations of the proposed approach

An assumption of the proposed approach is that the speaker identity is not affected by the speech pathology, which is certainly untrue for speech pathologies which are dysphonic, i.e. where the voice characteristics are known to be affected. By performing AB testing with GT speakers, we have tried to account for these scenarios in the perceptual evaluations. From the speaker similarity experiment, we have seen that in some cases (i.e., M05) listeners had difficulties of recognising the voice characteristics even in the GT. These results confirm that the proposed approach cannot be used for all types of speech pathologies. To solve this issue, we would need to have a deeper understanding of what happens to the speaker characteristics in these speech pathologies. For example, the speaker embeddings themselves could be used to predict the new pathological speaker embeddings of the same speaker, transformed according to the vocal pathology (i.e. type of dysphonia).

3.4. Accessibility of VC to atypical speakers

VC of atypical speech produced similar naturalness in the high intelligibility case as typical speech on VQ-VAE based methods. Nevertheless, we see that there is room for improvement compared to typical speech, as other studies employing certain non-parallel VC approaches can achieve human-like naturalness. Unfortunately, these VC approaches cannot easily be used for our task as they often leverage linguistic features or ASR bottleneck features [28, 29]. The need for ASR features is especially problematic as these features are extracted from ASR systems, whose performance on atypical speech is generally much worse than that on typical speech, meaning that the quality of these extracted features are also expected to be lower for these speakers. Therefore, we conclude that accessibility to VC is limited for atypical speakers, but this is because parallel and ASR-based techniques can hardly be used by them.

4. Conclusions

In this paper, we propose a new approach to pathological speech synthesis, by customising an existing pathological speech sample to a new speaker's voice characteristics. In order to do this pathological-to-pathological speech conversion, we use an autoencoder-based voice conversion (VC) technique. When comparing our results with the ones obtained in the VCC2020 challenge dataset [15], we can see that ours are somewhat lower, which is most likely due to channel differences, the decrease in the speech intelligibility and the different sampling rate. We find that even real pathological speech seems to affect perceived naturalness as shown by MOS scores, meaning that there is a bound on achievable naturalness for pathological speech conversion. Overall, we observe a decreasing trend in MOS with decreasing intelligibility. Therefore, for low and mid intelligibility, the difference in perceived naturalness between real and VC is small. Conversion of voice characteristics for low intelligibility speakers is successful, for high intelligibility it is also possible to transfer the voice characteristics partially. However, more experimentation is needed for the mid intelligibility with more speakers: we experienced that in one case the VC failed, and on the other participants fail to recognise the speaker even from the real recordings. Whether the differences in the results for the different intelligibility levels is due to the intelligibility levels or due to other speech characteristics needs to be further investigated. The question of pathological intergender (male to female) and female VC also needs to be investigated. The performance of the approach is comparable to the one observed with typical speakers, therefore the current method is accessible to atypical speakers. However, in the paper, we outlined some issues such as the need for linguistic resources and parallel data, as an obstacle for more natural VC for pathological speakers.

5. Acknowledgements

B.M.H. is funded through the EU's H2020 research and innovation programme under MSC grant agreement No 766287. The Department of Head and Neck Oncology and Surgery of the Netherlands Cancer Institute receives a research grant from Atos Medical (Hörby, Sweden), which contributes to the existing infrastructure for quality of life research.

6. References

[1] B. Vachhani, C. Bhat, and S. K. Kopparapu, "Data augmentation using healthy speech for dysarthric speech recognition." in *Interspeech*, 2018, pp. 471–475.

[2] J. Harvill, D. Issa, M. Hasegawa-Johnson, and C. Yoo, "Synthesis of new words for improved dysarthric speech recognition on an expanded vocabulary," in *International Conference on Acoustics, Speech and Signal Processing*, 2021.

[3] Y. Jiao, M. Tu, V. Berisha, and J. Liss, "Simulating dysarthric speech for training data augmentation in clinical speech applications," in *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2018, pp. 6009–6013.

[4] B. M. Halpern, J. Fritsch, E. Hermann, R. van Son, O. Scharenborg, and M.-M. Doss, "An objective evaluation framework for pathological speech synthesis," *Submitted to Signal Processing Letters*, 2021.

[5] B. M. Halpern, R. van Son, M. van den Brekel, and O. Scharenborg, "Detecting and Analysing Spontaneous Oral Cancer Speech in the Wild," in *Proc. Interspeech 2020*, 2020, pp. 4826–4830. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2020-1598

[6] F. Rudzicz, "Adjusting dysarthric speech signals to be more intelligible," *Computer Speech & Language*, vol. 27, no. 6, pp. 1163–1177, 2013.

[7] S. Aryal and R. Gutierrez-Osuna, "Data driven articulatory synthesis with deep neural networks," *Computer Speech & Language*, vol. 36, pp. 260–273, 2016.

[8] H. Kim, M. Hasegawa-Johnson, A. Perlman, J. Gunderson, T. S. Huang, K. Watkin, and S. Frame, "Dysarthric speech database for universal access research," in *Ninth Annual Conference of the International Speech Communication Association*, 2008.

[9] F. Rudzicz, A. K. Namasivayam, and T. Wolff, "The TORGO database of acoustic and articulatory speech from speakers with dysarthria," *Language Resources and Evaluation*, vol. 46, no. 4, pp. 523–541, 2012.

[10] C. Middag, "Automatic analysis of pathological speech," Ph.D. dissertation, Ghent University, 2012.

[11] W.-N. Hsu, Y. Zhang, and J. Glass, "Learning latent representations for speech generation and transformation," in *Proc. Interspeech 2017*, 2017, pp. 1273–1277. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2017-349

[12] T. Kaneko and H. Kameoka, "Cyclegan-vc: Non-parallel voice conversion using cycle-consistent adversarial networks," in *2018 26th European Signal Processing Conference (EUSIPCO)*. IEEE, 2018, pp. 2100–2104.

[13] T. Kaneko, H. Kameoka, K. Tanaka, and N. Hojo, "CycleGAN-VC2: Improved CycleGAN-based non-parallel voice conversion," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6820–6824.

[14] Kaneko, Takuhiro and Kameoka, Hirokazu and Tanaka, Kou and Hojo, Nobukatsu, "CycleGAN-VC3: Examining and Improving CycleGAN-VCs for Mel-Spectrogram Conversion," in *Proc. Interspeech 2020*, 2020, pp. 2017–2021. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2020-2280

[15] T. V. Ho and M. Akagi, "Non-parallel voice conversion based on hierarchical latent embedding vector quantized variational autoencoder," in *Proc. Joint Workshop for the Blizzard Challenge and Voice Conversion Challenge 2020*, 2020, pp. 140–144.

[16] S. Feng, O. Kudina, B. M. Halpern, and O. Scharenborg, "Quantifying bias in automatic speech recognition," *arXiv preprint arXiv:2103.15122*, 2021.

[17] M. Adda-Decker and L. Lamel, "Do speech recognizers prefer female speakers?" in *Ninth European Conference on Speech Communication and Technology*, 2005.

[18] L. Moro-Velazquez, J. Cho, S. Watanabe, M. A. Hasegawa-Johnson, O. Scharenborg, H. Kim, and N. Dehak, "Study of the Performance of Automatic Speech Recognition Systems in Speakers with Parkinson's Disease," in *Proc. Interspeech 2019*, 2019, pp. 3875–3879. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2019-2993

[19] A. Koenecke, A. Nam, E. Lake, J. Nudell, M. Quartey, Z. Mengesha, C. Toups, J. R. Rickford, D. Jurafsky, and S. Goel, "Racial disparities in automated speech recognition," *Proceedings of the National Academy of Sciences*, vol. 117, no. 14, pp. 7684–7689, 2020.

[20] E. Hermann and M. M. Doss, "Dysarthric speech recognition with lattice-free MMI," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6109–6113.

[21] C. Veaux, J. Yamagishi, K. MacDonald *et al.*, "CSTR VCTK Corpus: English Multi-speaker Corpus for CSTR Voice Cloning Toolkit (2017)," *URL http://dx. doi. org/10.7488/ds*, 2017.

[22] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "librosa: Audio and music signal analysis in python," 2015.

[23] T. Sainburg, "timsainb/noisereduce: v1.0," Jun. 2019. [Online]. Available: https://doi.org/10.5281/zenodo.3243139

[24] R. Skerry-Ryan, E. Battenberg, Y. Xiao, Y. Wang, D. Stanton, J. Shor, R. Weiss, R. Clark, and R. A. Saurous, "Towards end-to-end prosody transfer for expressive speech synthesis with Tacotron," in *international conference on machine learning*. PMLR, 2018, pp. 4693–4702.

[25] R. Yamamoto, E. Song, and J.-M. Kim, "Parallel WaveGAN: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6199–6203.

[26] Z. Yi, W.-C. Huang, X. Tian, J. Yamagishi, R. K. Das, T. Kinnunen, Z. Ling, and T. Toda, "Voice conversion challenge 2020—intra-lingual semi-parallel and cross-lingual voice conversion—," in *Proc. Joint Workshop for the Blizzard Challenge and Voice Conversion Challenge 2020*, 2020, pp. 80–98.

[27] T. Toda, L.-H. Chen, D. Saito, F. Villavicencio, M. Wester, Z. Wu, and J. Yamagishi, "The voice conversion challenge 2016." in *Interspeech*, 2016, pp. 1632–1636.

[28] L.-J. Liu, Z.-H. Ling, Y. Jiang, M. Zhou, and L.-R. Dai, "WaveNet Vocoder with Limited Training Data for Voice Conversion," in *Proc. Interspeech 2018*, 2018, pp. 1983–1987. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2018-1190

[29] X. Tian, J. Wang, H. Xu, E. S. Chng, and H. Li, "Average Modeling Approach to Voice Conversion with Non-Parallel Data." in *Odyssey*, vol. 2018, 2018, pp. 227–232.

11th ISCA Speech Synthesis Workshop (SSW 11)
26-28 August 2021, Budapest, Hungary

Location, Location: Enhancing the Evaluation of Text-to-Speech synthesis using the Rapid Prosody Transcription Paradigm

Elijah Gutierrez[1], *Pilar Oplustil-Gallegos*[2], *Catherine Lai*[1,2]

[1]Linguistics and English Language, University of Edinburgh, United Kingdom
[2]The Centre for Speech Technology Research, University of Edinburgh, United Kingdom
s1740779@sms.ed.ac.uk, p.s.oplustil-gallegos@sms.ed.ac.uk, c.lai@ed.ac.uk

Abstract

Text-to-Speech synthesis systems are generally evaluated using Mean Opinion Score (MOS) tests, where listeners score samples of synthetic speech on a Likert scale. A major drawback of MOS tests is that they only offer a general measure of overall quality—i.e., the naturalness of an utterance—and so cannot tell us where exactly synthesis errors occur. This can make evaluation of the appropriateness of prosodic variation within utterances inconclusive. To address this, we propose a novel evaluation method based on the Rapid Prosody Transcription paradigm. This allows listeners to mark the locations of errors in an utterance in real-time, providing a probabilistic representation of the perceptual errors that occur in the synthetic signal. We conduct experiments that confirm that the fine-grained evaluation can be mapped to system rankings of standard MOS tests, but the error marking gives a much more comprehensive assessment of synthesized prosody. In particular, for standard audiobook test set samples, we see that error marks consistently cluster around words at major prosodic boundaries indicated by punctuation. However, for question-answer based stimuli, where we control information structure, we see differences emerge in the ability of neural TTS systems to generate context-appropriate prosodic prominence.

Index Terms: Speech Synthesis, TTS, TTS Evaluation, MOS, Prosody, Rapid Prosody Transcription, Speech Perception

1. Introduction

Modern text-to-speech (TTS) systems have attained a level of naturalness that is approaching human parity for isolated utterances [1]. This progress is in large part due to the rise of neural network based machine learning methods, which have drastically improved the overall quality of synthetic speech and enabled researchers to focus more attention on generating natural sounding prosodic variation. In recent years, there has been substantial research on achieving fine-grained control over synthetic prosody [2, 3, 4]. New prosodic control mechanisms have allowed TTS systems to produce more variable and expressive speech [5]. However, there has been relatively little work determining whether the prosody that is assigned to an utterance is actually licensed by a given context [6, 7], and it is not clear whether current subjective evaluation methods, such as Mean Opinion Score (MOS) tests, provide enough information to determine the contextual appropriateness [8].

The appropriatess of utterance prosody–which broadly includes pitch, energy, timing and other suprasegmental characteristics of speech–can vary greatly depending on context. In fact, prosodic differences can help disambiguate many aspects of discourse and dialogue structure [9, 10, 11, 12, 13]. Many studies have also shown the close relationship between context-induced expectations about the prosodic form of utterances and information structural notions like newness and givenness [14, 15, 16]. Incorporating discourse relations has been shown to improve the perceived naturalness of synthesized speech [17], while incorporating information structure into generated speech has been shown to improve naturalness of automated task oriented dialogues [18]. As neural TTS models continue to improve in their ability to generate variable prosody, it is important to note that not all variation is appropriate in all contexts and increased variation within an utterance is not always perceived as natural [2].

In order to evalute how and where TTS systems are really improving in terms of prosody, we need methods that give us a clearer view of what sort of prosodic patterns they generate, and how their appropriateness changes with context. To do this, we propose a new evaluation method that augments traditional MOS-based listening tests with finer-grained error annotations. Specifically, we draw on the Rapid Prosody Transcription (RPT) framework [19, 20] to obtain information about the location of perceived errors in the prosody of synthesized speech. In RPT, non-expert listeners mark the presence of prosodic phenomena (e.g. prominence or boundary placement) in real time. This approach allows us to more precisely identify contextual/linguistic sources of prosodic errors.

Much of the current work on TTS in context has focused on monologue or narrative style generation, where information structural relationships are generally unclear [6, 7], and prosodic expectations may not be strong. To address this, we created a schema for generating question-answer pairs with well defined information structure, which in turn project clear prosodic expectations for synthesized answers. Combined with word-level error annotation, this allows us to identify cases of contextually inappropriate prosodic variation.

In the following, we show that there is a strong negative correlation between measures based on error marking and MOS, from which we can recreate MOS based system rankings. Moreover, our question-answer stimuli can be used to induce stronger expectations about prosody than classic audiobook style test utterances, and so better highlights differences in the system prosodies. In general, inspection of the distibution of errors across systems for specific stimuli can lead to better understanding of the sources of system differences, which may otherwise be obscured by MOS alone.

2. Background

TTS researchers have developed a wide range of methods to evaluate the quality of synthetic speech [8]. However, subjective methods are still considered to be the gold standard in TTS evaluation. These generally involve asking listeners to rate speech samples on a specified dimension, usually naturalness

10.21437/SSW.2021-5

(i.e., how 'humanlike' synthesized speech sounds). The most commonly used subjective evaluation type is the Mean Opinion Score (MOS) test: listeners are presented with a synthetic stimulus and asked about their overall impression of it, scoring the stimulus on a (usually 5-point) Likert scale [21].

Some of the advantages of MOS tests are that they are straightforward to set up, they are quicker and less cognitively taxing than ranking tasks like MUSHRA, and MOS test design choices have been well extensively investigated [6, 22]. Recent work has expanded to evaluation beyond the single sentence [6, 7]. However, these studies generally focus on holistic evaluations over multi-utterances segments, rather than the parts of utterances that may change listener perception. Meanwhile, the relationship between linguistic context and sub-utterance prosody has been extensively studied, particularly in English, in terms of information structure [15, 16, 23], i.e. how information is organised in an utterance. This is usually cast in terms of given/new information (similarly topic/focus). These constraints are usually demonstrated using question-answer constructions: For example, 'ALEX ate the brownies' is an appropriate answer to 'Who ate the brownies?' because 'Alex' is the new information, while 'Alex ate the BROWNIES' is infelicitous because 'brownies' is contextually given. Thus, use of stimuli with clear information structure provides a precise way of probing whether prosody is context appropriate or not.

Though Information Structure theory can give us an idea of prosodic expectations, prosody perception is known for high inter-listener variability [19, 20] even with expert training [24]. So, the fact that the RPT framework was specifically developed to capture variability in prosody perception from non-expert listeners makes it a natural choice for exploring the perception of synthesized speech. In RPT, the perceptual salience of a prosodic features (e.g. prominence) is determined by the number of listeners who mark a specific segment (e.g. word) with that feature. Because RPT is a task that is conducted in real-time, responses are more sensitive to subtle local changes in quality than offline ones [21, 25].

While RPT has been extensively used to study perception of prosody in human speech [26], there has been little empirical work investigating within utterance prosody for TTS. The closest related work is Edlund et al.'s Audience Response System [27], where a group of listeners judged a synthetic sample simultaneously, pressing a button whenever they hear 'oddities'. This was used to identify common error regions and find the average response latency of listeners when marking errors. However, the definition of a perceptual error was left (deliberately) unclear. Edlund et al. used a single long-form stimulus of about 3 minutes in length. In contrast, our study compares different types of stimuli across multiple TTS systems and focuses on prosody.

3. Experimental Setup

3.1. Experiments and Hypotheses

We perform three listening tests to probe the usefulness of our proposed evaluation method. Experiment condition 1 (E1) is a standard MOS test, while Experiment condition 2 (E2) is a MOS test augmented with the RPT-based error marking task. We compare the results of these two tests to see whether orienting the evaluation to prosody and adding the error marking task affects MOS results.

In E1 and E2 listeners rate single utterances taken from the widely used LibriTTS test set [28]. In Experiment condition 3 (E3), listeners completed the augmented MOS test on question-answer stimuli designed to evoke specific information structural expectations. The goal here was to determine if the error marking would bring out listener expectations about utterance prosody, and hence allow us to distinguish between the appropriateness of prosodic renditions more precisely. This also allows us compare the types of error marks between the two types of test data (audiobook vs dialogue).

3.2. TTS systems

In each of the 3 listening tests, we compare three TTS systems: the Festival [29], Ophelia [30], and FastPitch [31].

Festival is a standard toolkit for building synthetic voices with unit selection. For these experiments, the 'SLT' voice distributed by FestVox was used, i.e. a female voice with a General American accent, built from the Arctic A corpus. We note that this voice is far from the current state-of-the-art in TTS, and so we use it as a baseline to see if listeners would ignore other signal naturalness issues when asked to attend to prosodic errors.

We use two neural TTS models as representative of the current state of the art in TTS. These were both trained on the Linda Johnson (LJ) Speech dataset [32], which consists of 13,100 recorded utterances from 7 non-fiction books. *Ophelia* models were trained using the default recipe (500 epochs for Text2Mel, 250 epoch for SSRN). *FastPitch* stimuli were synthesised using character (rather than phone) inputs via a pre-trained sequence-to-sequence model that was trained for 1000 epochs.

3.3. Stimuli

For E1 and E2, 30 sentences were sampled randomly from the evaluation set of the LibriTTS corpus [28], a popular audiobook corpus specially designed for TTS research. The maximum stimulus length was controlled to be 15 words to mitigate listener boredom and fatigue.

For E3, contexts and stimuli were generated in a similar manner to those used by [16] for their study of the acoustic correlates of information structure. We used a template-based approach, involving simple *Subject Verb Object* sentences, generating two types of question-answer pairs:

- Informational Focus: SVO
  e.g., Q: *What did Mary eat?*
  A: *Mary ate the cake.*
- Corrective Focus: No, SVO
  e.g. Q: *Did Mary buy the cookies?*
  A: *No, John bought the cookies.*

Questions were generated to change which constituent represented the new information/correction in the answer, which in English determines the appropriate prominence placement in the response stimuli. We created 10 stimuli per prominence position. Since there were two stimulus structures, this resulted in 10×3×2=60 stimuli in total.

3.4. Evaluation Tasks

The experiments were designed and distributed remotely using a customized version of the Language Markup and Experimental Design Software (LMEDS) [33]. Each stimulus was presented on its own page as follows.

For the standard MOS test (E1), a transcript of the audio stimulus was presented with a 'Play' button. Participants were asked to answer the question *'How natural does the speaker sound?'* on a 5-point Likert scale via a scale slider (MOS).

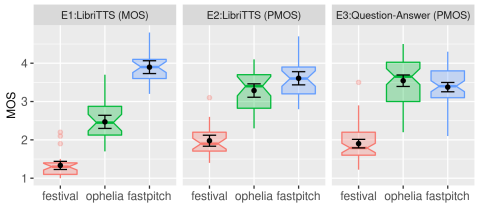


Figure 1: *Distribution of Mean Opinion Scores per experiment (boxplots and means with 95% confidence intervals in black).*

The augmented MOS tests (E2, E3) included the additional RPT-based error marking task, the MOS slider, and a further error type survey. The error marking task appeared first: participants were asked to listen to the stimulus and to click on any words in the transcript where the intonation did not sound correct (possibly none), highlighting them in red. For E3, participants were told to read the context question before marking errors on the answer stimulus. Participants were allowed to replay the stimulus up to 3 times and change their error marks. The MOS slider was positioned after the error marking task, rating *'How natural is the speaker's intonation?'* on a 5-point scale (PMOS). Finally, participants were asked to select which error types they noticed out of: 'Abrupt change in pitch', 'Awkward pause', 'Unexpected intonation' and 'Lacking intonation'. These choices were based on our initial impressions of potentially common errors. Participants also had access to an 'Other' box to enter additional comments or a custom response. In E2 and E3, participants were initially shown 3 examples of stimuli with prosodic errors along with an explanations of why they were considered odd or unnatural. Once this familiarisation phase was complete, participants moved on to the main task. In all three experiments, the audio stimuli were presented in a random order.

3.5. Participants and Groups

English-speaking participants were recruited with the crowd sourcing platform Prolific Academic.[1] Each participant was paid £2 for their participation in the study.

Participants were assigned a random group via Prolific and directed to a listening test based on a Latin square design for each evaluation condition (E1, E2: 3 groups of 10, E3: 6 groups of 10). Participants evaluated stimuli from every system, but did not evaluate the same text stimulus more than once.

After consenting to participate in the study, participants were instructed to wear headphones for best audio quality, to ensure they had a stable connection to the server, and to focus their attention on the evaluation task. A brief explanation of what was meant by intonation was also given for E2 and E3. Each participant rated 30 stimuli via the LMEDS interface described above. The standard MOS (E1) test took 8 minutes to complete on average, while the augmented MOS tests (E2, E3) took 15 minutes.[2]

4. Results

Figure 1 shows the distribution of mean Likert scale ratings per stimuli for the three experimental conditions. For E1 this is the classic 'naturalness' (MOS), while for E2 and E3 this is a prosodic naturalness rating (PMOS). Comparing the results for E1 and E2 (LibriTTS), we see that the MOS and PMOS scores show the same overall ranking of systems. However, the absolute difference between the system means is reduced in E2, with a marked increase for the scores for Festival and Ophelia. Table 1 shows the means and Interquartile ranges (IQR) for the 3 tests (IQR is reported as a measure of dispersion for consistency instead of standard deviation as system distributions were skewed). The overall mean MOS is significantly different for all systems in E1 (paired t-test, $p < 0.01$ with Bonferroni correction), resulting in the ranking Fastpitch > Ophelia > Festival. However, in the PMOS conditions (E2, E3), the difference between FastPitch and Ophelia is no longer significant at the same level (i.e, $p > 0.01$). Ratings of Ophelia-produced stimuli were the most variable for all conditions, with the greatest dispersion shown for the question-answer condition.

Table 1: *Mean / IQR per stimulus mean MOS*

System	E1 (MOS)	E2 (PMOS)	E3 (PMOS)
Festival	1.33 / 0.30	1.98 / 0.49	1.90 / 0.60
Ophelia	2.47 / *0.75*	3.29 / *0.88*	**3.54** / *1.03*
FastPitch	**3.90** / 0.50	**3.61** / 0.70	3.38 / 0.70

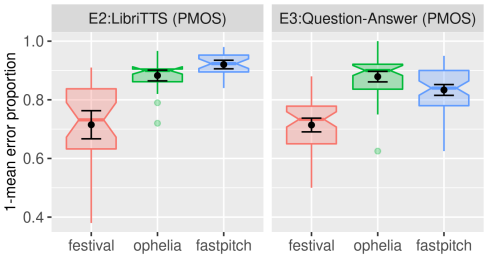


Figure 2: *Distribution of mean error rates (per stimulus).*

These distributional differences indicate that shifting participants focus to prosodic errors changed how they rated the stimuli. This also suggests that lower ratings for Festival and Ophelia in E1 were due to non-prosodic issues. Conversely, the higher ratings for FastPitch are for overall better synthesis quality, but not necessarily for more natural prosodic realization. As we shift to test stimuli with clearer prosodic expectations, the gap between systems in terms of prosodic naturalness is reduced and sometimes reversed relative to what we'd expect given only a standard MOS naturalness test.

To see how the error marking task relates to PMOS, we calculated the error marking rate (number errors/number of words) per stimuli and participant. Figure 2 shows the distribution of the mean error rate per stimuli (shown as 1-mean error rate to mirror PMOS ranking). We see that the overall system rankings are the same as that shown in Figure 1 for PMOS. Unsurprisingly, the correlation between stimulus PMOS and error rate is strongly negative when we pool data across all conditions (Pearson's $R = -0.75$). All differences in mean error rate are significant (paired t-tests, $p < 0.01$, Bonferroni correction) except between Ophelia and FastPitch in E2, i.e. when we look at the word level errors in for question-answer stimuli Ophelia performs significantly better than FastPitch. This indicates that the fine-grained evaluation has better ability to differentiate the

[1] https://www.prolific.co

[2] Further details/stimuli: http://sweb.inf.ed.ac.uk/clai/tts-rpt

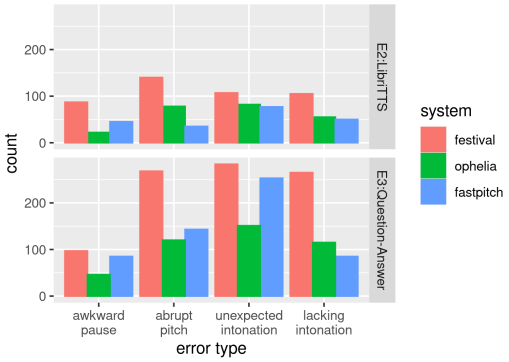


Figure 3: *Counts of error types per system for E2, E3.*

Test set	system	α	α_p	N_p
LibriTTS	festival	0.12	0.18	7.67
	ophelia	0.16	0.26	5.60
	fastpitch	**0.24**	**0.28**	**4.90**
Question-Answer	festival	0.10	**0.24**	6.53
	ophelia	**0.28**	0.18	**3.90**
	fastpitch	0.18	**0.24**	5.20

Table 2: *Mean interannotator agreement: Krippendorff's α, Krippendorff's α restricted to participants that marked at least one error in the stimulus (α_p), the number of participants who marked an error in a stimulus (N_p, max 10).*

system prosody when prosodic expectations are designed to be stronger (E3), but this difference may not be apparent for classic narrative style test sets.

Figure 3 shows the distribution of error types selected per experimental condition. This again supports the idea that prosodic expectations had a larger role when evaluating question-answer pairs. Overall, we see a lower number of error types selected for the LibriTTS set than the question-answer set. In particular, participants seemed less likely to detect abrupt pitch changes in FastPitch compared to Ophelia in the LibriTTS stimuli. However, we see a marked increase in unexpected intonation errors for the FastPitch question-answer set. In contrast, FastPitch garnered slightly less 'lacking intonation' errors than Ophelia for the question-answer stimuli. This suggests that issues with FastPitch came from an excess of prosodic variation, which was more salient for the question-answer set.

We originally expected that the question-answer pairs would lead to greater interannotator agreement on error locations compared to the LibriTTS data due to stronger prosodic expectations. To investigate this, Krippendorff's α [34] was calculated across the annotations for each stimulus in E2 and E3. We used the coding error=1, no error=0 for the annotation. Since participants could mark no errors in a stimulus, we added an additional 'word' to each annotation marked 1 if the participant marked no other errors, and 0 otherwise. This ensured that 'no error' annotations would be counted as agreeing. To see more clearly if errors tended to be marked on the same words, we also calculated agreement per stimulus discarding annotations with no error marks (α_p). We also count the number of participants who marked any error in a stimulus (N_p).

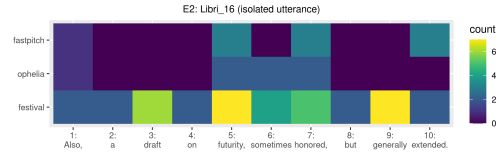


Figure 4: *Error Heatmap (Libri16), PMOS: Festival=1.7, Ophelia=3.6, Fastpitch=2.90.*

System	E2: LibriTTS	E3: Question-Answer
Festival	0.50	0.52
Ophelia	0.67	0.45
FastPitch	0.73	0.60

Table 3: *Proportion of time the most error marked word in a stimulus preceded punctuation. Note, LibriTTS includes much more within utterances punctuation and punctuation variation than the Question-Answer set.*

Agreement statistics are shown in Table 2. The LibriTTS results were as expected: as the lowest quality system, Festival, displays the lowest inter-annotator agreement, while Ophelia and FastPitch exhibit greater agreement for both α and α_p. The mean values for N_p also align with the PMOS ranking. For the question-answer test set, α and N_p also reflects the PMOS ranking. However, α_p, is higher for Festival and FastPitch, indicating that, while there was less agreement on whether there was an error in a stimuli: when participants marked an error they were more likely to choose the same word in the FastPitch and Festival cases. However, we note that both types of α value are still in the low agreement range, so other types of errors likely came into play (cf. Figure 3).

A benefit of the error annotation is that we can visualize the distribution of errors across systems to direct further investigation. For example, Figure 4 shows the error heatmap for a LibriTTS stimulus where FastPitch was rated lower than Ophelia in PMOS. This shows that error markings for FastPitch tended to occur on words attached to punctuation marks. To check whether this occurred more generally, we calculated the proportion of times that the most error marked word per stimulus preceded punctuation. The results in Table 3 indicate that punctuation was a more salient issue for FastPitch than for Ophelia.

Figure 5 shows F0 contours corresponding to the heatmap in Figure 4. Out of the 11 error types checked for the FastPitch version, 5 were for 'awkward pause', 2 for 'abrupt pitch' and 4 for 'unexpected prosody', while for Ophelia 3/5 votes were for 'lacking intonation'. On the FastPitch version we observe unexpected H* like pitch accents on 'honoured,' and 'extended.', while the Ophelia rendition has a continuation rise on 'honoured,' and a fall to low pitch through 'extended'. This supports the idea that punctuation produces specific prosodic expectations which were violated by the high level of prosodic variability (i.e., expressiveness) of FastPitch.

Similarly, Figure 6 shows error distributions for a contrastive focus example. Figure 7 indicates the error marks on 'cupcakes' in the FastPitch version are due to an unexpected pitch accent: 'cupcakes' is given relative to the context question and so should be deaccented. Interestingly, pitch tracking for the Ophelia version fails on 'cupcakes' due to issues in the signal quality, resulting in creaky-sounding (i.e., low pitched)

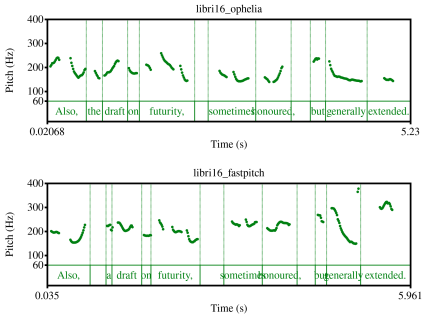


Figure 5: *F0 differences for Libri16: FastPitch has unexpected pitch accents on before punctuation.*

Figure 6: *Error Heatmap (focus16); PMOS: Festival=1.5 Fastpitch=3.2 Ophelia=3.0*

voice. This is in line with information structure expectations but still may have reduced overall stimuli PMOS. We can also see that the large pitch excursion on the FastPitch 'No' was perceived as an error. While this doesn't produce an information structural clash, it does present an unexpected level of emphasis without further contextual information to justify it.

5. Discussion

Our results indicate that error markings are consistent with rankings from MOS tests. However, differences between systems changed when participants were primed to focus on prosodic issues rather than naturalness in general. This means that the large lead FastPitch had over Ophelia in the naturalness (E1) is likely due to improvements in speech quality separate to prosody. It appears participants did separate out prosody and other quality issues, even for Festival which exhibited much lower naturalness than our neural TTS models. In fact, the error rate measure was better than PMOS at discriminating system prosody when combined with the question-answer test set, where prosodic expectations are more constrained.

It's important to note that neither FastPitch or Ophelia take into account preceding context in their generation processes. The lower ranking of FastPitch in the question-answer test is likely due to overly-variable (unexpected) prosody, rather than Ophelia being intrinsically better in context. MOS scores for Ophelia were generally more variable, especially in E3. So, it is likely that Ophelia generates a more typical 'reading style' intonation, which works well for some question-answer pairs, but not for others.

Default 'reading' intonation can work well for narrative-style (e.g., LibriTTS), but can be problematic when prosodic expectations are stronger, such as in task-oriented dialogues. This motivates more design and use of context sensitive stimuli, where factors contributing to prosodic expectations are well understood. A key factor for English prosody is information structure, but other factors will likely be important for other languages. In general, there are many paths which might lead listeners to give similar a (P)MOS, especially if the stimuli contains other types of errors (cf. relatively low interannotator agreement). This indicates that simply asking more specific MOS questions isn't enough to pinpoint differences in systems. The results of the current study support the case for more fine-grained error analysis.

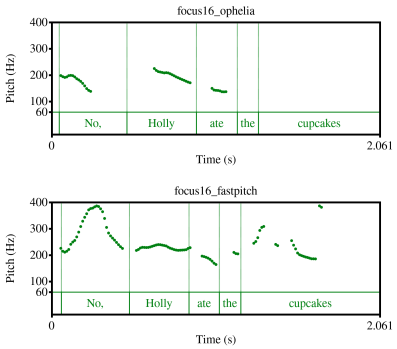


Figure 7: *F0 differences (focus16): FastPitch produces an extra prominence on 'cupcakes' (cf. Figure 6)*

The real-time marking used in this study can help TTS researchers and designers understand what non-expert listeners pay attention to when they evaluate and perceive synthetic speech. For example, our study suggests that listeners have specific expectations about what should happen around prosodic boundaries signalled by punctuation. Similarly, the results from the question-answer testset provide evidence for an expectation-driven view of prosody perception [16, 23]. Further analysis of the acoustic properties around error marks will help improve our understanding of these expectations in future work. Similarly, this method may shed light on cases where additional context actually allows for greater prosodic variability than in the isolated case [6].

While we have not done a comprehensive usability study of this method, many participants reported in the post-survey feedback form that they didn't find the experiment tiring and were even entertained by the experiment. This feedback suggests that the methodology is feasible and may help mitigate loss of attention in evaluating long-form TTS [35]. The fact that non-expert listeners can be used for the evaluation means that the methodology is scalable and gives a more realistic account of how a synthetic voice is perceived than a method using expert prosodic labelling.

6. Conclusion

This study introduced a novel evaluation paradigm that augments the standard MOS test with an RPT-based error marking task. Our experiments showed how this fine-grained error marking can uncover differences in systems in prosody generation. We confirmed that our error marking method can be used to distinguish prosodic quality of different TTS systems with a greater degree of precision than MOS-only tests. The experiments highlighted the usefulness of including question-answer

test materials, and more generally stimuli which induce clear prosodic expectations. This new test set provided evidence for an expectation-driven model of prosody perception in TTS. This highlighted the fact that the high prosodic variability, often associated with expressive TTS, may be perceived as errors when it doesn't match prosodic expectations induced by the context.

Future work will involve a more detailed study of the acoustic properties of the error markings, and the priming effect of RPT-based error marking on PMOS scores. We would also like to extend this work to evaluate other long-form synthesis, e.g. narrative and conversational TTS, to better understand when contexts admits prosodic variation. We would also like to extend the paradigm to evaluate, for example, speaker intent and speaker stance.

Acknowledgements. This work was supported in part by: ANID, Becas Chile, nº 72190135.

7. References

[1] Z. Malisz, G. E. Henter, C. Valentini-Botinhao, O. Watts, J. Beskow, and J. Gustafson, "Modern speech synthesis for phonetic sciences: A discussion and an evaluation," in *Proceedings of ICPhS 2019*, 2019.

[2] O. Watts, Z. Wu, and S. King, "Sentence-level control vectors for deep neural network speech synthesis," in *Proceedings Interspeech 2015*, 2015.

[3] Z. Hodari, C. Lai, and S. King, "Perception of prosodic variation for speech synthesis using an unsupervised discrete representation of F0," in *Proceedings of Speech Prosody 2020*, 2020.

[4] Y. Lee and T. Kim, "Robust and fine-grained prosody control of end-to-end speech synthesis," in *Proceedings of ICASSP 2019*. IEEE, 2019.

[5] Z. Hodari, O. Watts, and S. King, "Using generative modelling to produce varied intonation for speech synthesis," in *Proceedings of SSW 2019*, 2019.

[6] R. Clark, H. Silen, T. Kenter, and R. Leith, "Evaluating Long-form Text-to-Speech: Comparing the Ratings of Sentences and Paragraphs Rob," in *Proceedings of SSW 2019*, 2019.

[7] S. Tyagi, M. Nicolis, J. Rohnke, T. Drugman, and J. Lorenzo-Trueba, "Dynamic prosody generation for speech synthesis using linguistics-driven acoustic embedding selection," in *Interspeech*, H. Meng, B. Xu, and T. F. Zheng, Eds. Shanghai, China: ISCA, 2020, pp. 4407–4411.

[8] P. Wagner, J. Beskow, S. Betz, J. Edlund, J. Gustafson, G. Eje Henter, S. Le Maguer, Z. Malisz, E. Szekely, C. Tannander, and J. Vosse, "Speech Synthesis Evaluation — State-of-the-Art Assessment and Suggestion for a Novel Research Program," pp. 105–110, 2019.

[9] C. Lai, M. Farrús, and J. D. Moore, "Integrating lexical and prosodic features for automatic paragraph segmentation," *Speech Communication*, vol. 121, pp. 44–57, 2020.

[10] J. Kleinhans, M. Farrús, A. Gravano, J. M. Pérez, C. Lai, and L. Wanner, "Using prosody to classify discourse relations," in *Proceedings of Interspeech 2017*, 2017.

[11] E. Shriberg, A. Stolcke, D. Jurafsky, N. Coccaro, M. Meteer, R. Bates, P. Taylor, K. Ries, R. Martin, and C. Van Ess-Dykema, "Can prosody aid the automatic classification of dialog acts in conversational speech?" *Language and speech*, vol. 41, no. 3-4, pp. 443–492, 1998.

[12] T. Tran, "Neural models for integrating prosody in spoken language understanding," Ph.D. dissertation, University of Washington, 2020.

[13] A. Gravano and J. Hirschberg, "Turn-taking cues in task-oriented dialogue," *Computer Speech & Language*, vol. 25, no. 3, pp. 601–634, 2011.

[14] M. Steedman, "Information structure and the syntax-phonology interface," *Linguistic Inquiry*, vol. 31, no. 4, pp. 649–689, 2000.

[15] S. Calhoun, "The centrality of metrical structure in signaling information structure: A probabilistic perspective," *Language*, pp. 1–42, 2010.

[16] M. Breen, E. Fedorenko, M. Wagner, and E. Gibson, "Acoustic correlates of information structure," *Language and Cognitive Processes*, vol. 25, no. 7, pp. 1044–1098, 2010.

[17] A. Aubin, A. Cervone, O. Watts, and S. King, "Improving speech synthesis with discourse relations." in *Interspeech 2019*, 2019, pp. 4470–4474.

[18] M. White, R. A. Clark, and J. D. Moore, "Generating tailored, comparative descriptions with contextually appropriate intonation," *Computational Linguistics*, vol. 36, no. 2, pp. 159–201, 2010.

[19] Y. Mo, J. Cole, and E.-K. Lee, "Naïve listeners' prominence and boundary perception," *Proc. Speech Prosody 2008*, 2008.

[20] J. Cole and S. Shattuck-Hufnagel, "New methods for prosodic transcription: Capturing variability as a source of information," *Laboratory Phonology*, vol. 7, no. 1, 2016.

[21] V. J. van Heuven and R. van Bezooijen, "Quality evaluation of synthesized speech," in *Speech coding and synthesis*, 1995, no. 21, pp. 707–738.

[22] M. Wester, C. Valentini-Botinhao, and G. E. Henter, "Are we using enough listeners? No! An empirically-supported critique of Interspeech 2014 TTS evaluations," in *Proceedings of Interspeech 2015*, 2015.

[23] T. B. Roettger, T. Mahrt, and J. Cole, "Mapping prosody onto meaning–the case of information structure in american english," *Language, Cognition and Neuroscience*, vol. 34, no. 7, pp. 841–860, 2019.

[24] A. K. Syrdal and J. McGory, "Inter-transcriber reliability of ToBI prosodic labeling," in *Proceedings of ICSLP 2000*, 2000.

[25] J. V. Ralston, D. B. Pisoni, S. E. Lively, B. G. Greene, and J. W. Mullennix, "Comprehension of synthetic speech produced by rule: Word monitoring and sentence-by-sentence listening times," *Human factors*, vol. 33, no. 4, pp. 471–491, 1991.

[26] J. Cole, "Prosody in context: a review," *Language, Cognition and Neuroscience*, vol. 30, no. 1-2, pp. 1–31, 2015.

[27] J. Edlund, C. Tånnander, and J. Gustafson, "Audience response system-based assessment for analysis-by-synthesis." in *Proceedings of ICPhS*, 2015.

[28] H. Zen, V. Dang, R. Clark, Y. Zhang, R. J. Weiss, Y. Jia, Z. Chen, and Y. Wu, "LibriTTS: A Corpus Derived from LibriSpeech for Text-to-Speech," *Proc. Interspeech 2019*, 2019.

[29] R. A. Clark, K. Richmond, and S. King, "Multisyn: Open-domain unit selection for the festival speech synthesis system," *Speech Communication*, vol. 49, no. 4, pp. 317–330, 2007.

[30] CSTR-Edinburgh, "Ophelia," 2018. [Online]. Available: https://github.com/CSTR-Edinburgh/ophelia

[31] A. Łańcucki, "Fastpitch: Parallel text-to-speech with pitch prediction," *arXiv preprint arXiv:2006.06873*, 2020.

[32] I. Keith and J. Linda, "The LJ Speech Dataset," 2017. [Online]. Available: https://keithito.com/LJ-Speech-Dataset/

[33] T. Mahrt, "LMEDS: A Platform for Collecting Prosodic Annotations Online," 2016. [Online]. Available: https://github.com/timmahrt/LMEDS

[34] K. Krippendorff, "Computing krippendorff's alpha-reliability," 2011. [Online]. Available: https://repository.upenn.edu/asc_papers/43/

[35] A. Govender and S. King, "Using pupillometry to measure the cognitive load of synthetic speech," in *Proceedings of Interspeech 2018*, 2018.

11th ISCA Speech Synthesis Workshop (SSW 11)
26-28 August 2021, Budapest, Hungary

Speech Synthesis from Text and Ultrasound Tongue Image-based Articulatory Input

Tamás Gábor Csapó[1,2], *László Tóth*[3], *Gábor Gosztolya*[3,4], *Alexandra Markó*[2,5]

[1]Department of Telecommunications and Media Informatics,
Budapest University of Technology and Economics, Budapest, Hungary
[2]MTA-ELTE Lendület Lingual Articulation Research Group, Budapest, Hungary
[3]Institute of Informatics, University of Szeged, Hungary
[4]MTA-SZTE Research Group on Artificial Intelligence, Szeged, Hungary
[5]Department of Applied Linguistics and Phonetics, Eötvös Loránd University, Budapest, Hungary
csapot@tmit.bme.hu, {tothl, ggabor}@inf.u-szeged.hu, marko.alexandra@btk.elte.hu

Abstract

Articulatory information has been shown to be effective in improving the performance of HMM-based and DNN-based text-to-speech synthesis. Speech synthesis research focuses traditionally on text-to-speech conversion, when the input is text or an estimated linguistic representation, and the target is synthesized speech. However, a research field that has risen in the last decade is articulation-to-speech synthesis (with a target application of a Silent Speech Interface, SSI), when the goal is to synthesize speech from some representation of the movement of the articulatory organs. In this paper, we extend traditional (vocoder-based) DNN-TTS with articulatory input, estimated from ultrasound tongue images. We compare text-only, ultrasound-only, and combined inputs. Using data from eight speakers, we show that that the combined text and articulatory input can have advantages in limited-data scenarios, namely, it may increase the naturalness of synthesized speech compared to single text input. Besides, we analyze the ultrasound tongue recordings of several speakers, and show that misalignments in the ultrasound transducer positioning can have a negative effect on the final synthesis performance.

Index Terms: articulation-to-speech, ultrasound, DNN-TTS

1. Introduction

Speech synthesis has the goal of generating human-like speech from some a specific input representation. Traditionally, this research focuses on text-to-speech synthesis, when the input is text or an estimated linguistic representation. However, a research field that has risen in the last decade is articulation-to-speech synthesis (more frequently called as articulatory-to-acoustic mapping, AAM), when the goal is to synthesize speech from some representation of the movement of the articulatory organs, without having direct access to the textual contents [1, 2]. With the advent of neural vocoders, DNN-based text-to-speech synthesis has reached a mature level, i.e. if there is a large speech database (tens of hours) available, the final synthesized speech can reach the naturalness of human communication. However, such a large database is not always available, especially when other biosignals are recorded in parallel with speech. Therefore, in limited data scenarios, DNN-TTS systems with traditional vocoders can be used. In case of articulation-to-speech mapping, there is a lack of such large databases, mainly because of the limited possibilities for recording articulatory movement in parallel with speech. Most of the articulatory recording equipment becomes highly uncomfortable for the speaker after roughly an hour. For example, recording Ultrasound Tongue Image (UTI) data requires wearing a headset, while for Electromagnetic Articulatory (EMA) recordings, cables are glued onto the tongue of the speaker. Therefore, it is worth dealing with traditional (not end-to-end) DNN-TTS methods, in case we have speech and related biosignals to process. With recent methods like WORLD [3], MagPhase [4], or our Continuous vocoder [5], speech analysis and generation in statistical parametric speech synthesis has reached a mature level.

1.1. Articulatory-to-Acoustic Mapping

Speech sounds result from a coordinated movement of articulation organs (vocal cords, tongue, lips, etc.). The relationship between articulation and the resulting speech signal has been studied recently by machine learning tools as well. One of the research fields investigating such relationship is articulatory-to-acoustic (forward) mapping, when the input is a speech-related biosignal (e.g. tongue or lip movement), and the target is synthesized speech. AAM can contribute to the development of 'Silent Speech Interface' systems (SSI [1, 2]). The essence of SSI is recording the articulation organs while the user of the device actually does not make a sound, but yet the machine system can synthesize speech based on the movement of the organs. In the long-term, this potential application can contribute to the creation of a communication tool for speech-impaired people (e.g. those who lost voice after laryngectomy). Voice assistants are getting popular lately, but they are still not in every home. One of the reasons is privacy concerns; some people do not feel comfortable if they have to speak loud, having others around – but an SSI equipment can be a solution for that.

For AAM, one potential biosignal is ultrasound tongue imaging [6, 7, 8, 9]. For the articulatory-to-acoustic conversion, typically, traditional [8] or neural vocoders [9] are used, which synthesize speech from the spectral parameters predicted by the DNNs from the articulatory input.

1.2. Ultrasound tongue imaging

Ultrasound tongue imaging (UTI) is a technique suitable for the acquisition of articulatory data. Phonetic research has employed 2D ultrasound for a number of years for investigating tongue movements during speech [10]. Stone summarized the typical methodology of investigating speech production using

10.21437/SSW.2021-6

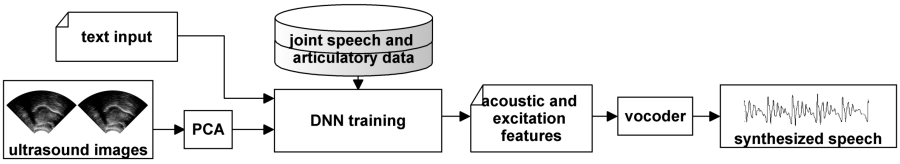


Figure 1: *Block diagram of the proposed approach.*

ultrasound [11]. Usually, when the subject is speaking, the ultrasound transducer is placed below the chin, resulting in midsagittal images of the tongue movement. Coronal images can also be acquired, depending on the orientation of the transducer. The typical result of 2D ultrasound recordings is a series of gray-scale images in which the tongue surface contour has a greater brightness than the surrounding tissue and air. Compared to other articulatory acquisition methods (e.g. EMA, X-ray, XRMB, and vocal tract MRI), UTI has the advantage that the tongue surface is fully visible, and ultrasound can be recorded in a non-invasive way [11, 8, 12]. An ultrasound device is easy to handle and move, since it is small and light, and thus it is suitable for fieldworks, as well. Besides, it is a significantly less expensive piece of equipment than the above mentioned devices. Because of these advantages, in our study, we are using ultrasound as the articulatory information.

1.3. TTS extended with articulatory data

Articulatory information has been shown to be effective in improving the performance of HMM-based and DNN-based text-to-speech synthesis – in an overview, Richmond and his colleagues summarize the use of articulatory data in speech synthesis applications [13]. Ling et al. tested several ways of integrating EMA-based features into HMM-TTS [14]. They estimated the joint distribution of acoustic and articulatory features during training, by applying model clustering, state synchrony and cross-stream feature dependency. According to the results, the accuracy of acoustic parameter prediction and the naturalness of synthesized speech could be improved. Next, vowel creation [15] and articulatory control was added to HMM-TTS [16]: with an appropriate articulatory feature sequence, new vowels can be generated even when they do not exist in the training set, without using acoustic samples. The results have been also integrated into the MAGE framework [17]. Cao et al. proposed a solution to integrate EMA-based articulatory data to DNN-TTS [18]. The integration was done in two ways: 1) articulatory and acoustic features were both the target of the DNN, 2) an additional DNN represented the articulatory-to-acoustic mapping. Both naturalness and speaker identity was improved, compared to a baseline system without articulatory data.

As shown above, integrating articulatory data to text-to-speech synthesis can improve the vocoding quality by providing more information about the vocal tract, but there is few research on this. Articulatory features derived from medical imaging data (e.g. ultrasound or MRI) have not been used before for additional input of HMM-TTS or DNN-TTS.

1.4. Contributions of this paper

In this paper, we extend traditional (vocoder-based) DNN-TTS with articulatory input, estimated from ultrasound tongue images. We show on the data of several speakers that this can have advantages in limited-data scenarios, in increasing the naturalness of synthesized speech compared to text input.

2. Methods

2.1. Data

We experimented with four English male (03mn, 04me, 05ms, 07me) and four female subjects (01fi, 02fe, 06fe, and 09fe) from the UltraSuite-TaL80 database [19] (`https://ultrasuite.github.io/data/tal_corpus/`). In parallel with speech (digitized at 48 kHz), the tongue movement was recorded in midsagittal orientation using the "Micro" ultrasound system of Articulate Instruments Ltd. at 81.5 fps. Lip video was also recorded in UltraSuite-TaL80, but we did not use that information in the current study. The ultrasound data and the audio signals were synchronized using the tools provided by Articulate Instruments Ltd. Each speaker read roughly 200 sentences – the duration of the recordings was about 15 minutes, which we partitioned into training, validation and test sets in a 85-10-5 ratio.

2.2. Processing the ultrasound data

In our experiments, articulatory features estimated from the raw scanline data of the ultrasound (i.e., echo-returns) were used as additional input of the text-to-acoustic prediction networks. We resized the 64×842 pixel images to 64×128 pixels using bicubic interpolation, and calculated PCA coefficients, similarly to EigenTongues [20]. While calculating the PCA, we aimed at keeping the 70% of the variance of the original images, thus having 128 coefficients. To be in synchrony with the acoustic features (frame shift of 5 ms), the ultrasound data was resampled to 200 Hz.

2.3. DNN-TTS framework and DNN training

Fig. 1 illustrates the proposed approach, i.e. the combined articulatory and text input for the acoustic feature prediction using a DNN. The experiments were conducted in the Merlin DNN-TTS framework [21] (`https://github.com/CSTR-Edinburgh/merlin`). Textual / phonetic parameters are first converted to a sequence of linguistic features as input (based on a decision tree), which are extended with the PCA-compressed version of the ultrasound tongue images. Next, neural networks are employed to predict acoustic and excitation features as output for synthesizing speech, at a 5 ms frame step with the WORLD vocoder (60-dimensional MGC, 5-dimensional BAP, and 1-dimensional LF0, with delta and delta-delta features). The DNN used here is a feed-forward multilayer perceptron architecture (six hidden layers, 1024 neurons in each). We applied tangent hyperbolic activation function, SGD optimizer, and a batch size of 256. The input features had min-max normalization, while output acoustic features had

mean-variance normalization. We trained the networks for 25 epochs with a warm-up of 10 epochs, applying early stopping, and a learning rate of 0.002 after that with exponential decay. We only trained an acoustic model, and the durations were not modeled.

For baseline, we created two systems: one with text-only input, and another one with ultrasound-only input. The text-only input follows the standard Merlin recipe. The ultrasound-only input was achieved in a way that the decision tree which calculates the linguistic features was replaced with an empty tree. This way, all the remaining parameters of the training are the same in the three systems, and only the input of the networks is different.

3. Experimental Results

To measure the validation and test error, we calculated both spectral prediction error (Mel-Cepstral Distortion, MCD), and excitation related errors (BAP, F0-RMSE, F0-correlation, and F0-VUV). As we only trained acoustic models, and the durations were not modeled, warping the acoustic features in time was not necessary for calculating the error measures. Several synthesized samples can be found at `http://smartlab.tmit.bme.hu/ssw11_txt-ult2wav`.

Table 1 summarizes the MCD results. For all speakers, the 'ult2wav' (articulatory-to-speech synthesis) system achieved the highest MCD errors (between 6.9–8.4 dB), indicating that these are relative different from the original natural utterances. The 'txt2wav' (text-to-speech synthesis) system can achieve significantly lower MCD errors, which are typically in the range of DNN-TTS with limited data (5.7–6.4 dB). Finally, the 'txt+ult2wav' (text-to-speech synthesis extended with articulatory input) system resulted in the lowest MCD scores (in the range of 5.5–6.2 dB). According to this, adding the ultrasound-based articulatory information could enhance the prediction of the spectral features.

The results of the excitation features are summarized in Tables 2, 3, 4, and 5. In case of BAP (being an error difference calculated on the ban aperiodicities), the tendencies are similar as in the case of MCD: 'ult2wav' > 'txt2wav' > 'txt+ult2wav'. However, in case of the F0-related measures (RMSE, CORR, and VUV), the results are less straightforward. In terms of F0-RMSE, the additional articulatory input could not help during text-to-F0 prediction – but the F0 errors with all three systems are in similar range, indicating that ultrasound itself contains some information, of which the F0 can be predicted. This is in accordance with our earlier ultrasound-to-F0 prediction experiments [22, 23]. F0-CORR, on the other hand, is similar to MCD and BAP: here, adding the articulatory information was helpful, compared to text-only input. Interestingly, with some speakers (04me and 09fe), 'ult2wav' achieved higher correlations than 'txt2wav'. Finally, as can be seen in Table 5, voicing can be estimated very poorly from ultrasound-only input, and adding the articulatory information to the text input did not help to improve the voiced/unvoiced decision.

Overall, we found that adding ultrasound-related articulatory information besides the textual input was useful for the spectral and BAP prediction, and in some of the F0 measures. However, there is strong speaker dependency in the results.

Table 1: *MCD errors on the dev/test set.*

	MCD		
Spkr	ult2wav	txt2wav	txt+ult2wav
01fi	8.005 / 8.094	5.720 / 5.636	5.639 / 5.565
02fe	7.674 / 7.585	5.974 / 5.625	5.767 / 5.564
03mn	7.328 / 7.153	5.703 / 5.652	5.523 / 5.442
04me	7.300 / 7.126	5.797 / 5.864	5.634 / 5.635
05ms	8.037 / 8.239	5.777 / 5.741	5.651 / 5.661
06fe	6.997 / 7.050	5.652 / 5.447	5.490 / 5.236
07me	8.426 / 8.396	5.989 / 5.943	5.851 / 5.928
09fe	7.818 / 8.351	6.351 / 6.566	6.230 / 6.439

Table 2: *BAP errors on the dev/test set.*

	BAP		
Spkr	ult2wav	txt2wav	txt+ult2wav
01fi	0.433 / 0.428	0.291 / 0.269	0.290 / 0.276
02fe	0.311 / 0.311	0.246 / 0.247	0.241 / 0.254
03mn	0.426 / 0.402	0.319 / 0.322	0.317 / 0.323
04me	0.338 / 0.346	0.285 / 0.262	0.270 / 0.265
05ms	0.385 / 0.400	0.302 / 0.283	0.287 / 0.276
06fe	0.521 / 0.560	0.373 / 0.391	0.386 / 0.392
07me	0.689 / 0.764	0.437 / 0.450	0.454 / 0.464
09fe	0.458 / 0.511	0.350 / 0.397	0.343 / 0.394

Table 3: *F0-RMSE errors on the dev/test set.*

	F0-RMSE		
Spkr	ult2wav	txt2wav	txt+ult2wav
01fi	22.333 / 22.062	21.301 / 19.837	22.987 / 20.087
02fe	27.742 / 35.703	25.833 / 33.186	27.461 / 33.504
03mn	11.269 / 10.094	10.036 / 9.582	10.200 / 9.330
04me	17.809 / 23.491	21.672 / 28.472	15.955 / 22.793
05ms	11.786 / 11.892	11.569 / 13.208	10.855 / 10.724
06fe	51.407 / 40.897	40.784 / 39.614	42.861 / 39.871
07me	24.407 / 27.420	20.767 / 26.082	20.561 / 24.422
09fe	54.811 / 61.934	48.048 / 51.004	54.527 / 54.714

Table 4: *F0-CORR errors on the dev/test set.*

	F0-CORR		
Spkr	ult2wav	txt2wav	txt+ult2wav
01fi	0.528 / 0.602	0.627 / 0.702	0.634 / 0.701
02fe	0.347 / 0.265	0.400 / 0.470	0.360 / 0.477
03mn	0.255 / 0.303	0.548 / 0.468	0.498 / 0.470
04me	0.715 / 0.741	0.523 / 0.423	0.782 / 0.745
05ms	0.550 / 0.590	0.565 / 0.560	0.649 / 0.734
06fe	0.425 / 0.657	0.672 / 0.649	0.631 / 0.652
07me	0.415 / 0.377	0.624 / 0.448	0.631 / 0.499
09fe	0.551 / 0.448	0.528 / 0.646	0.562 / 0.594

Table 5: *F0-VUV errors on the dev/test set.*

	F0-VUV		
Spkr	ult2wav	txt2wav	txt+ult2wav
01fi	27.162 / 28.483	9.122 / 7.411	9.381 / 7.972
02fe	24.228 / 19.541	10.763 / 8.063	9.927 / 8.092
03mn	18.959 / 16.357	6.833 / 6.828	7.142 / 7.674
04me	21.597 / 22.342	11.602 / 9.717	11.320 / 10.239
05ms	26.693 / 30.381	11.560 / 12.669	12.202 / 12.929
06fe	24.201 / 21.477	12.217 / 7.514	13.079 / 8.352
07me	24.598 / 25.851	11.191 / 9.870	11.394 / 10.566
09fe	22.161 / 27.173	8.608 / 11.318	9.867 / 11.700

4. Effect of ultrasound transducer position

Next, we further investigate the strongly speaker-dependent results found in Section 3. The articulatory tracking devices (like the ultrasound used in this study) are obviously highly sensitive to the speaker and the position of the device. A source of variance comes from the possible misalignment of the recording equipment. For example, for ultrasound recordings, the probe fixing headset has to be mounted onto the speaker before use, and in practice it is impossible to mount it onto exactly the same spot as before. Therefore, such recordings are not directly comparable. Ultrasound-based SSI systems might not turn out to be robust against slight changes in probe positioning, which can cause shifts and rotations in the image used as input.

4.1. Ultrasound transducer positioning and misalignment

In order to fix head movement during the ultrasound recordings, various solutions have been proposed, e.g. the HATS system aimed to provide reliable tongue motion recordings by head immobilization and positioning the transducer in a known relationship to the head [24]. The metal headset of Articulate Instruments Ltd. is a popular and well designed solution which was used in a number of studies (e.g. articulatory-to-acoustic mapping [8, 23]). Recently, a non-metallic system by [25] and UltraFit by [26] are lightweight headsets to record ultrasound and EMA data. During the recording of UltraSuite-TaL [19]), the UltraFit headset was used.

Despite these substantial efforts, it is still a question whether the use of a headset itself is enough to ensure that the transducer is not moving during the recordings. Even if a transducer fixing system is used, large jaw movements during speech production (or drinking, swallowing) can cause the ultrasound transducer to move, and misalignment or full displacement might occur. Besides, the subjects, having discomfort due to the fixing system, sometimes readjust the headset. This way the recordings from the same session will not be directly comparable, which can be a serious issue during analysis of tongue contours. Although there exist methods for non-speech ultrasound transducer misalignment detection [27, 28], they cannot be directly used in speech production research.

In our earlier work [29, 30], we presented an initial idea for analyzing such misalignment. The method employs Mean Square Error (MSE) distance to identify the relative displacement between the chin and the transducer. We visualized these measures as a function of the timestamp of the utterances. Experiments were conducted on various ultrasound tongue datasets (UltraSuite, and recordings of Hungarian children and adults). The results suggested that extreme values of MSE indicate corruptions or issues during the data recordings, which can either be caused by transducer misalignment, lack of gel, or missing contact between the skin and the transducer.

4.2. Measuring ultrasound transducer misalignment

The speaker-by-speaker differences of the ultrasound-to-speech conversion of the current study might also be explained with the issues of the ultrasound tongue image representation. In order to quantify the amount of misalignment, we used the MSE calculation method from our earlier study [29, 30]. We compared all utterances of the eight speakers from UltraSuite-TaL with each other in the order in which they were recorded . First, for a given speaker and given session, we go through all of the ultrasound recordings (utterances), and calculate the pixel by pixel mean image (across time) of each utterance (see Fig. 1 in [30]). Next, we compare these mean images: we measure the Mean Square Error (MSE) between the UTI pixels ([0-255] grayscale values). MSE is an error measure, therefore the lower numbers indicate higher similarity across images. For a session with n consecutive utterances, all compared with each other, the result is an $n \times n$ matrix (see Fig. 2 in [30]). We assume that if there is misalignment in the ultrasound transducer, than the matrix of measures would show this. The full details of the method, including two more similarity measures were introduced in [29].

The results of the ultrasound transducer misalignment MSE are shown in Fig. 2. For each speaker, the first 85% of the data was used for training, the next 10% for development, and the remaining 5% for testing. On the MSE matrices of Fig. 2, the bottom left corner (or the top right corner, because the error is symmetric) indicates the differences in the positioning of the ultrasound transducer, between the training and the development/test data. If the color is yellowish, it means a higher MSE difference, i.e. larger misalignment of the transducer. For some of the speakers, the test utterances are clearly far away (in terms of average ultrasound image) from the training utterances. For speakers 01fi, 04me, 05ms, and 07me this tendency is visible, and comparing the MSE figures (Fig. 2) with the MCD results on the development/test set (Table 1), we can observe higher errors for them than for the remaining speakers. In case of speaker 06fe, the MSE matrix in Fig. 2 is relatively homogenous, and his MCD in Table 1 is the lowest. Quantifying the exact relation between the ultrasound transducer misalignment and the acoustic / excitation errors remains future work. Also, it might be possible to auto-rotate the ultrasound images to compensate such misalignments, by comparing the actual image to an average tongue shape.

5. Discussion and Conclusions

In Sec. 1.3, we summarized the earlier approaches that extended TTS systems with articulatory data. Most of these studies were conducted with HMMs [14, 16, 17], but the ideas could be applied similarly using deep neural networks, as in our experiments. All of these previous works are applying EMA as articulatory data, which is a point tracking equipment, and therefore processing that data is significantly different from the ultrasound signal that we used here. Also, the previous studies differ in the way how they include the articulatory information: it might be the input [18], or the target of the machine learning method [14, 15, 16], or also an internal representation [18]. Besides, there are many examples for DNN-based articulatory-to-acoustic mapping applying ultrasound as input, but without using the textual information [8, 9, 22, 23, 31]. Although the system proposed in the current study is not suitable for direct

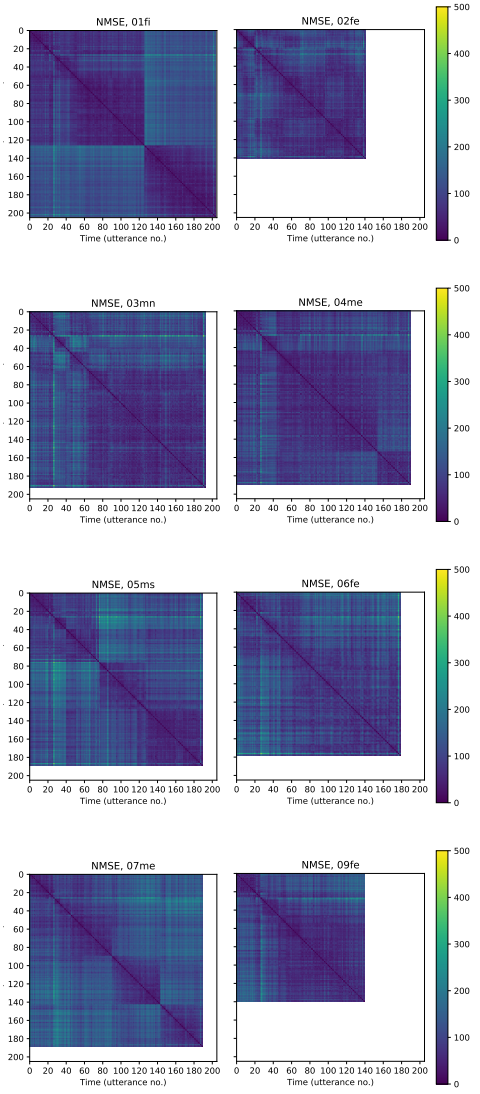


Figure 2: *Ultrasound transducer misalignment as a function of the utterance number within the recording session. MSE: lower values (blue colors) indicate smaller misalignment. The diagonals contain NaN values.*

TTS or for a Silent Speech Interface, as for the combined mapping, both text and articulatory input are required, our methods are a kind of scientific exploration, and the text-to-speech and ultrasound-to-speech results shown above might be useful for other modalities having similar properties (e.g. rtMRI and lip images).

In this paper, we extended traditional (vocoder-based) DNN-TTS with articulatory input. The articulatory input was estimated from ultrasound tongue images, with a PCA-based compression to 128 dimensions. We have shown on the data of eight speakers from the UltraSuite-TaL dataset that this can have advantages in limited-data scenarios (e.g. when the training data is in the range of 200 sentences for each speaker), in increasing the naturalness of synthesized speech compared to text-only or ultrasound-only input. During our experiments, we were training speaker-dependent DNNs. Creating an average voice, and adapting to a specific speaker remains future work, as it is not a trivial task. For speaker-independent training, the challenge will be to find a suitable representation of the ultrasound images, as the PCA trained on the articulatory data of one speaker is not transferable for other speakers. In the future, we plan to investigate extending DNN-TTS with other types of biosignals (e.g. MRI or video of the lips).

The implementations are accessible at `https://github.com/BME-SmartLab/txt-ult2wav`.

6. Acknowledgements

The authors were funded by the National Research, Development and Innovation Office of Hungary (FK 124584 and PD 127915 grants). This research was supported by the project "Integrated program for training new generation of scientists in the fields of computer science", no EFOP-3.6.3-VEKOP-16-2017-00002. The project has been supported by the European Union and co-funded by the European Social Fund. We would like to thank CSTR for providing the Merlin toolkit and the UltraSuite-TaL articulatory database.

7. References

[1] B. Denby, T. Schultz, K. Honda, T. Hueber, J. M. Gilbert, and J. S. Brumberg, "Silent speech interfaces," *Speech Communication*, vol. 52, no. 4, pp. 270–287, 2010.

[2] J. A. Gonzalez-Lopez, A. Gomez-Alanis, J. M. Martin Donas, J. L. Perez-Cordoba, and A. M. Gomez, "Silent Speech Interfaces for Speech Restoration: A Review," *IEEE Access*, vol. 8, pp. 177 995–178 021, sep 2020.

[3] M. Morise, F. Yokomori, and K. Ozawa, "WORLD: A Vocoder-Based High-Quality Speech Synthesis System for Real-Time Applications," *IEICE Transactions on Information and Systems*, vol. E99.D, no. 7, pp. 1877–1884, 2016.

[4] F. Espic, C. Valentini-Botinhao, and S. King, "Direct modelling of magnitude and phase spectra for statistical parametric speech synthesis," in *Proc. Interspeech*, Stockholm, Sweden, 2017, pp. 1383–1387.

[5] M. S. Al-Radhi, O. Abdo, T. G. Csapó, S. Abdou, G. Németh, and M. Fashal, "A continuous vocoder for statistical parametric speech synthesis and its evaluation using an audio-visual phonetically annotated Arabic corpus," *Computer Speech and Language*, vol. 60, p. 101025, mar 2020.

[6] B. Denby and M. Stone, "Speech synthesis from real time ultrasound images of the tongue," in *Proc. ICASSP*, Montreal, Quebec, Canada, 2004, pp. 685–688.

[7] T. Hueber, E.-L. Benaroya, G. Chollet, G. Dreyfus, and M. Stone, "Development of a silent speech interface driven by ultrasound

and optical images of the tongue and lips," *Speech Communication*, vol. 52, no. 4, pp. 288–300, 2010.

[8] T. G. Csapó, T. Grósz, G. Gosztolya, L. Tóth, and A. Markó, "DNN-Based Ultrasound-to-Speech Conversion for a Silent Speech Interface," in *Proc. Interspeech*, Stockholm, Sweden, 2017, pp. 3672–3676.

[9] T. G. Csapó, C. Zainkó, L. Tóth, G. Gosztolya, and A. Markó, "Ultrasound-based Articulatory-to-Acoustic Mapping with WaveGlow Speech Synthesis," in *Proc. Interspeech*, 2020, pp. 2727–2731.

[10] M. Stone, B. Sonies, T. Shawker, G. Weiss, and L. Nadel, "Analysis of real-time ultrasound images of tongue configuration using a grid-digitizing system," *Journal of Phonetics*, vol. 11, pp. 207–218, 1983.

[11] M. Stone, "A guide to analysing tongue motion from ultrasound images," *Clinical Linguistics and Phonetics*, vol. 19, no. 6-7, pp. 455–501, jan 2005.

[12] V. Ramanarayanan, S. Tilsen, M. Proctor, J. Töger, L. Goldstein, K. S. Nayak, and S. Narayanan, "Analysis of speech production real-time MRI," *Computer Speech and Language*, vol. 52, pp. 1–22, 2018.

[13] K. Richmond, Z. Ling, and J. Yamagishi, "The use of articulatory movement data in speech synthesis applications: An overview - Application of articulatory movements using machine learning algorithms," *Acoustical Science and Technology*, vol. 36, no. 6, pp. 467–477, 2015.

[14] Z.-H. Ling, K. Richmond, J. Yamagishi, and R.-H. Wang, "Integrating Articulatory Features Into HMM-Based Parametric Speech Synthesis," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 6, pp. 1171–1185, aug 2009.

[15] Z.-H. Ling, K. Richmond, and J. Yamagishi, "Vowel Creation by Articulatory Control in HMM-based Parametric Speech Synthesis," in *Proc. Interspeech*, Portland, OR, USA, 2012, pp. 991–994.

[16] ——, "Articulatory Control of HMM-Based Parametric Speech Synthesis Using Feature-Space-Switched Multiple Regression," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 1, pp. 207–219, jan 2013.

[17] M. Astrinaki, A. Moinet, J. Yamagishi, K. Richmond, Z.-h. Ling, S. King, and T. Dutoit, "Mage - Reactive articulatory feature control of HMM-based parametric speech synthesis," in *Proc. ISCA SSW8*, Barcelona, Spain, 2013, pp. 207–211.

[18] B. Cao, M. Kim, J. van Santen, T. Mau, and J. Wang, "Integrating Articulatory Information in Deep Learning-Based Text-to-Speech Synthesis," in *Proc. Interspeech*, Stockholm, Sweden, 2017, pp. 254–258.

[19] M. S. Ribeiro, J. Sanger, J.-X. X. Zhang, A. Eshky, A. Wrench, K. Richmond, and S. Renals, "TaL: a synchronised multi-speaker corpus of ultrasound tongue imaging, audio, and lip videos," in *2021 IEEE Spoken Language Technology Workshop (SLT)*, Shenzhen, China, 2021, pp. 1109–1116.

[20] T. Hueber, G. Aversano, G. Chollet, B. Denby, G. Dreyfus, Y. Oussar, P. Roussel, and M. Stone, "Eigentongue feature extraction for an ultrasound-based silent speech interface," in *Proc. ICASSP*, Honolulu, HI, USA, 2007, pp. 1245–1248.

[21] Z. Wu, O. Watts, and S. King, "Merlin: An Open Source Neural Network Speech Synthesis System," in *9th ISCA Speech Synthesis Workshop*. Sunnyvale, CA, USA: ISCA, sep 2016, pp. 202–207.

[22] T. Grósz, G. Gosztolya, L. Tóth, T. G. Csapó, and A. Markó, "F0 Estimation for DNN-Based Ultrasound Silent Speech Interfaces," in *Proc. ICASSP*, Calgary, Canada, 2018, pp. 291–295.

[23] T. G. Csapó, M. S. Al-Radhi, G. Németh, G. Gosztolya, T. Grósz, L. Tóth, and A. Markó, "Ultrasound-based Silent Speech Interface Built on a Continuous Vocoder," in *Proc. Interspeech*, Graz, Austria, 2019, pp. 894–898.

[24] M. Stone and E. Davis, "A head and transducer support system for making ultrasound images of tongue/jaw movement," *The Journal of the Acoustical Society of America*, vol. 98, pp. 3107–3112, 1995.

[25] D. Derrick, C. Carignan, W.-r. Chen, M. Shujau, and C. T. Best, "Three-dimensional printable ultrasound transducer stabilization system," *The Journal of the Acoustical Society of America*, vol. 144, no. 5, pp. EL392–EL398, nov 2018.

[26] L. Spreafico, M. Pucher, and A. Matosova, "UltraFit: A Speaker-friendly Headset for Ultrasound Recordings in Speech Science," in *Proc. Interspeech*, Hyderabad, India, 2018, pp. 1517–1520.

[27] M. M. Narayanan, N. Singh, A. Kumar, C. Babu Rao, and T. Jayakumar, "An absolute method for determination of misalignment of an immersion ultrasonic transducer," *Ultrasonics*, vol. 54, no. 8, pp. 2081–2089, dec 2014.

[28] B. Bolsterlee, S. C. Gandevia, and R. D. Herbert, "Effect of Transducer Orientation on Errors in Ultrasound Image-Based Measurements of Human Medial Gastrocnemius Muscle Fascicle Length and Pennation," *PLOS ONE*, vol. 11, no. 6, p. e0157273, jun 2016.

[29] T. G. Csapó and K. Xu, "Quantification of Transducer Misalignment in Ultrasound Tongue Imaging," in *Proc. Interspeech*, online, 2020, pp. 3735–3739.

[30] T. G. Csapó, K. Xu, A. Deme, T. E. Gráczi, and A. Markó, "Transducer Misalignment in Ultrasound Tongue Imaging," in *12th International Seminar on Speech Production*, 2020.

[31] N. Kimura, M. C. Kono, and J. Rekimoto, "Sottovoce: An ultrasound imaging-based silent speech interaction using deep neural networks," in *CHI '19: Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, Glasgow, UK, 2019, pp. 1–11.

11th ISCA Speech Synthesis Workshop (SSW 11)
26-28 August 2021, Budapest, Hungary

Combining speakers of multiple languages to improve quality of neural voices

Javier Latorre, Charlotte Bailleul, Tuuli Morrill, Alistair Conkie, Yannis Stylianou

Apple

{jlatorrechimoto,cbailleul,tuuli_morril,aconkie,istylianou}@apple.com

Abstract

In this work, we explore multiple architectures and training procedures for developing a multi-speaker and multi-lingual neural TTS system with the goals of a) improving the quality when the available data in the target language is limited and b) enabling cross-lingual synthesis. We report results from a large experiment using 30 speakers in 8 different languages across 15 different locales. The system is trained on the same amount of data per speaker. Compared to a single-speaker model, when the suggested system is fine tuned to a speaker, it produces significantly better quality in most of the cases while it only uses less than 40% of the speaker's data used to build the single-speaker model. In cross-lingual synthesis, on average, the generated quality is within 80% of native single-speaker models, in terms of Mean Opinion Score.

Index Terms: multi-speaker synthesis, multilingual synthesis, fine-tuning, neural speech synthesis

1. Introduction

The quality of synthetic speech has improved dramatically since the development of methods based on neural networks, [1, 2]. However, using this technology requires high computational capacity and large amounts of training data. Several researchers have shown that unlike unit-selection text-to-speech (USEL), neural TTS can compensate for the lack of speech data from the target speaker by adding data from other speakers. Most of the research published in this respect has used support speakers in the same language as the target. However, the most common case when developing TTS voices for a new language is that there are no additional supporting speakers in that new language. In that context, the only available options are to record more speakers and/or to use support speakers from different languages.

In this paper, we show the results of applying the latter approach on a large-scale experiment involving 30 target speaker in 8 languages across 15 different locales. Our goal was to address the following questions: a) how effective is it to combine speakers from different languages compared with just training only on the data of the target speaker; b) what type of model architecture and training protocol yields the best quality when using multilingual data; and c) to which extent can the voices created in this way speak some of the other languages included in the training data?

In addition to the standard numerical results, we also show the analysis of the most common errors pointed out by the evaluation subjects. We believe that the results of these experiments will be useful for researchers and practitioners developing synthetic voices.

The structure of the paper is as follows. Section 2 reviews the recent literature on using data from other speakers to create new voices and on the application of this method to create polyglot voices. Section 3 describes the architecture of the models used in the experiment as well as the way in which these models were trained. Section 4 describes the conditions and results of our experiments. It also shows the analysis of most commonly mentioned mistakes for each of the systems. Section 5 discusses some of the results and suggests some possible future directions. Finally, in section 6 conclusions are drawn.[1].

2. Related work

The idea of using data from other speakers to improve the quality of synthetic speech has been explored extensively [3, 4, 5]. Although there has been some work in training multi-speaker text-to-wave models [6], most of the recent work has been in phone-to-spectrogram. For instance, in [7] the effect of reducing the amount of data from the target speaker and compensating for it with data from other speakers was studied. The effect of having imbalanced training data was further analized in [8]. Even more extreme examples were presented in [9], where only 5 minutes of speech were used to get high quality or even in [10] where a single utterance is used. When there are not sufficient support speakers, some authors have suggested to artificially expand the number of training speakers [11] or making use of low quality data [12].

Mixing languages has also been widely studied, although in most cases with the goal of creating polyglot voices. Within the sequence-to-sequence framework, [14] and [15] introduced several modifications to allow training polyglot voices using only monolingual speakers. A non sequence-to-sequence model was proposed in [13].

Even without aiming to create polyglot voices, using compensatory data from speakers in other languages is also a potential solution to the lack of data. However, this option has received less attention. An architecture inspired by the speaker and language factorisation (SLF) approach [16] but within the DNN/LSTM framework was proposed in [17]. Other authors have also shown that mixing data from multiple speakers and languages can yield equal or even better quality than single speaker models [18, 19]. Finally, in [20], 8 Indian languages were combined directly in a very similar way to the one we suggest here but using a DeepVoice3 [21] architecture.

3. Model training

3.1. Model architecture

The basic architecture of our models is Tacotron2 [2]. The main input is a sequence of phones and punctuation marks and the output is a sequence of 80-dimensional mel-spectrogram features. These are computed from speech signals with a sampling rate of 24kHz, using a 25ms analysis window and the Mel filterbank generated using Librosa Toolkit [22]. An end-pointing flag

[1]Samples can be found in https://apple.github.io/ml-polyglot_tacotron2_finetuning-samples

10.21437/SSW.2021-7

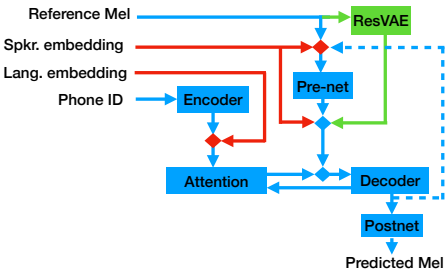


Figure 1: *Model architectures. The rhombi indicate concatenation. The dotted line linking the postnet output to the pre-net input is used only at inference time*

was also concatenated with the mel-spectrogram vector to make the final 81-dimensional output vectors.

The encoder consists of a look-up table that converts the sequence of phone-IDs into a sequence of 512-dimensional vectors, three 1D-CNNs and one bi-LSTM layers. The attention is a stepwise monotonic attention [23]. On the decoder side, the pre-net consists of two fully connected (FF) layers. The decoder itself is formed by two LSTMs followed by one FF layer to decode the mel-spectrograms and another one to generate the end-pointing signal. These mel-spectrograms are finally passed through a post-net module consisting of 5 1D-CNNs. The training loss combines the L1 for the end-pointing and the output of the decoder, and an L2 for the output of the post-net. For each output step, two output vectors were generated.

On top of that standard architecture, two variants were built, as depicted in fig. 1 The first variant (in green) consists of adding a 16-dimensional residual variational auto-encoder (resVAE) following [14]. The main goal of the resVAE is to normalise differences between the utterances that cannot be described from the input. The resVAE consists of 6 2D-CNN layers, each followed by batch normalisation, a GRU, a common FF layer and 2 additional FF layers, one for the mean and another for the covariance diagonal. A sample from this single Gaussian distribution is then concatenated at the input of the decoder. As usual, an additional loss factor for the KLD w.r.t a diagonal Gaussian was added. During inference the resVAE network is bypassed and a constant 0-vector is used instead.

The second variant (in red) is the addition of speaker and language embeddings. The speaker embedding (SE) consists of a 128 dimensional d-vector obtained from a speaker verification model [12]. One advantage of using speaker embeddings versus one-hot is that we can have different values for each utterance. Unfortunately, the SE of each utterance also contains information about the acoustics of that utterance [24]. To avoid this and simulate something akin to a VAE, a single Gaussian model of the embeddings of each speaker was computed and sampled during training. At inference, the mean of the Gaussian was used. The language embeddings (LE) are 32 dimensional vectors obtained from a one hot encoding of the locale associated with each speaker.

We experimented with different ways of adding SE and LE. For SE, we found that the best option is to insert it both at the input of the decoder, concatenated with the output of the attention and with the output of the pre-net as in [25]. This configuration yields the best results in terms of quality, voice similarity

Table 1: *Evaluated models*

System	Fine Tuned (FT)	resVAE	SE+LE	#total steps
FT	Yes	No	No	$3x10^6$
FTres	Yes	Yes	No	$3x10^6$
FTresSE	Yes	Yes	Yes	$3x10^6$
resSE	No	Yes	Yes	$4.5x10^6$

to the target speaker and voice consistency when synthesising mix-lingual sentences. For LE, the best option was to concatenate it with the output of the encoder before the attention. Concatenating LE at the beginning of the encoder or after the attention made the models' training unstable. In any case, the effect of LE was almost negligible, presumably because the phonetic sequence itself already contains enough information about the language.

Previous internal evaluations on models with SE but without fine tuning showed a preference for adding the resVAE. For that reason, all our models with SE also include resVAE. In some initial models we also included a domain adversarial NN (DANN) loss against the identification of the speaker from the encoder outputs as suggested in [14]. Although DANN provided some good results when mixing only 2-3 languages with at least 4 speakers each [24], it introduced instability when we added languages for which only two speakers were available.

3.2. Training procedure

Models that do not include any speaker information need to be fine-tuned in order to get a stable voice. Models that include SE can be used either directly, as in [7], or they can also be fine-tuned. All the base models were trained on exactly the same data. For the fine-tuning to each target speaker we used exactly the same utterances of that speaker that were used as part of the base-model training. We didn't consider experiments in which an existing model was fine-tuned to an unseen speaker because if the data for the new speaker is available, it can always be mixed with the existing speakers to create a new base model.

All models were trained on a single GPU with a batch size of 16. We used the Adam optimiser [26] with 0.9 and 0.999 for beta1 and beta2, respectively, an initial learning rate of 0.001, 4000 warm-up steps and "Noam decay scheme" [27]. The seed models were trained for 2.5 million steps and then fine-tuned for another 0.5 million steps. The non fine-tuned seed model with speaker embeddings was further trained up to 4.5 million steps. The systems that were finally evaluated are shown in Table 1 .

3.3. Normalisation of the phonetic transcriptions

We normalised the transcriptions across all locales to share a single unified language-agnostic set of phones based on XSAMPA [28]. Previous experiments had shown that in crosslingual synthesis complex phones such as diphthongs, nasalized vowels, syllabic consonants and affricates, tend to get confused and the synthesis only produces half of the phone. To avoid this problem, we split such complex phones. In this way, diphthongs were split into two vowels, affricates into a closure with no audible release plus a fricative, syllabic consonants into the consonant preceded by schwa, and nasalized vowels into a vowel followed by a velar nasal consonant.

Syllabic stress marks were also added to the vowels of the stressed syllables for all languages. It should be noted that for inlingual synthesis, (in which the spoken language is the same language as that of the target voices) most languages do not need explicit stress marks, especially those languages for which

stress is not phonemic. However, we found that in crosslingual synthesis, (which is when the synthesised utterances were in a language other than that of the target voices) the lack of stress marks caused serious intelligibility problems, even in languages which are supposed to have no phonemic stress, such as French. In crosslingual synthesis, the voices tended to apply the stress pattern of its own language, e.g., Spanish voices speaking French tended to put the stress in the penultimate syllable. Such changes of the stress patterns made the parsing of the prosodic words very difficult and thus, affected the intelligibility of the utterances.

4. Experiments

We ran two subjective evaluations, one for inlingual synthesis and another for crosslingual synthesis. All the evaluations were 5 points mean opinion score (MOS) tests conducted via crowdsource on each respective locale. The question asked was "How do you rate the overall quality of the voice?". Each utterance was evaluated by 15 different subjects and no subject was allowed to judge more than 360 samples. With these settings, the total number of listeners per voice was around 120 for the inlingual experiments and 140 for the crosslingual one.

For each evaluation, raw scores were normalized by z-scoring by subject. Mixed effects linear regression models were fitted to the data with subjects and items (sample content/sentence) as random effects and the synthesis method/voice as the fixed effect. T-tests for pairwise contrasts for each pair of voices/systems provided estimated p-values (with Bonferroni correction for the number of contrasts).

4.1. Data

The models were trained on 30 proprietary voices consisting of two speakers for 15 different locales in 8 languages: Australia, India, Ireland, South Africa, UK and US for English; Mexico and Spain for Spanish; Canada and France for French, Brazil for Portuguese, and Denmark, Germany, Italy and The Netherlands for their respective main languages. From each speaker we used 8500 utterances randomly selected from the total corpus, which on average corresponds to 7.73 hours/speaker. This amount of data corresponds on average to 37% of the data used to train the single speaker (SingSpkr) models.

4.2. Vocoder

In all the experiments, we used speaker-dependent waveRNN neural vocoders [29]. The same vocoder trained on all the data was used for each voice across all the models. The reasons for this are: a) we only wanted to evaluate differences in the acoustic model and, b) there exist proposals for universal waveRNN that work for both seen and unseen speakers [30, 31]

4.3. Inlingual synthesis

For each of the 15 locales, 150 utterances were evaluated with each of the 2 speakers' voices. In addition to the systems described in Table 1, we also evaluated SingSpkr models with the same architecture of FT models, trained from scratch on all the available data of each speaker. The sentences were the same for all the systems but not necessarily the same for both speakers. In order to provide anchors, each evaluation included 50 recorded utterances from each of the target voice talents as the high anchor and the same 150 evaluation sentences[2] generated by a hybrid unit selection system (USEL) [32] as the lower one.

[2]For 1 of the 15 locales we used 75 instead of 150 USEL utterances

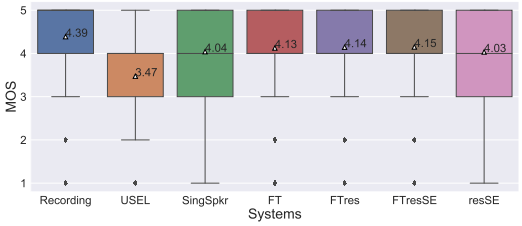


Figure 2: *MOS scores across all voices for inlingual synthesis*

Table 2: *Number of voices significantly different from SingSpkr models in inlingual synthesis.*

	USEL	FT	FTres	FTresSE	resSE
better	1	11	14	14	4
equal	1	18	14	15	15
worse	28	1	2	1	11

Figure 2 shows the box plot with the summary of the results across all voices. On average, all the fine-tuned models outperformed the SingSpkr models. The average difference between the models is around 0.1 MOS scores. Note that this is by using less than 40% of the target speaker data of the SingSpkr models. Obviously, there are variations depending on the voice. A voice-by-voice analysis is provided in Table 2. This result confirms that for most voices any of the fine-tuned models perform equal or better than the SingSpkr models. By contrast, resSe was found to be significantly worse than SingSpkr for 11 voices and only better for 4, even though both systems appear to be identical in fig. 2. Our results confirm those reported in [20] for premium voices with 15+ hours of data. Finally, we did not find any significant differences among the 3 fine-tune approaches, although both FTres and FTresSE seem to be marginally better than FT, presumably due to their higher capacity.

4.4. Crosslingual synthesis and evaluation

Our main purpose was to create a base model from which new voices for new languages can be created rapidly. However, given that the seed models are trained on multiple languages, we were curious to know to which extent the fine-tuned models still retained some multilingual capacity. Evaluating each of the 30 voices over the 7 non-native languages would have been ideal, but also very costly. For that reason, we evaluated only the non-native voices when synthesising 4 different foreign languages, American English (en-US), Mexican Spanish (es-MX), France French (fr-FR) and Germany German (de-DE). For each target language 50 utterances from each of the non-native voices were evaluated. Voices in the same main language but from a different locale were not considered. To avoid conflating the differences between native/non-native speakers with those between synthetic/natural speech, we only included as upper anchor 50 utterances generated by the native SingSpkr voice in the target language. These SingSpkr voices are the same as the ones described in Sec. 4.3. To reduce the number of different voices/systems in a single evaluation, the stimuli were split into two groups: one for the voices with the lower median fundamental frequency (F0) and another for the voices with higher median F0 from each locale. This yields a total of 8 independent MOS evaluations. In total, the number of individual voices evaluated on each experiment were 10 for English, 14 for Span-

Table 3: *Number of model comparisons across the 8 crosslingual evaluations in which the MOS difference was significant*

Systems	$\#1^{st}$ better	$\#2^{nd}$ better	#No diff.
resSE vs. FT	1	0	7
resSE vs. FTres	1	0	7
resSE vs. FTresSE	1	0	7
FT vs. FTres	0	3	5
FT vs. FTresSE	2	6	0
FTres vs. FTresSE	2	2	4

ish and French and 15 for German. Subjects were not warned that they were going to listen to foreign accented speech.

Table 3 shows for how many of the 8 evaluations the MOS difference between models were significant, and Table 4 reports the average MOS for each combination of target-language and voice-locale. In general, all systems' performance is very similar. For most voices the non fine-tuned system resVAE is usually better than the fine-tuned ones. This result is not surprising since fine-tuned models tend to "forget" previous knowledge. However, with the exception of the low-pitch voices in French those differences were not significant. Among the fine-tuned models, FTres and FTresSE were better than FT on average, probably because of the higher capacity introduced by the additional resVAE. However, the addition of the speaker embedding does not seem to provide any advantage when the model is fine-tuned.

Despite these differences, the combination of voice and target language has a much stronger impact over the speech quality than the model type. As shown in Table 4, some combinations achieve scores around 4.0 while others fall below 3.0.

4.5. Analysis of the comments

In inlingual synthesis, the main problems noted were in terms of pauses (either misplaced or too few), pace (usually too fast), unnatural intonation, and audio quality deterioration. These problems seem to affect more the resSE model. Word stress also seems to be sometimes slightly misplaced in some languages. In the resSE model, some non-phonemic distinctions are also less accurately predicted, e.g., the Italian trill is sometimes chosen instead of the flap.

In crosslingual synthesis, the foreign accent of a voice is usually well identified, but in some cases deemed too pronounced to the extent of impeding intelligibility, especially when in combination with insufficient pausing and fast pace. We also notice a degraded audio quality, affecting some voices more than others, with some occasional "blabber". Intonation contours are sometimes incorrect and sometimes deemed as monotone. In terms of pronunciation, the model without fine tuning seems to retain less accent and produces a more accurate approximation of the target language phones. This effect is notable, for example, with the American English rhotic and the French voices: the model without fine tuning being the closer to the English alveolar approximant (although getting inaudible in word final position or pre-consonantical position), and other models having a pronunciation closer or identical to the French rhotic. For most voices, lexical stress seems to be placed correctly.

In some language pairs, some phonemic distinctions are lost. For example, the Spanish trill/flap pair is not always maintained when synthesising with French, English, or German voices. The English phone /h/ is often dropped in the synthesis with the French voices. Actually, human French speakers often do drop that phoneme. However, it contributes to the impression of strong foreign accent as more proficient speakers would tend to realise it. Another factor contributing to the impression of strong foreign accent is that intonation and some phonological phenomena are ported to the target language. For instance, word final rhotic is dropped by British English voices, and sometimes French voices insert liaison in Spanish. It is also interesting to note that some American English subjects expected a genuine non-native accent. For example, they expected /t/ or /d/ instead

Table 4: *Crosslingual MOS per locale. The numbers in the target language column are the average MOS of the two 'Native SingSpkr' voices in that language.*

Target language	Speaker locale	FT	FTres	FTresSE	resSE
American English 4.2	da-DK	3.77	**3.84**	3.78	3.7
	de-DE	3.88	3.88	3.96	**4.01**
	es-ES	3.66	3.74	3.77	**3.78**
	es-MX	3.81	3.82	3.81	**3.93**
	fr-CA	3.75	3.82	3.81	**3.93**
	fr-FR	3.62	3.69	3.72	**3.79**
	it-IT	3.72	3.69	**3.76**	**3.76**
	nl-NL	3.84	3.87	3.85	**3.9**
	pt-BR	3.68	3.71	3.81	**3.88**
France French 4.35	da-DK	3.08	3.25	3.26	**3.34**
	de-DE	3.77	3.84	3.84	**3.86**
	en-AU	3.08	3.11	3.2	**3.42**
	en-GB	3.28	3.33	3.3	**3.51**
	en-IE	3.29	3.32	**3.49**	3.46
	en-IN	3.46	3.43	3.5	**3.7**
	en-US	3.27	3.29	3.29	**3.57**
	en-ZA	3.25	3.47	3.47	**3.69**
	es-ES	3.37	3.46	3.48	**3.67**
	es-MX	3.55	3.53	3.58	**3.7**
	it-IT	3.62	3.54	3.66	**3.81**
	nl-NL	3.18	3.52	3.32	**3.66**
	pt-BR	3.27	3.4	3.56	**3.72**
Mexican Spanish 4.45	da-DK	2.88	**3.08**	2.87	2.67
	de-DE	3.3	3.35	3.18	**3.38**
	en-AU	2.94	2.91	2.9	**3.04**
	en-GB	3.15	2.93	3.02	**3.13**
	en-IE	3.03	2.98	3.07	**3.21**
	en-IN	3.34	**3.45**	3.15	3.32
	en-US	**3.24**	3.14	3.12	3.14
	en-ZA	2.95	3.08	3.07	**3.2**
	fr-CA	3.23	3.05	3.09	**3.46**
	fr-FR	3.41	3.45	3.37	**3.47**
	it-IT	3.95	**4**	3.92	3.95
	nl-NL	3.03	3.08	2.85	**3.08**
	pt-BR	3.63	3.62	3.61	**3.64**
Germany German 3.96	da-DK	3.27	**3.32**	3.3	3.3
	en-AU	3.47	3.58	3.57	**3.63**
	en-GB	3.51	3.51	3.55	**3.66**
	en-IE	3.57	3.7	3.66	**3.82**
	en-IN	3.6	3.67	3.6	**3.77**
	en-US	3.62	**3.66**	3.52	**3.66**
	en-ZA	3.66	3.75	3.69	**3.8**
	es-ES	2.88	3.12	3.09	**3.44**
	es-MX	3.2	3.29	3.46	**3.55**
	fr-CA	3.16	3.25	3.28	**3.67**
	fr-FR	3.34	3.4	3.42	**3.59**
	it-IT	3.16	3.18	3.12	**3.57**
	nl-NL	3.43	3.55	3.51	**3.65**
	pt-BR	2.8	2.97	3.27	**3.55**
Total		3.39	3.44	3.44	**3.57**

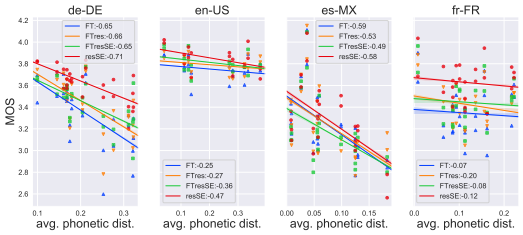


Figure 3: *Average crosslingual MOS per voice w.r.t average phonetic distance between voice and language*

of flaps in the French, Portuguese and German voices.

5. Discussion

5.1. Differences by language

There are two interesting observations from the crosslingual evaluation. The first one is the large variation in the MOS depending on the combination of target voice/language as shown in Table 4. One possible explanation for this is that the phonetic differences between the voice's language and the target language matters. Figure 3 shows the average MOS[3] of the voices/systems in the crosslingual evaluation with respect to the average phonetic distance between the speaker data and the test sentence of the target computed as

$$AvgPhoneDist = \sum_{\forall t \in T} P(t|T) \min_{\forall s \in S} dist(e_t, e_s) \quad (1)$$

where T and S are the sets of unique phones in the test utterances of the target language and in the target speaker data, respectively; e_t and e_s denote the phone embeddings for t and s taken from the look-up-table of the resSE model, and $dist$ is the cosine similarity function. Note that for all $t \in S$ the minimum distance is 0.

Figure 3 shows that the impact of the phonetic distance depends on the target language. For instance, Mexican subjects penalised foreign accented voices heavily, even when the average phonetic distance is small. On the contrary, for French subjects other factors seem to be more important. For example, for the British, Australian and Irish English voices, the factors that produce the most negative impact are very unnatural and strongly pronounced intonation, unnatural parsing of groups of words and pace which generally affected intelligibility. For the British English voices, the intelligibility is also affected by the porting of the non-rhotic character of British English to French: final /r/ are often dropped and the quality and length of the previous vowel is modified. The pronunciation of French diaeresis also seems problematic in terms of intelligibility, being realised as a diphthong (as in "pays" for instance). On the other hand, German, Italian, Portuguese, and Spanish voices were preferred in terms of general intelligibility, even though the intonation was found too monotone, the pauses sometimes incorrectly placed or missing, and the foreign accent too strong.

For American English, MOS is also strongly correlated with the average phonetic distance, but mainly due to the smaller dispersion. Otherwise, the curves are flatter than for German or Spanish. This links with the second observation which is that the average MOS for American English is higher than for the other languages. One explanation of that higher score is that an average of 7% of the 8500 training utterances of the non-English voices were in English, with another 13% having at least one English word. The English proficiency of the voice talents varied greatly, from fully bilingual to very accented. Moreover, the English utterances in the training data of many voices were transcribed using the phones of the voice's language, which might be the reason for the relatively larger phonetic distances for en-US. Still, that English data seems to have contributed to improve the synthesis of English utterances with non-English voices. Another possible explanation for the higher MOS for American English may be that subjects in that locale (and to some extent in France French too) are more used to listening to foreign accents than their Mexican or German counterparts and therefore, have a larger tolerance for them. Further experiments are needed to confirm which hypothesis is correct.

[3]The MOS values have been shifted so that the average MOS across all the samples of the two evaluation groups of each target language are the same

5.2. Pauses

One of the most commented problems for inlingual synthesis was errors with pausing. Since the model does not include any explicit pause predictor, or part-of-speech tagging, the pause prediction depends entirely on the phonetic transcription and punctuation marks. In single-language models, the network might be able to perform some level of syntactic parsing, for example identify the most common function words. In a multilingual framework, this is harder because the same phonetic sequence might also correspond to a content word in a different language. But also, different languages have different rules regarding the punctuation. So, whereas in some languages it is used mostly to indicate pausing, in others they have a more grammatical function. These kinds of differences are hard to disambiguate by just looking at the phone sequence. Including a LE was expected to help with such language-dependent issues. However, simply concatenating a global LE at the input of the attention didn't work.

6. Conclusions

This paper confirms that data from speakers in other languages can be used to compensate for the lack of target speaker data. We have presented a large-scale experiment on building neural TTS models by mixing speech from 30 speakers of 15 different locales in 8 different languages. The results show that for the vast majority of voices, fine-tuning a multi-lingual and multi-speaker model produces equal or better quality than single-speaker models trained with more than 2.5 times the amount of speaker-specific data.

An evaluation of these models synthesizing speech in a language different from that of the target speaker has confirmed that the models also preserve good multilingual capability. On average, the MOS on these models in a crosslingual scenario is around 80% of the MOS obtained by inlingual single-speaker native voices. Although this may not be enough for a general stand-alone voice in that language, it is sufficient for code-switching. Our results showed that although non fine-tuned voices are marginally better for crosslingual synthesis, for inlingual synthesis they are generally significantly worse than the fine-tuned ones. Finally, we have presented a qualitative analysis of the main problems identified by subjects during the inlingual and crosslingual evaluations.

7. References

[1] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "WaveNet: A Generative Model for Raw Audio," in *Arxiv*, 2016. [Online]. Available: https://arxiv.org/abs/1609.03499

[2] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan, R. A. Saurous, Y. Agiomvrgiannakis, and Y. Wu, "Natural TTS Synthesis by Conditioning Wavenet on MEL Spectrogram Predictions," in *Proc. ICASSP*, 2018, pp. 4779–4783.

[3] J. Yamagishi, T. Kobayashi, Y. Nakano, K. Ogata, and J. Isogai, "Analysis of Speaker Adaptation Algorithms for HMM-Based Speech Synthesis and a Constrained SMAPLR Adaptation Algorithm," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 1, pp. 66–83, 2009.

[4] V. Wan, J. Latorre, K. Yanagisawa, N. Braunschweiler, L. Chen, M. J. F. Gales, and M. Akamine, "Building HMM-TTS Voices on Diverse Data," *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, no. 2, pp. 296–306, 2014.

[5] Y. Fan, Y. Qian, F. K. Soong, and L. He, "Multi-speaker modeling and speaker adaptation for DNN-based TTS synthesis," in *Proc. ICASSP*, 2015, pp. 4475–4479.

[6] J. Park, K. Zhao, K. Peng, and W. Ping, "Multi-Speaker End-to-End Speech Synthesis," 2019. [Online]. Available: https://arxiv.org/abs/1907.04462

[7] J. Latorre, J. Lachowicz, J. Lorenzo-Trueba, T. Merritt, T. Drugman, S. Ronanki, and V. Klimkov, "Effect of Data Reduction on Sequence-to-sequence Neural TTS," in *Proc. ICASSP*, 2019, pp. 7075–7079.

[8] H.-T. Luong, X. Wang, J. Yamagishi, and N. Nishizawa, "Training Multi-Speaker Neural Text-to-Speech Systems using Speaker-Imbalanced Speech Corpora," 2019. [Online]. Available: https://arxiv.org/abs/1904.00771

[9] Y. Deng, L. He, and F. Soong, "Modeling Multi-speaker Latent Space to Improve Neural TTS: Quick Enrolling New Speaker and Enhancing Premium Voice," 2019. [Online]. Available: https://arxiv.org/abs/1812.05253

[10] E. Cooper, C. Lai, Y. Yasuda, F. Fang, X. Wang, N. Chen, and J. Yamagishi, "Zero-Shot Multi-Speaker Text-To-Speech with State-Of-The-Art Neural Speaker Embeddings," in *Proc. ICASSP*, 2020, pp. 6184–6188.

[11] E. Cooper, C.-I. Lai, Y. Yasuda, and J. Yamagishi, "Can Speaker Augmentation Improve Multi-Speaker End-to-End TTS?" 2020. [Online]. Available: https://arxiv.org/abs/2005.01245

[12] Q. Hu, E. Marchi, D. Winarsky, Y. Stylianou, D. Naik, and S. Kajarekar, "Neural Text-to-Speech Adaptation from Low Quality Public Recordings," in *Proc. 10th ISCA Speech Synthesis Workshop*, 2019, pp. 24–28. [Online]. Available: http://dx.doi.org/10.21437/SSW.2019-5

[13] I. Himawan, S. Aryal, I. Ouyang, S. Kang, P. Lanchantin, and S. King, "Speaker Adaptation of a Multilingual Acoustic Model for Cross-Language Synthesis," in *Proc. ICASSP*, 2020, pp. 7629–7633.

[14] Y. Zhang, R. J. Weiss, H. Zen, Y. Wu, Z. Chen, R. Skerry-Ryan, Y. Jia, A. Rosenberg, and B. Ramabhadran, "Learning to Speak Fluently in a Foreign Language: Multilingual Speech Synthesis and Cross-Language Voice Cloning," in *Proc. Interspeech*, 2019, pp. 2080–2084.

[15] M. Chen, M. Chen, S. Liang, J. Ma, L. Chen, S. Wang, and J. Xiao, "Cross-Lingual, Multi-Speaker Text-To-Speech Synthesis Using Neural Speaker Embedding," in *Proc. Interspeech*, 2019, pp. 2105–2109.

[16] H. Zen, N. Braunschweiler, S. Buchholz, M. J. F. Gales, K. Knill, S. Krstulovic, and J. Latorre, "Statistical Parametric Speech Synthesis Based on Speaker and Language Factorization," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 6, pp. 1713–1724, 2012.

[17] B. Li and H. Zen, "Multi-Language Multi-Speaker Acoustic Modeling for LSTM-RNN Based Statistical Parametric Speech Synthesis," in *Proc. Interspeech*, 2016, pp. 2468–2472.

[18] Y. Fan, Y. Qian, F. K. Soong, and L. He, "Speaker and language factorization in DNN-based TTS synthesis," in *Proc. ICASSP*, 2016, pp. 5540–5544.

[19] Q. Yu, P. Liu, Z. Wu, S. K. Ang, H. Meng, and L. Cai, "Learning cross-lingual information with multilingual BLSTM for speech synthesis of low-resource languages," in *Proc. ICASSP*, 2016, pp. 5545–5549.

[20] K. R. Prajwal and C. V. Jawahar, "Data-Efficient Training Strategies for Neural TTS Systems," in *8th ACM IKDD CODS and 26th COMAD*. Association for Computing Machinery, 2021, p. 223–227. [Online]. Available: https://doi.org/10.1145/3430984.3431034

[21] W. Ping, K. Peng, A. Gibiansky, S. O. Arik, A. Kannan, S. Narang, J. Raiman, and J. Miller, "Deep Voice 3: Scaling Text-to-Speech with Convolutional Sequence Learning," 2018. [Online]. Available: https://arxiv.org/abs/1710.07654

[22] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "librosa: Audio and music signal analysis in python," in *Proceedings of the 14th python in science conference*, vol. 8, 2015.

[23] M. He, Y. Deng, and L. He, "Robust Sequence-to-Sequence Acoustic Modeling with Stepwise Monotonic Attention for Neural TTS," 2019. [Online]. Available: https://arxiv.org/abs/1906.00672

[24] S. Maiti, E. Marchi, and A. Conkie, "Generating Multilingual Voices Using Speaker Space Translation Based on Bilingual Speaker Data," in *Proc. ICASSP*, 2020, pp. 7624–7628.

[25] L. Xue, W. Song, G. Xu, L. Xie, and Z. Wu, "Building a Mixed-Lingual Neural TTS System with Only Monolingual Data," in *Proc. Interspeech*, 2019, pp. 2060–2064.

[26] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," 2017.

[27] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention Is All You Need," 2017.

[28] J. C. Wells, "Computer-coding the IPA: a proposed extension of SAMPA," 1995, [Online; accessed 28-January-2018]. [Online]. Available: https://www.phon.ucl.ac.uk/home/sampa/ipasam-x.pdf

[29] N. Kalchbrenner, E. Elsen, K. Simonyan, S. Noury, N. Casagrande, E. Lockhart, F. Stimberg, A. van den Oord, S. Dieleman, and K. Kavukcuoglu, "Efficient Neural Audio Synthesis," in *Proceedings of the 35th International Conference on Machine Learning*, 2018, pp. 2410–2419. [Online]. Available: http://proceedings.mlr.press/v80/kalchbrenner18a.html

[30] J. Lorenzo-Trueba, T. Drugman, J. Latorre, T. Merritt, B. Putrycz, R. Barra-Chicote, A. Moinet, and V. Aggarwal, "Towards Achieving Robust Universal Neural Vocoding," in *Proc. Interspeech*, 2019, pp. 181–185.

[31] D. Paul, Y. Pantazis, and Y. Stylianou, "Speaker Conditional WaveRNN: Towards Universal Neural Vocoder for Unseen Speaker and Recording Conditions," in *Proc. Interspeech*, 2020, pp. 235–239.

[32] T. Capes, P. Coles, A. Conkie, L. Golipour, A. Hadjitarkhani, Q. Hu, N. Huddleston, M. Hunt, J. Li, M. Neeracher, K. Prahallad, T. Raitio, R. Rasipuram, G. Townsend, B. Williamson, D. Winarsky, Z. Wu, and H. Zhang, "Siri On-Device Deep Learning-Guided Unit Selection Text-to-Speech System," in *Proc. Interspeech*, 2017, pp. 4011–4015.

11th ISCA Speech Synthesis Workshop (SSW 11)
26-28 August 2021, Budapest, Hungary

Methods of slowing down speech

Christina Tånnander[1,2], *Jens Edlund*[1]

[1]KTH, Speech, Music and Hearing
[2]Swedish Agency for Accessible Media
christina.tannander@mtm.se, edlund@speech.kth.se

Abstract

A slower speaking rate of human or synthetic speech is often requested by for example language learners or people with aphasia or dementia. Slow speech produced by human speakers typically contain a larger number of pauses, and both pauses and speech have longer segment durations than speech produced at a standard or fast speaking rate.

This paper presents several methods of prolonging speech. Two speech chunks of about 30 seconds each, read by a professional voice talent at a very slow speaking rate, were used as reference. Seven pairs of stimuli containing the same word sequences were produced, one by the same professional, reading at her standard speaking rate and six by a moderately slow synthetic voice trained on the same human voice. Different combinations of pause insertions and stretching were used to match the total length of the corresponding reference stimulus. Stretching was applied in different proportions to speech and non-speech, and pauses were inserted at punctuations, at certain phrase boundaries, between each word, or by copying the pause locations of the reference reading.

128 crowdsourced listeners evaluated the 16 stimuli. The results show that all manipulated readings are less consistent with expectations of slow speech than the reference, but that the synthesised readings are comparable to stretched human speech. Key factors are the relation between speech and silence and the duration of talkspurts.

1. Introduction

Language learners and people with cognitive impairments (e.g. aphasia or dementia) often prefer a slower speaking rate when listening to longer texts read aloud. A number of studies have attempted to find a balance between speaking rate and comprehension among aphasics (see for example [2]–[4]).

The most obvious method to produce read speech with a slow speaking rate is to instruct a voice talent to read very slowly, but to create books or other long texts at different speaking rates with human readings would be prohibitively expensive. Using slow speech synthesis trained on or adapted to very slow speech materials is another option, but again, this may not always be practicable.

In this study, we investigate several methods to create very slow speech using existing speech materials. The results are evaluated against human reference readings at a very slow speaking rate by a professional voice talent, and a stretched human reading in a listening experiment with 128 crowd sourced listeners.

2. Background

Talking books

We are mainly concerned, here, with texts read aloud for people with vision impairments or reading difficulties: societal information, news, and so-called *talking books*. The difference between an audiobook and a talking book varies somewhat in different countries, but is often present and similar in meaning. The Swedish Agency for Accessible Media (MTM) states that a talking book "is intended for persons with a permanent or temporary print disability", that they are "produced with public funds and in accordance with Section 17 of the Copyright Act", and that the "the recording of a talking book must conform with the original, which must be a published work" [1]. In other words, the option to simplify or otherwise change the written text to make it easier to understand is available.

Speaking rate

The speed at which speech is produced can be measured in different ways. *Speaking rate* is defined as the rate at which a certain idealised (e.g. phonological or orthographic) unit is produced per total speech and non-speech (silences, breath etc.) duration [5]. *Articulation rate*, on the other hand, is the number of actual speech units (e.g. phonemes) divided by the duration of the actual speech, non-speech such as silences and breathings excluded [6]. In speech science, *speech rate* is sometimes used with the same meaning as speaking rate, and sometimes as a metric more closely tied to the acoustic signal and its variations.

Both speaking and articulation rate can be measured in different ways, for example the number of syllables per time unit, such as syllables per second [5], [7], [8], syllables per minute [9] or average syllable duration [10]. Another common metric of speaking rate is words per minute (wpm). Since word length differ between languages and speaking situations, wpm can be a too rough metric in many situations. In research, wpm is often presented alongside metrics that reflect the pronunciation of the words and data about average syllables per word [9].

Measuring speaking rate is not trivial and even if researchers use the same metrics, the counting of words, syllables or phones can differ. There is no obvious unified way to count phones or other speech units, for example glottal stops, affricates, diphthongs or syllabic consonants [11]. Also, the number of phones or syllables can be differentiated into the *intended* number of speech units and the number of units that are actually *realized.* It has been shown that listener's perception of speaking rate reflects both the intended and realized speaking rates [12].

10.21437/SSW.2021-8

Slow speech

Slow speech is characterised by a larger number of pauses, longer pause durations and longer phone durations [5]. [6] found that the articulation rate makes up only a small part of the changes in speaking rate, and the largest change was the total pause durations. Other things affect speaking rate as well. There is evidence of regional variations in articulation rate [7], [13], and the number of pauses inserted in read speech can depend on text genre (e.g. news reports and novels) [14]. Pauses tend to be longer the more syllables there are in the utterance [14]–[16].

The purpose of a recording can also be seen in speech characteristics. TTS recordings, for example, have been characterized as having low speaking rate as well as low mean pitch and standard deviation of energy [17], and spontaneous speech as faster than read speech, with a greater variance [11].

Simply stretching speech while maintaining F_0 and other characteristics is, unsurprisingly, not consistent with human speech. In humans, a slower speaking rate correlates with a lower F_0 [18], hyperarticulated speech is characterized by a slower speaking rate, a higher number of pauses, more syllables, which altogether result in a longer total duration of speech and non-speech [8]. Perceived speaking rate is also affected by non-durational characteristics: [19] found that high, fairly monotonous speech segments lead to a higher perceived speaking rate.

Typical speaking rate

In British English, the speaking rate vary between 140 (lecture) to 210 wpm (conversation), with corresponding syllables per second of 190 and 260 [9]. Similarly, a summary of different acoustic features among different English speech corpora shows that the lowest speaking rate was found in audiobooks, followed by recordings for TTS and broadcast news, while corpora consisting of conversational speech show a higher speaking rate [17]. Proficiency matters, too. A study investigating pausing among English language learners reported that native speakers pause 7.15 times per hundred words (phw), while the learners pause much more frequently, between 10.76 to 14.43 phw, depending on proficiency.

[20] reported that a Swedish professional speaker had a speaking rate of 130 wpm in normal mode, 111 in slow mode and 106 wpm in distinctive mode (146 in fast mode). These variations were mainly associated with total pause durations (longer pauses and a larger number of pauses). At a slow speaking rate, the sum of the pause durations was almost 50% longer than at a normal speaking rate, while the phoneme durations differed only by 4%.

Controlling speaking rate in speech synthesis

Modern, unsupervised methods for training speech synthesis often capture prosody well. It does so behind the scenes, leaving limited room for investigation or control for the researcher. Control of prosody, or the lack thereof, is a well-known issue and an active research area, and in some cases, the investigation of prosody is the very reason for creating a synthetic voice. [21] controlled expressiveness and sentence wise speaking rate without losing quality and naturalness. [22] facilitated the independent control of pitch, pitch range, phone durations, energy and spectral tilt by including these in their model, but their evaluation showed a significant decrease in MOS score when slowing down or speeding up the voice. This may have been a result of an overly generic evaluation question, confounding for example a dispreference for slow speech with a poor quality rating for slow speech.

3. Method

Participants

Listeners were recruited through Prolific, a subject pool for online experiments [23]. At the time of the experiment, Prolific had 815 active subjects between the ages of 18 and 67 reporting as fluent in Swedish. We recruited 64 of these for each of two utterances, totalling 128 sessions, and paid marginally above the recommended fee. Each test took between 5 and 6 minutes to complete and listeners were rewarded £0.8. Listeners were allowed to take part in both studies, but only once in each.

Experiment platform

A prototype listening test platform at the Swedish national research infrastructure Språkbanken Tal was used. The platform is fully WCAG 2.1 [24] compliant and presents a single stimuli (sound file) per page. Listeners were guided through their test and then returned to the Prolific web site. Only a very small number of listeners (<3%) timed out or returned their task undone.

Texts and reference stimuli

Two Swedish texts, **TEXT1** and **TEXT2**, each containing two sentences from a campaign concerning covid-19 information, were used, see Table 1. A recording of the texts was already available in a typical speaking rate, and the same voice talent was employed to rerecord the sentences at a very slow speaking rate. The results of these slow recordings were used as references (**REF1** and **REF2**).

Table 1. *Number of sentences, words, syllables and minor delimiters in the two texts.*

Text	Sentences	Words	Syllables	Minor delimiters
TEXT1	2	31	57	1
TEXT2	2	33	66	2

Stimuli

The human stimuli were based on the human recordings reading **TEXT1** and **TEXT2** at a typical speaking rate and at a slow speaking rate. The duration and articulation rates of these files are shown in Table 2. To illustrate the temporal aspects of the texts using the synthetic voice used in the stimuli creation, the data from a synthesised reading with pauses at major and minor delimiters is included in the table.

Table 2. *Duration (seconds) and articulation rate (syllables/second) for the human recordings and synthesis (with pauses at major and minor delimiters).*

	TEXT1		**TEXT2**	
	Dur.	**Art. rate**	**Dur.**	**Art. rate**
Human normal	17	2,08	18,7	1,97
Human slow (**REF**)	30,5	3,90	32,5	3,58
TTS	20,4	3,13	22,7	2,97

Table 3. *A description of the eight stimuli used in the study: pause placements, prolongation (*STRETCHED *means the stretching of the whole file,* PAUSES *the prolongation of non-specch and* BOTH *combines* PAUSES *and* STRETCHED*), Number of pauses, proportions of non-speech, and average pause durations for both texts.*

	Pause placements	Pro-longation	Number of pauses		Non-speech (%)		Avg. pause duration	
			TEXT1	TEXT2	TEXT1	TEXT2	TEXT1	TEXT2
REF	HUMSLOW	NONE	16	15	19,67	20,62	330	381
HUMSTRETCHED	HUMAN	STRETCHED	6	7	11,80	14,15	457	507
TTSSTRETCHED	ORTHOGRAPHIC	STRETCHED	2	4	10,16	13,85	925	792
TTSORTHOPAUSES	ORTHOGRAPHIC	PAUSES	2	4	35,08	38,46	4933	2895
TTSDESIGNEDPAUSES	DESIGNED	PAUSES	8	7	43,93	40,31	1600	1736
TTSWORDPAUSES	WORD	PAUSES	30	32	37,05	34,46	342	309
TTSHUMPAUSES	HUMSLOW	PAUSES	16	15	40,00	39,08	710	781
TTSHUMPAUSESSTRETCHED	HUMSLOW	BOTH	16	15	22,95	21,54	360	382

REF1 and **REF2** were of about 30 seconds duration each, and in order to eliminate effects of durational variation in the evaluation, all manipulated stimuli were made to match these durations. **HUMSTRETCHED1** and **HUMSTRETCHED2** were created by stretching the typical human recordings to the same duration as the slow readings.

For the synthesized stimuli, we trained a voice with Nvidia's PyTorch implementation of Tacotron and WaveGlow on nearly 18 hours of female speech data from the same voice talent as the recorded data in **REF1** and **REF2**, originally recorded for unit selection synthesis [25][26]. The words in the training data were split into five relative speaking rate categories. To ensure there were enough speech data in each category, they were balanced to contain approximately the same number of words (27 000). Each word was prepended with its speaking rate category in the training. This makes it possible to synthesize at five different speaking rates, by inserting the speaking rate category before each word in the input to the synthesizer. The slowest speaking rate, along with hyper-articulated phonemic transcriptions, was used for all synthetizations in this study. Note that the slow synthetic data created in this manner is not nearly as slow as **REF1** and **REF2**, simply because the voice is not trained on deliberately slow speech (see Table 2).All stimuli are available at http://www.sprakbanken.speech.kth.se/surveys/slow/.

Four different *pause placements* were used: **ORTHOGRAPHIC** pauses were inserted at major and minor delimiters in the orthography (e.g. commas and stops); **DESIGNED** pauses were inserted at selected syntactic boundaries aiming for equally-sized speech chunks (other policies are possible); **WORD**, where pauses were inserted between all words; and finally **HUMANSLOW**, where we copied the locations of perceptual pauses (>120 ms of non-speech [27]) in the **REF1** and **REF2**. The initial pause durations were what came out of the synthesis, and all versions were still shorter than the corresponding **REF1** and **REF2**.

The stimuli were synthesized with pauses in the locations described, and the duration of each pause was manually manipulated by inserting (or sometimes deleting) silence copied from the same file. For the pause locations in **TTSORTHOPAUSES**, **TTSDESIGNEDPAUSES** and **TTSHUMPAUSES** the pause durations were altered to match the proportion of the same pause location in the **REF** files. For **TTSWORDPAUSES**, we kept the original pause durations between each word from the synthetization, and manipulated the pause locations that also occurred in the **REF** files proportionally, to end up at the file durations of the **REF** files. Finally, **TTSHUMPAUSESSTRETCHED** were first given the same pause durations as the **REF** files, then the entire files were stretched to the required durations. The details of the resulting stimuli are presented in Table 3, and a visualization of the speech/non-speech patterns of the readings of **TEXT1** is shown in Figure 1.

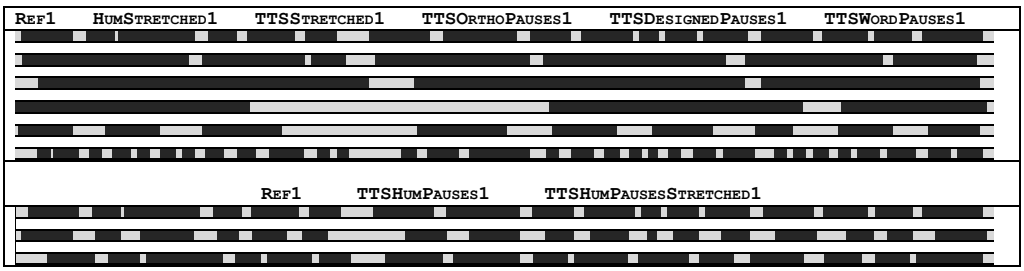


Figure 1. *Chronogram of speech (black segments) and non-speech (grey segments) for Text1.First section from top to bottom:* **REF1**, **HUMSTRETCHED1**, **TTSSTRETCHED1**, **TTSORTHOPAUSES1**, **TTSDESIGNEDPAUSES1**, **TTSWORDPAUSES1**. *Second section:* **REF1** *(repeated),* **TTSHUMPAUSES**, **TTSHUMPAUSESSTRETCHED**.

Procedure

In order to avoid overly long sessions and listening fatigue, we divided the stimuli in two different tests, each containing all 8 versions of one of the two texts. For each test, the order of the stimuli was varied systematically. A single listener could participate in both tests, but only once per test. For each stimulus, they were presented with the framing sentence "Imagine that you have requested a short text to be read for you *very slowly*." and the question "How well does this reading match your expectations?". The response alternatives were a five-grade scale with the option "Matches very well", "Matches well", "Matches neither well nor poorly", "Matches poorly" and "Matches very poorly". We use **ACCEPTANCE** for this variable.

4. Analysis & results

102 recruits started the listening study. 3 did not finish, and the experiment was stopped when 8 recruits had responded to each test set in each systematically varied order. The listening times varied between 5 and 6 minutes. 29 listeners participated in both studies (i.e. judged both texts once) and 70 took part in one study only.

A one-way ANOVA showed statistically-significant difference in **ACCEPTANCE** by stimuli identity ($f(15)=12.088$, $p < 0.001$). Pairwise comparisons showed no significant difference within any pair of the same stimuli type but different texts. This let us combine **TEXT1** and **TEXT2**, so that we considered only stimuli type – the manner in which the stimulus was created. One-way ANOVA again showed statistically-significant difference in **ACCEPTANCE** by stimuli type ($f(7)=22.664$, $p < 0.001$).

Post-hoc pairwise t-tests using the Bonferroni correction were performed across all pairs of stimuli types. There was a significant difference between **REF** and all other stimuli types. In all, 14 pairs were significant. These pairs and the effect sizes are presented in Table 4.

For good measure, the difference between the two human readings (**REF** and **HUMSTRECTCHED**) and the other stimuli were verified with Dunnett's test for comparing several treatments with a control. All synthesized stimuli were significantly different than **REF** at the 0.001 level, and only **TTSORTHOPAUSES** was differed from **HUMSTRETCHED**, again at the .001 level.

Finally, we performed a Tukey's test for all pairs of stimuli type. This singled out the same 14 pairs as significantly different, at the same levels as the repeated t-tests with Bonferroni correction.

The literature, our intuition from listening to the stimuli, and the initial results all hint at a combination of the duration of talkspurts and their frequency as being key to slow speech (note that the speech/pause ratio and other similar metrics can be derived from these two measures). As talkspurts can clearly be both too long and too short, and their frequency too high or too low, quadratic polynomials we fitted to their averages (**AVTSDUR** and **AVTSFREQ**) for all stimuli. As expected, both significantly predict **ACCEPTANCE**, and there are interaction effects. The additive model's F statistic is 18.181*** (df = 4; 1019), and the corresponding multiplicative model yields 14.561*** (df = 8; 1015). Adjusted R2 is 0.063 and 0.096, respectively.

Table 4. *Each row describes the stimuli listed in the leftmost column, starting with the number of judgements, the average, and the standard deviation. The last three columns contain the significant pairwise comparisons, with (1) representing* **REF**, *(2) representing* **TTSORTHOPAUSES**, *and (3)* **TTSDESIGNEDPAUSES**. *Each cell shows effect size and significance (0,05=*, 0.01=**, 0.005=***,0.001=****)*

	N	Avg	SD	(1)	(2)	(3)
REF (1)	128	4,2	1,0	(1)	(2)	(3)
HUMSTRETCHED	128	3,1	1,4	-0.9 (L) ****	0.50 (M) **	-
TTSSTRETCHED	128	3,2	1,3	-0.91 (L) ****	0.61 (M) ****	-
TTSORTHOPAUSES (2)	128	2,4	1,2	-1.6 (L) ****	n/a	-
TTSDESIGNEDPAUSES (3)	128	2,8	1,2	-1.3 (L) ****	-	n/a
TTSWORDPAUSES	128	3,3	1,2	-0.85 (L) ****	0.69 (M) ****	0.41 (S) *
TTSHUMPAUSES	128	3,2	1,1	-0.96 (L) ****	0.66 (M) ****	-
TTSHUMPAUSES STRETCHED	128	3,4	1,2	-0.75 (M) ****	0.80 (M) ****	0.51 (M) **

5. Discussion

The reference readings score higher than all other readings on the question of how well it corresponds to expectations of a very slow reading. This is to be expected, not only because it is read by a human professional who has been instructed to read very slowly, but because none of the other readings are designed, originally, to create very slow speech.

Compared to the typically-paced and stretched human reading **HUMSTRETCHED**, only two of the TTS varieties perform significantly worse: **TTSORTHOPAUSES**, in which only the very few orthographic pauses (commas and full stops) are extended to reach the duration of the reference utterances. These readings were included in part as a test case to see that the crowd workers behaved as could be expected, and in part to highlight the fact that pause lengthening has an upper bound. **TTSORTHOPAUSES** is judged as significantly worse than 6 out of the 7 other readings. Finally the reading with pauses inserted between constituents at regular intervals, **TTSDESIGNEDPAUSES**, fares poorly against the two highest ranked TTS readings, but the effect is small.

Turning to the average scores, the reference utterances stand out with a 4.2 average, as does **TTSORTHOPAUSES** with 2.4. The rest of the readings receive scores slightly above 3, with the stretched human voice ending up somewhere in the middle. Having listened to the stimuli, we propose that with the exception of **TTSORTHOPAUSES**, the stimuli are designed to be as pleasant to listen to as possible. The one other exception, perhaps, is **TTSWORDPAUSES**, with a pause between every single word. We did not expect this to be a viable solution, but having listened to the result ourselves, it really does not sound bad.

6. Conclusions & future work

The goal of this study has been to see if acceptable slow speech can be created with relatively simple means, without rerecording databases. Out of six different methods of prolonging synthesised utterances to match, in duration, *very* slow human speech, five achieved the same rating as original human speech that had been stretched. This is promising.

The results suggest that the relation between speech and non-speech play a role: about 10% of the variation in **ACCEPTANCE** is explained by a regression model based on these factors, in spite of the materials being highly varied in nature and not at all varied systematically in terms of speech/non-speech relation. As mentioned in the discussion, the literature supports this finding, and the very poor acceptance of **TTSORTHOPAUSES** is perhaps related to the uncomfortable pauses Sacks et al call "lapse"[28], the minimum duration of which Jefferson and others have approximated to 1 second [29].

We believe that we now have the tools to create workable very slow speech using only moderately slow speech synthesis, by manipulating the placement and durations of pauses, and the next step is a structured study of the relation between talkspurt durations and pause durations.

7. Acknowledgements

The survey was partly funded by Vinnova (2018-02427). The results will be made more widely accessible through the Swedish Research Council funded national infrastructure Nationella språkbanken and Swe-Clarin (2017-00626).

References

[1] 'Talking books', *Swedish Agency for Accessible Media*. https://www.mtm.se/english/products-and-services/talking-books/ (accessed Apr. 28, 2021).

[2] S. E. Blumstein, B. Katz, H. Goodglass, R. Shrier, and B. Dworetsky, 'The Effects of Slowed Speech on Auditory Comprehension in Aphasia', *Brain and Language*, vol. 24, pp. 246–265, 1985.

[3] K. Hux, J. A. Brown, S. Wallace, K. Knollman-Porter, A. Saylor, and E. Lapp, 'Effect of Text-to-Speech Rate on Reading Comprehension by Adults With Aphasia', *Am J Speech Lang Pathol*, vol. 29, no. 1, pp. 168–184, Feb. 2020, doi: 10.1044/2019_AJSLP-19-00047.

[4] J. E. Sung *et al.*, 'Real-time Processing in Reading Sentence Comprehension for Normal Adult Individuals and Persons with Aphasia', *Aphasiology*, vol. 25, no. 1, pp. 57–70, 2011, doi: 10.1080/02687031003714434.

[5] F. Goldman-Eisler, 'The Determinants of the Rate of Speech Output and their Mutual Relations', *Journal of Psychosomatic Research*, vol. 1, pp. 137–143, 1956.

[6] F. Goldman-Eisler, 'The Significance of Changes in the Rate of Articulation', *Lang Speech*, vol. 4, no. 3, pp. 171–174, Jul. 1961, doi: 10.1177/002383096100400305.

[7] E. Jacewicz, R. A. Fox, C. O'Neill, and J. Salmons, 'Articulation rate across dialect, age, and gender', *Lang Var Change*, vol. 21, no. 2, pp. 233–256, Jul. 2009, doi: 10.1017/S0954394509990093.

[8] B. Picart, T. Drugman, and T. Dutoit, 'Analysis and Synthesis of Hypo and Hyperarticulated Speech', in *Speech Synthesis Workshop (SSW)*, Kyoto, Japan, 2010, vol. 28, pp. 687–707, doi: https://doi.org/10.1016/j.csl.2013.04.008.

[9] S. Tauroza and D. Allison, 'Speech Rates in British English', *Applied Linguistics*, vol. 11, no. 1, pp. 90–105, 1990, doi: 10.1093/applin/11.1.90.

[10] T. H. Crystal and A. S. House, 'Articulation rate and the duration of syllables and stress groups in connected speech', *The Journal of the Acoustical Society of America*, vol. 88, no. 1, pp. 101–112, 1990.

[11] J. Trouvain, J. Koreman, A. Erriquez, and B. Braun, 'Articulation Rate Measures and Their Relation to Phone Classification in Spontaneous and Read German Speech', in *Adaptation-2001*, 2001, pp. 155–158.

[12] J. Koreman, 'Perceived speech rate: The effects of articulation rate and speaking style in spontaneous speech', *The Journal of the Acoustical Society of America*, vol. 119, pp. 582–96, Feb. 2006, doi: 10.1121/1.2133436.

[13] A. Leemann, 'Analyzing geospatial variation in articulation rate using crowdsourced speech data', *Journal of Linguistic Geography*, vol. 4, no. 2, pp. 76–96, 2016, doi: https://doi.org/10.1017/jlg.2016.11.

[14] G. Fant, A. Kruckenberg, and J. Barbosa, 'Individual variations in pausing. A study of read speech', presented at the Fonetik, Umeå, Sweden, 2003.

[15] J. Dankovicova, 'Articulation Rate Variation within the Intonation Phrase in Czech and English', in *Proceedings of the XIVth International Congress of Phonetic Sciences*, San Francisco, 1999, p. 4.

[16] B. Lindblom, 'Some Temporal Regularities of Spoken Swedish', in *Auditory Analysis and Perception of Speech*, Elsevier, 1975, pp. 387–396.

[17] E. Cooper, E. Li, and J. Hirschberg, 'Characteristics of Text-to-Speech and Other Corpora', in *9th International Conference on Speech Prosody 2018*, 2018, pp. 690–694, doi: 10.21437/SpeechProsody.2018-140.

[18] K. J. Kohler, 'Parameters of Speech Rate Perception in German Words and Sentences: Duration, Fo Movement, and Fo Level', *Lang Speech*, vol. 29, no. 2, pp. 115–139, Apr. 1986, doi: 10.1177/002383098602900202.

[19] A. C. M. Rietveld and C. Gussenhoven, 'Perceived speech rate and intonation', *Journal of Phonetics*, vol. 15, no. 3, pp. 273–285, Jul. 1987, doi: 10.1016/S0095-4470(19)30571-6.

[20] G. Fant, A. Kruckenberg, and L. Nord, 'Temporal organization and rhythm in Swedish', in *Proceedings of the XIIth ICPhS*, Aix-en-Provence, France, 1991, pp. 251–256.

[21] S. Shechtman and A. Sorin, 'Sequence to Sequence Neural Speech Synthesis with Prosody Modification Capabilities', presented at the Speech Synthesis Workshop (SSW 10), Vienna, Austria, 2019, doi: 10.21437/SSW.2019-49.

[22] T. Raitio, R. Rasipuram, and D. Castellani, 'Controllable Neural Text-to-Speech Synthesis Using Intuitive Prosodic Features', presented at the Interspeech, Sep. 2020, doi: DOI: 10.21437/Interspeech.2020-2861.

[23] S. Palan and C. Schitter, 'Prolific.ac - A subject pool for online experiments', *Journal of Behavioral and Experimental Finance*, vol. 17, pp. 22–27, 2017.

[24] 'Web Content Accessibility Guidelines (WCAG) 2.1', *W3C*. https://www.w3.org/TR/WCAG21/ (accessed Apr. 28, 2021).

[25] C. Tånnander, 'Speech Synthesis and evaluation at MTM', in *Proceedings of Fonetik*, Gothenburg, Sweden, 2018, pp. 75–80,

[26] C. Tånnander and J. Edlund, 'Stress manipulation in text-to-speech synthesis using speaking rate categories. Fonetik, Lund, Sweden, submitted.

[27] M. Heldner, 'Detection thresholds for gaps, overlaps, and no-gap-no-overlaps', *JASA*, vol. 130, no. 1, pp. 508–513, Jul. 2011, doi: 10.1121/1.3598457.

[28] 'A Simplest Systematics for the Organization of Turn Taking for Conversation', in *Studies in the Organization of Conversational Interaction*, Academic Press, 1978, pp. 7–55.

[29] G. Jefferson, 'Preliminary notes on a possible metric which provides for a "standard maximum" silence of approximately one second in conversation', in *Conversation: An interdisciplinary perspective*, Clevedon, England: Multilingual Matters, 1989, pp. 166–196.

11th ISCA Speech Synthesis Workshop (SSW 11)
26-28 August 2021, Budapest, Hungary

Personality in the mix - investigating the contribution of fillers and speaking style to the perception of spontaneous speech synthesis

Joakim Gustafson, Jonas Beskow, Éva Székely

Division of Speech, Music and Hearing, KTH Royal Institute of Technology, Stockholm, Sweden

jocke@speech.kth.se, beskow@kth.se, szekely@kth.se

Abstract

Studies on human-human interactions have shown that that the fluency of a speaker influences the perception of personality. Adding fillers and discourse markers can make the speaker seem uncertain, more casual and spontaneous. With recent TTS developments it is now possible to investigate if the same holds for artificial speakers. In a previous experiment, it was shown that local insertion of fillers in a regular TTS voice influenced the perceived personality. In the current study we extend that work in two ways: Firstly, we recreate the English experiment adding a voice trained on spontaneous speech, where adding fillers also has a global effect on the synthesized speech. We also add Swedish read and spontaneous voices. Secondly, for the Swedish voices, we investigate the effect of using a multi-speaker model mixing a read speech voice and a spontaneous speech voice when generating disfluent synthetic speech.

Index Terms: spontaneous speech synthesis, personality traits, speaking styles, fillers

1. Introduction

The way people speak in conversation is dependent both on extralinguistic factors like age, gender, dialect and personality, and on situation-dependent factors, like affective state, cognitive load and feedback from the listener. Hence, the actual realization of a spoken utterance influences how listeners perceive the speaker, both in terms of personality and cognitive state. Filler words like filled pauses ("uh") and discourse markers ("you know") play an important role in communicating these in spontaneous speech. Filled pauses have been viewed in three ways [1]: as a floor-holding signal [2], as interjections [3], where .e.g. "um" has been found to announce a longer delay in the upcoming speech than "uh" [4], and as symptoms to a planning problem [5]. Thus, filled pauses appear to be useful for the listener in conversations: as a turn-handling cue [6], to improve comprehension [7, 8] and to understand the speaker's certainty of what they are saying [9]. The usage patterns of filled pauses have been found to vary with nationality, age, gender and socio-economic class [10]. Filled pauses have been found to influence the perception of personality traits like neuroticism, and extensive use of filled pauses have been rated negatively as unprepared, unsophisticated, and insecure [11].

Discourse markers are often used to indicate the speaker's stance. Depending on the speaker, context and prosodic realisation "you know" and "I think" can express both confidence and uncertainty, seeking confirmation of understanding from the listener [12]. "I mean" and "like" have been found to act as fillers, as hedging devices to what is being said [13] and to mark modification of what was previously said [14]. Discourse markers such as "like" are more common for younger speakers and in loose talk, where it is produced in the middle of fast and fluent speech [15]. They have been found to be markers of conscientiousness [16], as well as casualness, solidarity, politeness and spontaneity [17, 15, 18].

Prosodic features like pitch and speaking rate also influence the perception of personality [19]. Extraversion is associated with fast speaking rate and a wide pitch range, while introversion is perceived in slow, soft, deep and monotone voices [20]. Speaking rate influences the perception of several speaker traits, where slower speech is perceived as older in age [21] and more introvert [22], while faster speech is associated with higher knowledge and social attractiveness [23], greater persuasiveness [24], and higher competence and dominance [25]. In the current study, we aim to investigate how fillers and speaking style influence the perception of personality in read and spontaneous speech synthesis.

2. Related work

There have been several investigations in making read speech synthesis more spontaneous and expressive by automatically inserting fillers in its text input [26, 27, 28]. In order to make a diphone unit selection synthesizer more suitable for generating fillers, spontaneous speech utterances have been supplemented to its read speech training corpus [29]. Recently, [30] introduced a spontaneous speech synthesizer trained on a conversational podcast corpus, that could automatically insert and synthesize natural sounding fillers.

There have been some previous efforts in synthesizing voices with personality: a diphone synthesis voice was made more extrovert by providing it with the stereotypical extrovert features: high loudness, increased pitch, a great frequency range and a fast speaking rate [31]. In a project that developed voices for a speech-enabled computer game that features fairy-tale characters with different personalities, both speaking rate modifications and insertions of fillers was used [32].

The current paper builds on a previous study by Wester et al., where filled pauses where added to a read speech unit selection synthesizer in order to alter the perceived personality of the voice [33]. The authors found that adding fillers makes the artificial voice sound more neurotic, less open, less extrovert and less conscientious. In a follow-up study, they also investigated the effect of synthesis method and voice quality on the perceived personality and naturalness [34]. The result showed that increased voice quality enhances the personality the text conveyed, but it does not alter it to another personality. In this study we extend their work by using a state-of-the-art neural sequence-to-sequence speech synthesizer built from a spontaneous speech corpus. The main contributions of this work are that we extend their perceptual experiment on read speech to spontaneous speech synthesis, and that we investigate the perceptual effect of training a multi-speaker model, that allows us to mix between a read speech voice and a spontaneous speech voice when generating disfluent synthetic speech.

10.21437/SSW.2021-9

3. Speech synthesizers

In this paper we carry out studies on read and spontaneous speech synthesis in English and Swedish. For the English read speech synthesis we use the female Scottish CereVoice unit selection synthesis voice Heather. The English spontaneous speech voice and both Swedish voices are built using a PyTorch implementation[1] of Tacotron 2 [35]. The voices were trained using transfer learning for 200k iterations on top of a pre-trained model trained on large (ca. 20 hours) read speech corpora in English and Swedish. For vocoding, we fine-tuned the pre-trained universal model of WaveGlow to the English and Swedish conversational corpora [36].

The English spontaneous speech corpus is created from the audio recordings of the Trinity Speech-Gesture Dataset (TSGD) [37], which is comprised of 25 impromptu monologues by a male Irish actor. In each session (ca 10 minutes long) the actor tells a listener in the room about his hobbies, daily activities, and interests. The Swedish spontaneous speech corpus consist of 6 hours of speech extracted from a conversational podcast recorded by a male Swedish comedian. In the podcast, the co-median makes sandwiches and tell stories to his co-host. The data is very spontaneous and includes a lot of laughter and overlapping speech, which had to be removed from the TTS corpus before training the voice. Both spontaneous corpora were transcribed using ASR and subsequently manually corrected, to ensure that all fillers are transcribed accurately. Segmentation was done automatically into breath groups (stretches of speech delineated by breath events) using a deep learning-based breath detector described in [38]. The Swedish read speech corpus is an open source TTS corpus from the Norwegian Språkbanken [2]. The 11-hour speech corpus consists of 5200 sentences read by a professional speaker. In the current study, we make use of a version of the Swedish synthesizer where both voices have been trained at the same time in a multi-speaker version of Tacotron-2 [35], with a speaker embedding concatenated to the encoder outputs at every token as in [39]. For training a multi-speaker model, an 8 dimensional speaker embedding is appended to a pre-trained single speaker Tacotron-2 model built on spontaneous speech, with the weights of the additional nodes initialized at 0. This setup implies that interpolating between speaker vectors changes speaker identity and speaking style simultaneously, since the read speech and the spontaneous speech corpora were recorded by two different people.

The English read speech samples were taken from the study by Wester and colleagues [33]. In order to make disfluent versions of the synthesized prompts, the authors spliced in spontaneous fillers from the voice actor they used to train the TTS voice. The English and Swedish spontaneous voices described above both contain spontaneous fillers in the training corpus and could thus be generated at the same time as the linguistic content of the prompt. The Swedish read speech voice did not contain fillers in the training corpus, but as the multi-speaker model was trained together with the spontaneous voice, it was possible to produce fillers even when the read speaker's identity vector was applied at inference. In order to assess to what extent this affected the quality of speech with fillers, we also investigated the perceived personality trait of disfluent speech at different interpolation points between the two speaker id vectors. The English spontaneous speech and all Swedish samples in the evaluations below are available online [3].

[1] https://github.com/NVIDIA/tacotron2

[2] https://www.nb.no/sprakbanken/en/resource-catalogue/

[3] http://www.speech.kth.se/tts-demos/ssw2021personality

4. Experiments

In these experiments, our aim is to study how the way a speech synthesizer reads a text influences the perceived personality using the Big-Five model: *Extraversion* (Enthusiasm, Assertiveness); *Neuroticism* (Volatility, Withdrawal); *Conscientiousness* (Industriousness, Orderliness); *Agreeableness* (Compassion, Politeness) and *Openness* (Intellect, Openness). [40]. In order to measure the perceived personality traits, we used the ten Newcastle Personality Assessor (NPA) Questions, as in the original study (Appendix). We also used the same texts as the original study [33], which were designed to elicit different personality traits. They include a person's view of their working environment and a speed dating utterances with negative or positive emotions. They were translated to Swedish (Appendix).

4.1. Experiment 1: perceived personality depending on speaking style and fluency

The first study examined to what extent the perception of personality of synthesized speech depends on whether it is trained on read or spontaneous speech and if the input text contains fillers. We investigated this both in English and in Swedish, where all texts were synthesized in 4 versions: *read fluent* (Eng-Read-Flu, Swe-Read-Flu) , *read disfluent* (Eng-Read-Dis, Swe-Read-Dis), *spontaneous fluent* (Eng-Spon-Flu, Swe-Spon-Flu) and *spontaneous disfluent* (Eng-Spon-Dis, Swed-Spon-Dis). For each language we recruited 60 participants via Prolific. During the test, each synthesis file was presented at the top of a web page, with the 10 personality questions/statements below, where the subjects had to score each on a Likert scale from "Very Unlikely" to "Very Likely". For both languages all sound files were assessed by 30 subjects each.

4.2. Experiment 2: perceived personality depending on the mix of read and spontaneous speaking style

The aim of the second evaluation is to study the extent to which the perception of personality of synthesized speech depends on to which degree the voice speaks with a read or spontaneous speaking style. Using the multi-speaker model, 5 variants of the 13 prompts were generated, where the read/spontaneous speech ratios, set by interpolation between the two speaker identity vectors at inference, were 100/0 90/10, 50/50, 10/90 and 0/100. In the perceptual test, we focused on the personality traits where there was a difference in judgment of speaking style in the Swedish part of Experiment 1: Extraversion, Conscientiousness and Openness. Furthermore, since the ratings of these did not depend on fluency we only used the prompts with inserted fillers A total of 40 participants were recruited via Prolific to take part in a MUSHRA-like side by side assessment of how well the 5 variants agreed with the personality questions/statements.

4.3. Experiment 3: perceived spontaneity depending on speaking style and fluency

The third study, we investigated to what extent the perception of spontaneity depends on the insertion of fillers on the input text, and on whether the voice was trained on read speech or conversational podcast data. In Experiment 3 we used 10/90 and 90/10 speaker ratios, since they where less extreme in speaking rates, and we only included the 10 shortest of the 13 prompts. A total of 40 participants were recruited via Prolific to take part in an A/B test where they could listen to two version of the same prompt that differed either in fluency or speaking style, and select which one they thought sounded more spontaneous.

5. Results

5.1. Results 1: perceived personality depending on speaking style and fluency

Mean scores for the personality judgements in English and Swedish can be seen in Figure 1. A one-way ANOVA and a post-hoc Tukey multiple comparison test identified the following significant differences between the voices. For English, the spontaneous voice was perceived as significantly more extrovert than the read one, both for fluent and disfluent styles ($p < 0.001$). A similar pattern was seen for openness, however less strong when comparing the fluent styles ($p = 0.02$). For the read voice, the fluent style was more open than the disfluent style ($p < 0.001$). For neuroticism, the disfluent read voice was more neurotic than the fluent one ($p < 0.001$) but this relation did carry over to the spontaneous voice. The disfluent read voice was also more neurotic than the disfluent spontaneous voice ($p < 0.001$). For conscientiousness, the fluent read voice scored higher than the disfluent read voice.

For Swedish, the spontaneous voice was rated as more extrovert than the read voice ($p < 0.001$), while the read voice was rated as more open and ($p < 0.001$) and conscientious ($p < 0.001$). There were no significant differences in personality between the fluent and disfluent styles of the Swedish voices.

5.2. Results 2: perceived personality depending on the mix of read and spontaneous speaking style

Mean scores of the personality judgement for the Swedish voices on the continuum from 100%spontaneous to read speech (or 0% spontaneous) can be seen in figure 2 (left). A one-way ANOVA and a post-hoc Tukey multiple comparison test identified the following significant differences: For extraversion, there were significant differences ($p < 0.001$) between all voices except the extremes (100% vs 90% and 10% vs 0%), where more spontaneous was rated more extrovert. Regarding both openness and conscientiousness, the less spontaneous styles (50%, 10% and 0%) was rated significantly higher than the spontaneous ones ($p < 0.001$).

5.3. Results 3: perceived spontaneity depending on speaking style and fluency

Results from the pairwise comparisons of Swedish voices with respect to spontaneity can be seen in Figure 2 (right). The spontaneous voices were judged more spontaneous than the read voices, and the disfluent voices were judged more spontaneous than the fluent. All differences were significant ($p < 0.001$).

6. Discussion

The results of Experiment 1 show that there is a larger effect on the personality rating of adding fillers to the read speech samples that in the spontaneous speech. Adding fillers to read speech makes the voice significantly more neurotic and less open and less conscientious. This is consistent with the original study where adding fillers to read speech also made it less extrovert. For the Swedish voices, the inserted fillers had no significant effect on the personality ratings. The reason might be that the fillers inserted in the English unit selection synthesis were prosodically different than the surrounding speech, and thus more prominent. In the spontaneous English voice and both Swedish voices, fillers where treated as any word in the TTS, which meant that the prosodic realization of both the fillers and the surrounding speech where generated cohesively.

For both Swedish and English the spontaneous voices were rated significantly more extrovert than the read speech voices regardless of fluency. This is consistent with previous findings that extroversion is associated with greater pitch range and faster speaking rate. For the English samples with fillers, the spontaneous ones were rated as significantly more open and less neurotic than the read speech versions. For Swedish, the spontaneous samples were rated less open and less conscientious than the read speech versions. According to previous psychological studies, speakers with great prosodic variability are perceived as "competent" and "knowledgable", thus they should rate high on conscientiousness. At the same time, this trait is also described as "organized", "thorough", and "reliable", which matches speaking style of professional radio speakers, which is a slow and low pitched voice [41]. In our case the read speech voice is recorded with a professional low pitched speaker, which might explain the results.

In Experiment 2 we studied the effect of mixing speaking styles through different interpolations between speaker ids in a multi-speaker model. For openness and conscientiousness the difference between read and spontaneous speech was not very large. For extroversion the difference was quite large and the 50/50 mix is rated in the middle of the ratings for read and spontaneous speech. Overall the speaking rate and pitch range increases with more spontaneous speech in the mix, and this is reflected in the personality ratings. What we could find was that adding 10% spontaneous speech into the read speech voice improved the way it realized the fillers, and by adding 10% read speech into the spontaneous voice made it slightly slower and more articulated. At the same time, these small modifications did not have a significant effect on the personality ratings.

In Experiment 3, we decided to investigate how the 10/90 and 90/10 mixes of read and spontaneous speech voices were rated in terms on perceived spontaneity. Regardless of fluency, the voice with the weight mainly towards conversational speech was almost always rated as more spontaneous than the one with weight towards read speech. Regardless of speaking style, adding fillers makes a voice sound significantly more spontaneous.

7. Conclusions

In this paper we investigated the impact of speaking style and the addition of fillers on perceived personality traits and spontaneity. We confirmed the results of Wester et al. [33], that adding spontaneous fillers into read English speech synthesis makes it significantly more neurotic and less open and less conscientious, but in our listening tests, only slightly less extrovert. For English spontaneous speech synthesis adding fillers only had a significant difference for extraversion and openness. For Swedish, fillers did not change the perceived personality, but is changed the perceived spontaneity. These results are promising because it means that we can insert fillers in a voice in cases where it needs to sound more spontaneous, without changing the portrayed personality. We also found that it is beneficial both for a read speech voice and a spontaneous speech voice to co-train it with a voice with another speaking style, even if they differ in voice quality. It gives the possibility to either slightly adjust the speaking style and handling of fillers, or to create a voice style that exhibits characteristics halfway between read speech and spontaneous speech.

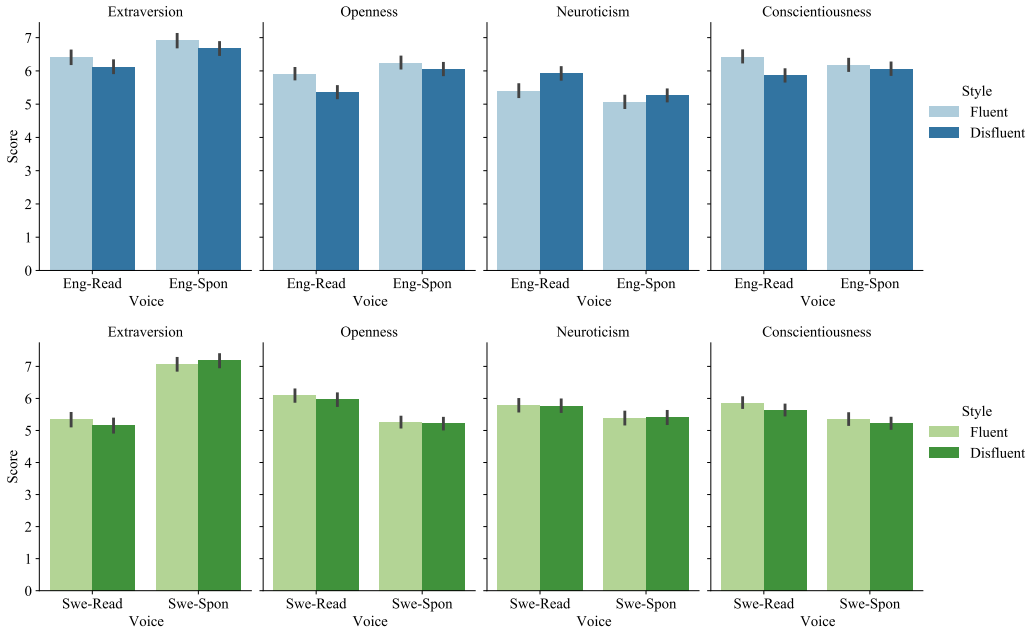


Figure 1: *Results of the personality rating experiments for EN (top) and SW (bottom)*

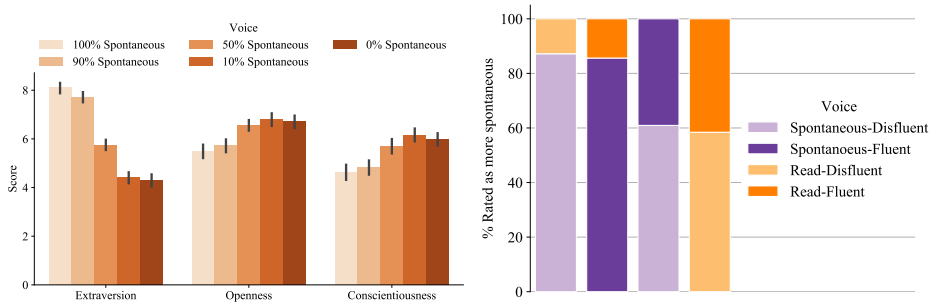


Figure 2: *Extended experiments on the Swedish voices. Left: MUSHRA-like simultaneous scoring of voices on a continuum from read to spontaneous w.r.t. to personality questions. Right: A/B comparison of* read *vs.* spontaneous *voice and* fluent *vs* disfluent *voice w.r.t. the question "Which one sounds more spontaneous?"*

8. Acknowledgements

This research is supported by the Swedish Research Council project Connected (VR-2019-05003), the Riksbankens Jubileumsfond project CAPTivating (P20-0298) and the Digital Futures project Advanced Adaptive Intelligent Systems (AAIS).

9. References

[1] H. H. Clark and J. Fox Tree, "Using uh and um in spontaneous speaking," *Cognition*, vol. 84, no. 1, pp. 73–111, 2002.

[2] H. Maclay and C. E. Osgood, "Hesitation phenomena in spontaneous english speech," *Word*, vol. 15, no. 1, pp. 19–44, 1959.

[3] D. James, "Some aspects of the syntax and semantics of interjections," in *Proc of Chicago Linguistic Society*, 1972.

[4] H. H. Clark and J. E. F. Tree, "Using uh and um in spontaneous speaking," *Cognition*, vol. 84, no. 1, pp. 73–111, 2002.

[5] W. J. Levelt, "Monitoring and self-repair in speech," *Cognition*, vol. 14, no. 1, pp. 41–104, 1983.

[6] A. Gravano and J. Hirschberg, "Turn-taking cues in task-oriented dialogue," *Computer Speech & Language*, vol. 25, no. 3, 2011.

[7] S. H. Fraundorf and D. G. Watson, "The disfluent discourse: Effects of filled pauses on recall," *Journal of memory and language*, vol. 65, no. 2, pp. 161–175, 2011.

[8] M. Corley, L. J. MacGregor, and D. I. Donaldson, "It's the way that you, er, say it: Hesitations in speech affect language comprehension," *Cognition*, vol. 105, no. 3, pp. 658–668, 2007.

[9] S. E. Brennan and M. Williams, "The feeling of another s knowing: Prosody and filled pauses as cues to listeners about the metacognitive states of speakers," *Journal of memory and language*, vol. 34, no. 3, pp. 383–398, 1995.

[10] G. Tottie, "On the use of uh and um in american english," *Functions of Language*, vol. 21, no. 1, pp. 6–29, 2014.

[11] N. Christenfeld, "Does it hurt to say um?" *Journal of Nonverbal Behavior*, vol. 19, no. 3, pp. 171–186, 1995.

[12] J. Holmes, "Functions of you know in women's and men's speech," *Language in society*, pp. 1–21, 1986.

[13] J. Fox Tree, "Folk notions of um and uh, you know, and like," 2007.

[14] D. Schiffrin, *Discourse markers*. Cambridge University Press, 1987, no. 5.

[15] G. Andersen, *Pragmatic markers and sociolinguistic variation: A relevance-theoretic approach to the language of adolescents*. John Benjamins Publishing, 2001, vol. 84.

[16] C. M. Laserna, Y.-T. Seih, and J. W. Pennebaker, "Um... who like says you know: Filler word use as a function of age, gender, and personality," *Journal of Language and Social Psychology*, vol. 33, no. 3, pp. 328–338, 2014.

[17] M. E. Siegel, "Like: The discourse particle and semantics," *Journal of Semantics*, vol. 19, no. 1, pp. 35–71, 2002.

[18] J. Miller, "Like and other discourse markers," *Comparative studies in Australian and New Zealand English*, pp. 317–337, 2009.

[19] W. Apple, L. A. Streeter, and R. M. Krauss, "Effects of pitch and speech rate on personal attributions." *Journal of personality and social psychology*, vol. 37, no. 5, p. 715, 1979.

[20] C. I. Nass and S. Brave, *Wired for speech: How voice activates and advances the human-computer relationship*. MIT press Cambridge, MA, 2005.

[21] J. D. Harnsberger, R. Shrivastav, W. Brown Jr, H. Rothman, and H. Hollien, "Speaking rate and fundamental frequency as speech cues to perceived age," *Journal of voice*, vol. 22, no. 1, 2008.

[22] S. Feldstein and B. Sloan, "Actual and stereotyped speech tempos of extraverts and introverts," *Journal of Personality*, vol. 52, no. 2, pp. 188–204, 1984.

[23] R. L. Street Jr and R. M. Brady, "Speech rate acceptance ranges as a function of evaluative domain, listener speech rate, and communication context," *Communications Monographs*, vol. 49, no. 4, pp. 290–308, 1982.

[24] N. Miller, G. Maruyama, R. J. Beaber, and K. Valone, "Speed of speech and persuasion." *Journal of personality and social psychology*, vol. 34, no. 4, p. 615, 1976.

[25] B. L. Smith, B. L. Brown, W. J. Strong, and A. C. Rencher, "Effects of speech rate on personality perception," *Language and speech*, vol. 18, no. 2, pp. 145–152, 1975.

[26] S. Sundaram and S. Narayanan, "An empirical text transformation method for spontaneous speech synthesizers," in *Proc. Eurospeech*, 2003, pp. 1221–1224.

[27] R. Dall, M. Tomalin, M. Wester, W. Byrne, and S. King, "Investigating automatic & human filled pause insertion for speech synthesis," in *Proceedings of Interspeech*, 2014.

[28] J. Adell, A. Bonafonte, and D. Escudero, "Filled pauses in speech synthesis: towards conversational speech," in *International Conference on Text, Speech and Dialogue*, 2007, pp. 358–365.

[29] S. Andersson, K. Georgila, D. Traum, M. Aylett, and R. A. Clark, "Prediction and realisation of conversational characteristics by utilising spontaneous speech for unit selection," in *Proceedings of Speech Prosody*, 2010.

[30] É. Székely, G. E. Henter, J. Beskow, and J. Gustafson, "How to train your fillers: uh and um in spontaneous speech synthesis," in *The 10th ISCA Speech Synthesis Workshop*, 2019.

[31] C. Nass and K. M. Lee, "Does computer-generated speech manifest personality? an experimental test of similarity-attraction," in *Proceedings of SIGCHI*, 2000, pp. 329–336.

[32] J. Gustafson and K. Sjölander, "Voice creation for conversational fairy-tale characters," in *5th Speech Synthesis workshop*, 2004.

[33] M. Wester, M. Aylett, M. Tomalin, and R. Dall, "Artificial personality and disfluency," in *Sixteenth Annual Conference of the International Speech Communication Association, Interspeech*, 2015.

[34] M. P. Aylett, A. Vinciarelli, and M. Wester, "Speech synthesis for the generation of artificial personality," *IEEE transactions on affective computing*, vol. 11, no. 2, pp. 361–372, 2017.

[35] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerry-Ryan, R. A. Saurous, Y. Agiomyrgiannakis, and Y. Wu, "Natural TTS synthesis by conditioning WaveNet on mel spectrogram predictions," in *Proc. ICASSP*, 2018, pp. 4779–4783.

[36] R. Prenger, R. Valle, and B. Catanzaro, "Waveglow: A flow-based generative network for speech synthesis," in *Proceedings of ICASSP*, 2019, pp. 3617–3621.

[37] Y. Ferstl and R. McDonnell, "Investigating the use of recurrent motion modelling for speech gesture generation," in *Proc. IVA*, 2018, pp. 93–98. [Online]. Available: https://trinityspeechgesture.scss.tcd.ie

[38] É. Székely, G. E. Henter, and J. Gustafson, "Casting to corpus: Segmenting and selecting spontaneous dialogue for TTS with a CNN-LSTM speaker-dependent breath detector," in *Proc. ICASSP*, 2019, pp. 6925–6929.

[39] R. Valle, J. Li, R. Prenger, and B. Catanzaro, "Mellotron: Multispeaker expressive voice synthesis by conditioning on rhythm, pitch and global style tokens," in *Proceedings of ICASSP*, 2020.

[40] L. R. Goldberg *et al.*, "A broad-bandwidth, public domain, personality inventory measuring the lower-level facets of several five-factor models," *Personality psychology in Europe*, vol. 7, no. 1, pp. 7–28, 1999.

[41] G. Mohammadi and A. Vinciarelli, "Automatic personality perception: Prediction of trait attribution based on prosodic features," *IEEE Transactions on Affective Computing*, vol. 3, no. 3, 2012.

10. Appendix

Newcastle Personality Assessor Questions/statements: Start a conversation with a stranger? (Ext); Make sure others are comfortable and happy? (Agr); Use difficult words? (Ope); Prepare for things in advance? (Con); Feel blue or depressed? (Neu); Plan parties or social events? (Ext) Insult people? (-Agr) Think about philosophical questions? (Ope); Let things get into a mess? (-Con); Feel stressed or worried? (Neu)

ID	Sentences
A1	I like to bring order to everything I do (YOU KNOW). I think the details and facts are often missed by others and (UM) I like to work based on concrete result. If faced by a problem I like to look at it logically and (LIKE) make a decision based on the specific problems at hand.
A2	(I MEAN) I'm good at encouraging others to work with each other and cooperate effectively. I think that if you look after and help colleagues you (UH) get the best out of them.(I MEAN)If you do good work then the people around you will also become more motivated.
A3	I'm great at getting people to work with each other and (I MEAN) sorting out misunderstandings and conflict. If you concentrate on the common ideas and values you all share (YOU KNOW) you can find real insight and discover new possibilities.
A4	I like to plan provide direction and (UM) make sure everyone knows what their responsibilities are. I think its very important to be a good example to others (LIKE) to be committed and to work hard on doing things the right way to achieve your goals.
A5	I'm good at encouraging others to contribute (UM) effectively. I think its important to enjoy your work and to be enthusiastic about what you do(YOU KNOW)
A6	I'm great at helping others plan and (LIKE) cooperate to get things done. Its important to work out what can be done and (UH) the best way to do it. (I MEAN) I like to work with others and help everyone come together behind a project.
A7	I'm good at developing new strategies and approaches to a problem and I think (UM) being committed to what you do is very important. I love innovation and overcoming challenges (YOU KNOW)
N1	I'm from West London ; which is a part of town I really dislike (YOU KNOW). it was a real pain it the arse to get here (I CAN TELL YOU) ; I used to like film until Hollywood (LIKE) ruined them all.
N2	What a mess this place is (I MEAN) I'm sure the organiser has got it in for me.I've always had problems with people either because they are stupid or (UH) jealous of me.
N3	(UM) you don't seem to have made much effort though given the losers here (LIKE) I'm not surprised you'd probably be happier (UM) um watching TV at home.
P1	I'm from a lovely little suburb with (UM) lots of trees and parks. The train is very quick and it was no (LIKE) trouble to get here. I love going to the beach and (LIKE) spending time with my friends.
P2	They've done a brilliant job at redecorating this bar (YOU KNOW) The people running it have been (UM) really nice to me. I always get on with people (I MEAN) we have so much to share with each other.
P3	(I MEAN) I must say you are looking very nice tonight Everyone is very nicely dressed and (LIKE) seem so successful (UM) I expect you are looking forward to coming again.

Table 1: *The prompts from Wester et al 2015 [33]. About Myself (A) Speed Dating Negative (N) and Speed Dating Positive (P)*

ID	Sentences
A1	Jag gillar och ha ordning på allt jag gör(SKULLE JAG SÄGA) Jag tycker att detaljerna och fakta ofta saknas i det andra gör och (EH) Jag gillar att arbeta baserat på konkreta resultat. Om jag ställs inför ett problem vill jag angripa det logisk och (TYP) fatta ett beslut baserat på det specifika problemet .
A2	(JAG ANSER ATT) jag är bra på att uppmuntra andra att arbeta med varandra och samarbeta effektivt. Jag tycker att om du tar hand om och hjälper kollegor så får du (EH) ut det bästa av dem. (DET ÄR JU SÅ ATT) om du gör bra arbete då kommer folk omkring dig också att bli mer motiverade.
A3	Jag är bra på att få människor att arbeta med varandra (LIKSOM) och reda ut missförstånd och konflikter. Om du koncentrerar dig på de gemensamma idéerna och värderingarna (ALLTSÅ) så kan du komma till verklig insikt och upptäcka nya möjligheter.
A4	Jag gillar att planera ge vägledning och (EH) se till att alla vet vad deras ansvar är. Jag tycker att det är mycket viktigt att man är ett bra exempel för andra (LIKSOM) att man är engagerad och arbetar hårt för att göra saker på rätt sätt för att uppnå sina mål.
A5	Jag är bra på att uppmuntra andra att bidra (EH) effektivt. Jag tycker att det är viktigt att man njuter av sitt arbete och att man är entusiastisk över det man gör (SÅ ATT SÄGA)
A6	Jag är bra på att hjälpa andra att planera och (LIKSOM) samarbeta för att få saker gjorda. Det är viktigt att ta reda på vad som kan göras och (EH) det bästa sättet att göra det. (JAG MENAR) jag gillar att arbeta med andra och hjälpa alla att känna sig delaktiga i ett projekt.
A7	Jag är bra på att utveckla nya strategier och tillvägagångssätt för att lösa problem och jag tycker att det är mycket viktigt (EHM) att man är engagerad i det man gör. Jag älskar innovation och att övervinna utmaningar (SÅ ATT SÄGA)
N1	Jag är från västra London som är en del av staden som jag verkligen ogillar (SKULLE JAG SÄGA). Det var ett jäkla sjå att komma hit (SÅ ATT SÄGA). Jag brukade gilla film tills Hollywood förstörde dem alla.
N2	Vilken röra det är på det här stället (JAG MENAR) jag är säker på att arrangören inte gillar mig Jag har alltid haft problem med folk antingen för att de är dumma eller (EH) avundsjuka på mig.
N3	(EH) du verkar inte ha gjort stora ansträngningar men med tanke på förlorarna här (LIKSOM) är jag inte förvånad du skulle förmodligen vara lyckligare (EHM) om du var hemma och kollade på tv
P1	Jag kommer från en härlig liten förort med (EH) massor av träd och parker. Tågresan var mycket kort och det var (TYP) inga problem att ta sig hit. Jag älskar att åka till stranden och (LIKSOM) spendera tid med mina vänner.
P2	Dom har gjort ett fantastiskt jobb med att renovera den här baren (ALLTSÅ), Dom som driver det har varit riktigt (EH) trevliga mot mig. Jag kommer alltid väl överrens med folk (JAG MENAR) vi har så mycket att dela med oss av till varandra.
P3	(JAG MENAR) jag måste säga att du ser väldigt bra ut ikväll. Alla är väldigt snyggt klädda och verkar (LIKSOM) så framgångsrika. (EH) Jag förväntar mig att du ser fram emot att komma tillbaka.

Table 2: *The Swedish translation of the texts. About Myself (A) Speed Dating Negative (N) and Speed Dating Positive (P)*

11th ISCA Speech Synthesis Workshop (SSW 11)
26-28 August 2021, Budapest, Hungary

Adaptation of Tacotron2-based Text-To-Speech for Articulatory-to-Acoustic Mapping using Ultrasound Tongue Imaging

Csaba Zainkó[1], László Tóth[2], Amin Honarmandi Shandiz[2], Gábor Gosztolya[3], Alexandra Markó[4,5], Géza Németh[1], Tamás Gábor Csapó[1,5]

[1]Department of Telecommunications and Media Informatics,
Budapest University of Technology and Economics, Budapest, Hungary
[2]Institute of Informatics, University of Szeged, Hungary
[3]MTA-SZTE Research Group on Artificial Intelligence, Szeged, Hungary
[4]Department of Applied Linguistics and Phonetics, Eötvös Loránd University, Budapest, Hungary
[5]MTA-ELTE Lendület Lingual Articulation Research Group, Budapest, Hungary
{zainko, nemeth, csapot}@tmit.bme.hu, {tothl, shandiz, ggabor}@inf.u-szeged.hu,
marko.alexandra@btk.elte.hu

Abstract

For articulatory-to-acoustic mapping, typically only limited parallel training data is available, making it impossible to apply fully end-to-end solutions like Tacotron2. In this paper, we experimented with transfer learning and adaptation of a Tacotron2 text-to-speech model to improve the final synthesis quality of ultrasound-based articulatory-to-acoustic mapping with a limited database. We use a multi-speaker pre-trained Tacotron2 TTS model and a pre-trained WaveGlow neural vocoder. The articulatory-to-acoustic conversion contains three steps: 1) from a sequence of ultrasound tongue image recordings, a 3D convolutional neural network predicts the inputs of the pre-trained Tacotron2 model, 2) the Tacotron2 model converts this intermediate representation to an 80-dimensional mel-spectrogram, and 3) the WaveGlow model is applied for final inference. This generated speech contains the timing of the original articulatory data from the ultrasound recording, but the F0 contour and the spectral information is predicted by the Tacotron2 model. The F0 values are independent of the original ultrasound images, but represent the target speaker, as they are inferred from the pre-trained Tacotron2 model. In our experiments, we demonstrated that the synthesized speech quality is more natural with the proposed solutions than with our earlier model.

Index Terms: articulation-to-speech, ultrasound, DNN-TTS

1. Introduction

Articulatory-to-acoustic mapping (AAM) methods aim to synthesize the speech signal directly from articulatory input, as opposed to text-to-speech, when speech is synthesized from the textual input. AAM applies the theory that articulatory movements are directly linked with the acoustic speech signal in the speech production process. A recent potential application of this mapping is a "Silent Speech Interface" (SSI [1, 2, 3]), which has the main idea of recording the soundless articulatory movement, and automatically generating speech from the movement information, while the subject does not produce any sound. Such an SSI system can be highly useful for the speaking impaired (e.g. after laryngectomy or elderly people), and for scenarios where regular speech is not feasible, but the information should be transmitted from the speaker (e.g. extremely noisy environments or military applications).

For the articulatory-to-acoustic mapping, the typical input can be electromagnetic articulography (EMA) [4, 5], ultrasound tongue imaging (UTI) [6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19], permanent magnetic articulography (PMA) [20, 21], surface electromyography (sEMG) [22, 23], Non-Audible Murmur (NAM) [24], electro-optical stomatography [25], impulse radio ultra-wide band (IR-UWB) [26], radar [27] or video of the lip movements [7, 28, 29]. From another aspect, there are two distinct ways of SSI solutions, namely 'direct synthesis' and 'recognition-and-synthesis' [2]. In the first case, the speech signal is generated without an intermediate step, directly from the articulatory data [4, 5, 6, 8, 9, 11, 12, 14, 15, 16, 20, 22, 23, 24, 28]. In the second case, silent speech recognition (SSR) is applied on the biosignal which extracts the content spoken by the person (i.e. the result of this step is text); this step is then followed by text-to-speech (TTS) synthesis [7, 10, 13, 25, 29, 30]. In the SSR+TTS approach, any information related to speech prosody is lost, whereas it may be kept with direct synthesis. Also, the smaller delay by the direct synthesis approach might enable conversational use.

For the direct conversion, typically, vocoders are used, which synthesize speech from the spectral parameters predicted by the DNNs from the articulatory input. One of the spectral representations that was found to be useful earlier for statistical parametric speech synthesis is Mel-Generalized Cepstrum in Line Spectral Pair form (MGC-LSP) [31, 32]. Since the introduction of WaveNet in 2016 [33], neural vocoders can generate highly natural raw samples of speech, conditioned on mel-spectrogram or other input. One of the most recent types of neural vocoders, WaveGlow [34] is a flow-based network capable of generating high-quality speech from mel-spectrograms. The advantage of the WaveGlow model is that it is relatively simple, yet the synthesis can be done faster than real-time. In [17], we integrated the WaveGlow neural vocoder into ultrasound-based articulatory-to-acoustic conversion.

In the latest years, most TTS solutions apply end-to-end methods, by operating directly on character or phoneme input sequences and producing raw speech signal outputs. One of the most widely used solutions is Tacotron2 [35], which applies a recurrent sequence-to-sequence feature prediction network that maps character embeddings to mel-scale spectrograms, followed by a neural vocoder. The encoder-decoder network, using the attention mechanism, encodes a specific attribute of speech and maps sequences of differing length. In [35], the input char-

10.21437/SSW.2021-10

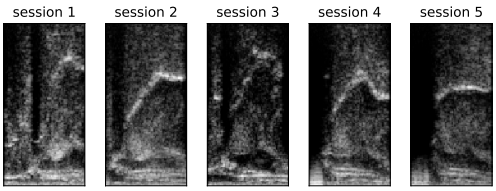


Figure 1: *Sample ultrasound images from the five sessions.*

acters are represented with a learned 512-dimensional embedding, which ensures that traditional text processing is not necessary on the input.

In the field of AAM, according to our knowledge, only a few studies have used fully end-to-end / sequence-to-sequence solutions [36, 37]. Zhang and his colleagues introduced TaLNet, which is based on an encoder-decoder architecture, using the attention mechanism. Both ultrasound and lip are used as the input of AAM, from English speakers of the UltraSuite-TaL database [38]. First, a Tacotron2 model is trained with a large amount of speech data, and after that, transfer learning is applied with the articulatory input. The presented approach was found to be significantly better than earlier baselines. In the study, they also checked the contribution of each articulatory input, and found that the weakest results could be achieved with the lip-only system, followed by ultrasound-only. The combination of ultrasound and lip (TaLNet) was found to be the best, suggesting that these two modalities complement each other well. In another study, by Mira and his colleagues, end-to-end video-to-speech synthesis was proposed, using GANs [37]. The video of the face is translated directly to speech, without an intermediate representation, applying an encoder-decoder architecture. They experimented on various databases and show that the choice of adversarial loss is a key for realistic results.

In this paper, we experiment with transfer learning and adaptation of a Tacotron2 text-to-speech model to improve the final synthesis quality of ultrasound-based articulatory-to-acoustic mapping with a limited database.

2. Methods

2.1. Data

For Tacotron2 and WaveGlow training, we chose 5 male and 6 female Hungarian speakers (altogether 23k sentences, roughly 22 hours) from the PPSD database [39]. This data served as the acoustic-only training material required for the encoder-decoder architecture and the neural vocoder.

For the articulatory data, we used the Hungarian parallel ultrasound and speech dataset that we recorded for earlier studies [16, 17, 40]. We selected a female speaker (speaker048), who was recorded in five sessions (once 209 sentences, and four times 59 sentences). The tongue movement was recorded in midsagittal orientation using the "Micro" ultrasound system of Articulate Instruments Ltd. at 81.67 fps. The speech signal was recorded with a Beyerdynamic TG H56c tan omnidirectional condenser microphone. The ultrasound data and the audio signals were synchronized using the tools provided by Articulate Instruments Ltd. In our experiments, the raw scanline data of the ultrasound was used as input of the networks, after being resized to 64×128 pixels using bicubic interpolation (see samples in Fig. 1), as we found earlier that this reduction does not cause significant information loss [41].

For the Tacotron2 speaker adaptation, speaker048's data was used (train: 318 sentences, and validation: 40 sentences).

Conv3D	**Conv3D(Transfer_Learninng)**
Conv3D(30,(5,13,13),Strides=(5,2,2))	Conv3D(30,(5,13,13),Strides=(5,2,2))
Dropout(0.2)	Dropout(0.2)
Conv3D(60,(1,13,13),Strides=(1,2,2))	Conv3D(60,(1,13,13),Strides=(1,2,2))
Dropout(0.2)	Dropout(0.2)
MaxPooling3D(poolsize=(1,2,2))	MaxPooling3D(poolsize=(1,2,2))
Conv3D(70,(1,13,13),Strides=(1,2,2))	Conv3D(70,(1,13,13),Strides=(1,2,2))
Dropout(0.2)	Dropout(0.2)
Conv3D(58,(1,13,13),Strides=(1,1,1))	Conv3D(58,(1,13,13),Strides=(1,1,1))
Dropout(0.2)	Dropout(0.2)
MaxPooling3D(poolsize=(1,2,2))	MaxPooling3D(poolsize=(1,2,2))
Flatten()	Flatten()
Dense(1000)	Dense(1000)
Dropout(0.2)	Dropout(0.2)
Dense(80,activation='linear')	**Dense(93, activation='softmax')**

Figure 2: *The layers of the 3D CNNs in the Keras implementation, along with their most important parameters. Left: baseline 3D CNN for melspectrogram prediction, right: proposed 3D CNN for symbol prediction.*

2.2. Ultrasound-to-Melspectrogram using 3D-CNN (baseline)

When we are dealing with image processing as input data, then convolutional neural networks are one of the most popular and effective methods which can extract complex features from data by adding deep layers [42]. In Silent Speech Interface, when we have ultrasound data as input, our input is not only just images but sequences of images which could be considered as a video. Standard CNN considers 2D images to extract features by convolving 2D filters over images. Therefore, to model temporal information, a third dimension has to be considered [43, 44]. Recurrent Neural Networks such as Long Short Term Memory (LSTM) are good examples of combining features extracted from both temporal and spatial parts of data [44]. Using LSTM networks have some drawbacks such as training difficulties, while some variants of these networks were proposed to mitigate this problem, such as quasi-recurrent neural networks [45].

Here we use another variation by adding a third dimension as (2+1)D CNN which shows good performance in video action recognition task [46]. It shows good results when used with ultrasound images and it could be considered as a substitute of CNN+LSTM [18]. In the baseline system of the current study, we apply the same 3D CNN which was used in [18] for predicting 80-dimensional melspectrogram features from ultrasound tongue image input.

This network processed 5 frames of video that were 6 frames apart (6 is the stride parameter of the convolution along the time axis) [18]. Following the concept of (2+1)D convolution, the five frames were first processed only spatially, and then got combined along the time axis just below the uppermost dense layer. Fig. 2 left shows the actual network configuration. The training was performed using the SGD optimizer with 0.06 starting learning rate. It was reduced when a validation MSE has stopped improving by factor 0.5. The batch size was 128. The training objective function was the mean squared error (MSE).

2.3. Ultrasound-to-Symbol using 3D-CNN

In the proposed system, we use the same structure of the 3D CNN as in the baseline system. The difference is in the tar-

get of the network: we predict symbols of Tacotron2 internal representation, having 93 dimensions. At first, we trained with the same methods as the baseline model, but the model was not applicable. We fine-tuned the optimizer, batch size, and other hyperparameters but the model still did not train. Sometimes the accuracy was zero or it learned only the silent symbol and predicted it everywhere. Finally, transfer learning was successful. We reused the baseline 3D-CNN model's weights at the convolutional layers. All convolutional layers were frozen and only the last two FC layers (with 1000 and 93 neurons) were trained. The weights of these two layers were initialized randomly. Here, cross-entropy is used as the loss function. Because the classes of symbols were not balanced, we used a specific loss function: the loss was weighted with the occurrence of the symbols. We used Adam optimizer and accuracy as a metric. The other parameters of the CNN are the same as the baseline, see Fig. 2 right.

2.3.1. Accuracy and the confusion matrix

The Ultrasound-to-Symbol 3D-CNN model reached 0.68 validation accuracy after 20 epochs (train acc.: 0.83). Early stopping was used with a patience parameter of 7. To improve our Tacotron2 model, the confusion matrix was used to generate augmented training data (see later in Sec. 2.4.3). Fig. 3 shows a simplified version of the confusion matrix (for visualization purposes only – the full matrix involves all 93 symbols: for this figure, we removed the symbols which were not used in the current models and pooled together the short and long versions of the symbols). The values are normalized by rows (target symbols) and converted to percentage values. The first row (on the top) is the most accurate symbol, and the last row (on the bottom) is the least accurate symbol. We expected that the errors are related to articulation, but in Fig. 3 it seems mainly noise-like. The symbols with lower accuracies were some vowels and nasals (e,a,ee,n,m in the figure, /ɛ,ɔ,eː,n,m/ in IPA). The symbols with higher accuracies were some less frequent consonants (z,ty,cs,zs in the figure, /ʒ,tʃ,c,z/ in IPA).

2.4. Symbol-to-melspectrogram using Tacotron2

We used a multi-speaker Tacotron2 model [35] based on the NVIDIA implementation (`https://github.com/NVIDIA/tacotron2`). The speakers' IDs are coded as a one-hot vector and added to the inputs of the LSTM cells both in the encoder and decoder. The model was trained by all 11 speakers of the PPSD database [39] at the same time. The order of all speakers' sentences was randomized. The input of the Tacotron2 is a sequence of symbols. Because Hungarian is an almost phonetic language, we used a mixed collection of letters and phonemes. The symbols of the input sequence follow the phonemes of the sentences, but we did not use allophones or other detailed discrimination. Only the long–short property is used to encode durational differences. The phonemes are represented with their approximate letter: the lowercase letters show the short phonemes, the capital letters indicate the long phonemes.

This multi-speaker model was trained during 156k iterations on a single NVIDIA Titan Xp. The sample rate of the sound was 22 050 Hz, the window size was 1024 and the hop length was 256. We used 80 mel channels between 0 Hz and 8000 Hz to keep compatibility with the WaveGlow model. The encoder's symbols embedding and embedding dimension was also 512. The decoder's RNN dimensions were 1024.

Our goal was to use our pre-trained Tacotron2 model (originally developed for TTS) without modification, therefore we made only some fine-tuning for AAM purposes. The ultrasound image sequence does not contain F0-related information, but it contains the timing of speech. Basically, the Tacotron2 does not handle timing information of a sentence, it can generate that via an attention mechanism. Fig. 4 top shows an example for the connection between the steps of the encoder and decoder with this initial Tacotron2 system. This sentence encoder contains 16 symbols plus two padding symbols at the borders of the sentence. The model generated 134 decoder frames. In this model, one frame is about 11.6ms, so this sentence was about 1.6s long. Clearly, the timings are not modeled well here.

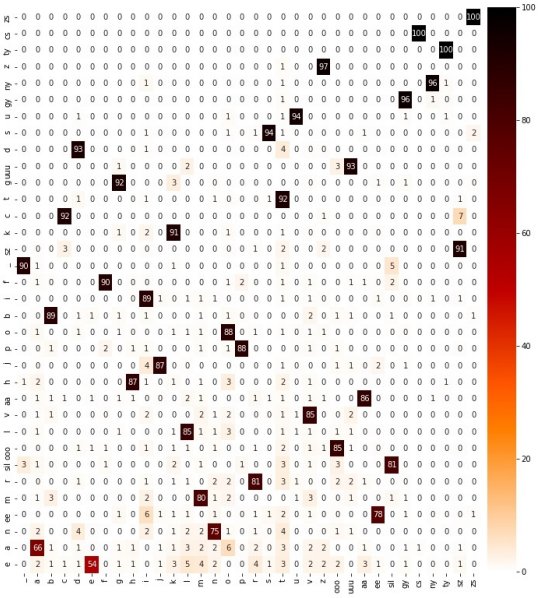


Figure 3: *Simplified confusion matrix of the proposed Ultrasound-to-Symbol 3D-CNN. The values are normalized and showed in percentages. Rows: target, columns: predicted.*

2.4.1. Time-synchronous Tacotron2 system

In order to use the proper timing of the input sequence, we generated a new training set from the original 11 speakers' dataset. The input symbols were repeated accordingly to the real duration of a phone. The repeating number was calculated from the ultrasound frame rate (81.67 fps). For example, at a 98ms long phone, the symbol was repeated 8 times. The attention mechanism adapted to the synchronized input during the fine-tuning. It required 7.5k iterations.

2.4.2. Proposed system #1

The speaker in the ultrasound dataset (speaker048) is independent of the 11 speakers of the training set of Tacotron2. The next step was fine-tuning to the new speaker. We chose a female speaker from the 11 others, and at the tuning, her speakerID one-hot vector was used. At this step, 84 iterations resulted in the smallest validation error. In the first proposed system,

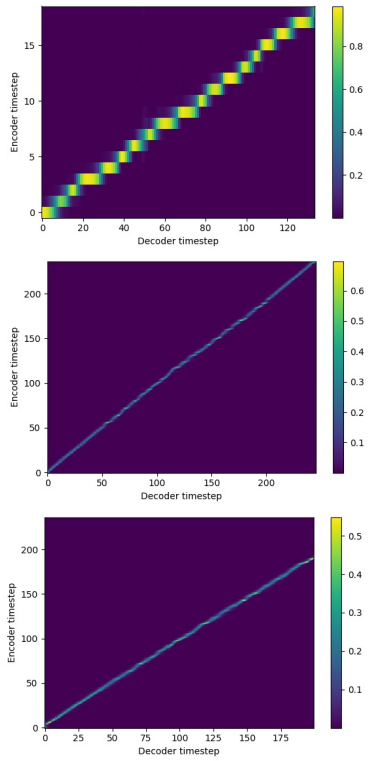


Figure 4: *Examples for the connection between the steps of the encoder and decoder. Top: Tacotron2 without timing information. Middle: Tacotron2 with timing information (Proposed #1). Bottom: Tacotron2 with timing information and with data augmentation (Proposed #2).*

this model was used. Fig. 4 middle shows the proper timing of the generated speech. The input of that sentence contains 237 symbols, and the system generated 246 output frames. The difference comes from the uncertainty of the end decision of the decoder. The figure also shows the Tacotron2 can tolerate some symbol errors, i.e. the line is not perfectly straight; there are some small steps, where the decoder ignores some input symbols.

2.4.3. Proposed system #2

Our experience was that Tacotron2 can tolerate some mistakes in the prediction of the 3D-CNN model (Sec. 2.3), but these mistakes cause audible distortion during the final synthesis. The distribution of the wrong predictions can be characterized by the confusion matrix (Sec. 2.3.1) of the 3D-CNN network. It is not accurate because it does not contain the position information of the mistakes, but it is suitable to generate similar training data for fine-tuning the Tacotron2 model. With the distribution of the symbol's error, we modified the 11 speakers training set. The symbol changing was based on the distribution but it was randomized. For every sentence, 20 different versions were generated. The output mel-spectrograms were not changed. 4.3k iterations provided the lowest validation error. Fig. 4 bottom shows the tuned model's connection between the encoder and decoder. There are two differences compared to the middle subfigure. The number of the encoder steps remained the same, but there are fewer decoder steps. The decoder learned to ignore the different types of silence symbols (pad, sil, start_sil, end_sil) which were mixed in the predicted symbol sequence. The other difference is that the line is smoother. It shows that a decoder step connects more encoder steps and the model can combine the information of good and bad symbols.

After that we also repeated the tuning to the speaker from the ultrasound dataset. Here we also generate modified training data with the phoneme errors. The procedure was the same as at the multi-speaker case. At this second step, 182 iterations were required. We used this model in the second proposed system.

2.5. Melspectrogram-to-speech with a neural vocoder

Similarly to the original WaveGlow paper [34], 80 bins were used for mel-spectrogram using librosa mel-filter defaults (i.e. each bin is normalized by the filter length and the scale is the same as in HTK, Hidden Markov Model Toolkit). FFT size and window size were both 1024 samples. For hop size, we use the base 256 samples. This 80-dimensional mel-spectrogram served as the training target of the Tacotron2 network. A WaveGlow model was trained with the Hungarian data (WaveGlow-HU). This latter training was done on a server with eight V100 GPUs, altogether for 635k iterations. In the synthesis phase, an interpolation in time was not necessary, different from [17]. The ultrasound frame rate was 270 samples, but the differences were compensated by the Tacotron2 model, the output frame rate of the model was 256 samples which is the same as the WaveGlow's hop size. Finally, the synthesized speech is the result of the inference with the trained WaveGlow-HU model conditioned on the mel-spectrogram input [34].

3. Experiments and Results

After training the above models, we synthesized sentences from the test part of the ultrasound dataset. These sentences have not been used during the training process, neither in the Ultrasound-to-Symbol model, nor in the Tacotron2 training and tuning process. The domain of the texts is also independent of the training and validation dataset: it contains the Hungarian version of 'The North Wind and the Sun'.

3.1. Subjective listening test

In order to determine which proposed version is closer to natural speech, we conducted an online MUSHRA-like test [47]. Our aim was to compare the natural sentences with the synthesized sentences of the baseline, the proposed approaches and a lower anchor system (the latter having constant F0 and 2D CNN predicted MGC-LSP, from [17]). In the test, the listeners had to rate the naturalness of each stimulus in a randomized order relative to the reference (which was the natural sentence), from 0 (very unnatural) to 100 (very natural). We chose nine sentences from the test set of the target speaker. The variants appeared in randomized order (different for each listener). The samples can be found at `http://smartlab.tmit.bme.hu/ssw11_tacotron2`.

Each sentence was rated by 23 native Hungarian speakers (11 females, 12 males; 14–47 years old), in a silent environment. On average, the test took 10 minutes to complete. Fig. 5 shows the average naturalness scores for the tested approaches.

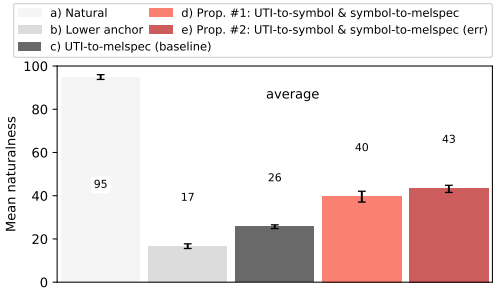


Figure 5: *Results of the subjective evaluation with respect to naturalness. The error bars show the 95% confidence intervals.*

The lower anchor received the weakest scores, followed by the baseline, and the proposed approaches. To check the statistical significances, we conducted Mann-Whitney-Wilcoxon ranksum tests with a 95% confidence level. Based on this, both proposed variants were evaluated as significantly more natural than the baseline. The listeners noted the difference between the two proposed versions: proposed#1, the one with standard training (Sec. 2.4.2) was rated as 40%, while proposed #2, the one with additional error training (Sec. 2.4.3) was rated as 43% – but this difference is not statistically significant.

As a summary of the listening test, we can conclude that splitting the ultrasound-to-speech prediction task into three parts increased the naturalness, mostly because of the Tacotron2 component which could be trained with a large amount of speech data, and transfer learning / adaptation was possible to the target speaker.

4. Discussion

In Sec. 1, we noted that currently only a few sequence-to-sequence / fully end-to-end solutions are available for articulatory-to-acoustic mapping [36, 37]. Our proposed solution has the following similarities and differences. Mira and his colleagues use the video of the face as input [37], Zhang and his colleagues use both ultrasound and lip video input [36], whereas in our study we use ultrasound tongue image input. As the three studies apply different databases, the results are not directly comparable. In [37], GANs are used with specific adversarial loss, whereas we apply 3D CNN to model the spatial and temporal dependencies of the articulatory and acoustic data. Similarly to [36], we apply Tacotron2 as the encoder-decoder network, but we extend the basic training with additional data augmentation, which includes the wrong predictions from the confusion matrix of the UTI-to-symbol prediction network. By using the symbols as intermediate representation, our solution is closer to the 'recognition-and-synthesis' type of SSIs.

5. Conclusions

In this paper, we experimented with transfer learning and adaptation of a Tacotron2 text-to-speech model to improve the final synthesis quality of ultrasound-based articulatory-to-acoustic mapping with a limited database (roughly 200 sentences). We used a Hungarian multi-speaker pre-trained Tacotron2 TTS model and a pre-trained WaveGlow neural vocoder (both trained on 11 speakers's data, altogether 23k sentences, roughly 22 hours of speech). The proposed articulatory-to-acoustic conversion framework is a fully end-to-end solution, including an encoder-decoder architecture and attention mechanism, and contains three steps: 1) from a sequence of ultrasound tongue image recordings, a 3D convolution neural network predicts the 93-dimensional embedding inputs of the pre-trained Tacotron2 model, 2) the Tacotron2 model converts this intermediate representation to a 80-dimensional mel-spectrogram, and 3) the WaveGlow model is applied for final inference. We demonstrated that the synthesized speech quality is significantly more natural with the proposed solutions than with our earlier model.

The code is accessible at `https://github.com/BME-SmartLab/UTI-to-STFT-Tacotron2`.

6. Acknowledgements

The research was partly supported by the European Union's Horizon 2020 research and innovation programme under grant agreement No. 825619 (AI4EU), by the National Research Development and Innovation Office of Hungary (FK 124584 and PD 127915 grants; APH-ALARM / 2019-2.1.2-NEMZ-2020-00012 project) and through the Artificial Intelligence National Laboratory Programme. The Titan X GPU used was donated by NVIDIA Corporation. We would like to thank the subjects for participating in the listening test.

7. References

[1] B. Denby, T. Schultz, K. Honda, T. Hueber, J. M. Gilbert, and J. S. Brumberg, "Silent speech interfaces," *Speech Communication*, vol. 52, no. 4, pp. 270–287, 2010.

[2] T. Schultz, M. Wand, T. Hueber, D. J. Krusienski, C. Herff, and J. S. Brumberg, "Biosignal-Based Spoken Communication: A Survey," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 12, pp. 2257–2271, dec 2017.

[3] J. A. Gonzalez-Lopez, A. Gomez-Alanis, J. M. Martin Donas, J. L. Perez-Cordoba, and A. M. Gomez, "Silent Speech Interfaces for Speech Restoration: A Review," *IEEE Access*, vol. 8, pp. 177 995–178 021, sep 2020.

[4] B. Cao, M. Kim, J. R. Wang, J. Van Santen, T. Mau, and J. Wang, "Articulation-to-Speech Synthesis Using Articulatory Flesh Point Sensors' Orientation Information," in *Proc. Interspeech*, Hyderabad, India, 2018, pp. 3152–3156.

[5] F. Taguchi and T. Kaburagi, "Articulatory-to-speech conversion using bi-directional long short-term memory," in *Proc. Interspeech*, Hyderabad, India, 2018, pp. 2499–2503.

[6] B. Denby and M. Stone, "Speech synthesis from real time ultrasound images of the tongue," in *Proc. ICASSP*, Montreal, Quebec, Canada, 2004, pp. 685–688.

[7] T. Hueber, E.-L. Benaroya, G. Chollet, G. Dreyfus, and M. Stone, "Development of a silent speech interface driven by ultrasound and optical images of the tongue and lips," *Speech Communication*, vol. 52, no. 4, pp. 288–300, 2010.

[8] T. Hueber, E.-l. Benaroya, B. Denby, and G. Chollet, "Statistical Mapping Between Articulatory and Acoustic Data for an Ultrasound-Based Silent Speech Interface," in *Proc. Interspeech*, Florence, Italy, 2011, pp. 593–596.

[9] A. Jaumard-Hakoun, K. Xu, C. Leboullenger, P. Roussel-Ragot, and B. Denby, "An Articulatory-Based Singing Voice Synthesis Using Tongue and Lips Imaging," in *Proc. Interspeech*, San Francisco, CA, USA, 2016, pp. 1467–1471.

[10] E. Tatulli and T. Hueber, "Feature extraction using multimodal convolutional neural networks for visual speech recognition," in *Proc. ICASSP*, New Orleans, LA, USA, 2017, pp. 2971–2975.

[11] T. G. Csapó, T. Grósz, G. Gosztolya, L. Tóth, and A. Markó, "DNN-Based Ultrasound-to-Speech Conversion for a Silent Speech Interface," in *Proc. Interspeech*, Stockholm, Sweden, 2017, pp. 3672–3676.

[12] T. Grósz, G. Gosztolya, L. Tóth, T. G. Csapó, and A. Markó, "F0 Estimation for DNN-Based Ultrasound Silent Speech Interfaces," in *Proc. ICASSP*, Calgary, Canada, 2018, pp. 291–295.

[13] L. Tóth, G. Gosztolya, T. Grósz, A. Markó, and T. G. Csapó, "Multi-Task Learning of Phonetic Labels and Speech Synthesis Parameters for Ultrasound-Based Silent Speech Interfaces," in *Proc. Interspeech*, Hyderabad, India, 2018, pp. 3172–3176.

[14] E. Moliner and T. G. Csapó, "Ultrasound-based silent speech interface using convolutional and recurrent neural networks," *Acta Acustica united with Acustica*, vol. 105, no. 4, pp. 587–590, 2019.

[15] G. Gosztolya, Á. Pintér, L. Tóth, T. Grósz, A. Markó, and T. G. Csapó, "Autoencoder-Based Articulatory-to-Acoustic Mapping for Ultrasound Silent Speech Interfaces," in *International Joint Conference on Neural Networks*, 2019.

[16] T. G. Csapó, M. S. Al-Radhi, G. Németh, G. Gosztolya, T. Grósz, L. Tóth, and A. Markó, "Ultrasound-based Silent Speech Interface Built on a Continuous Vocoder," in *Proc. Interspeech*, Graz, Austria, 2019, pp. 894–898.

[17] T. G. Csapó, C. Zainkó, L. Tóth, G. Gosztolya, and A. Markó, "Ultrasound-based Articulatory-to-Acoustic Mapping with WaveGlow Speech Synthesis," in *Proc. Interspeech*, 2020, pp. 2727–2731.

[18] L. Tóth and A. H. Shandiz, "3D Convolutional Neural Networks for Ultrasound-Based Silent Speech Interfaces," in *Proc. ICAISC*, Zakopane, Poland, 2020.

[19] A. H. Shandiz, L. Tóth, G. Gosztolya, A. Markó, and T. G. Csapó, "Improving Neural Silent Speech Interface Models by Adversarial Training," in *2nd International Conference on Artificial Intelligence and Computer Vision (AICV2021)*, 2021.

[20] J. A. Gonzalez, L. A. Cheah, A. M. Gomez, P. D. Green, J. M. Gilbert, S. R. Ell, R. K. Moore, and E. Holdsworth, "Direct Speech Reconstruction From Articulatory Sensor Data by Machine Learning," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 12, pp. 2362–2374, dec 2017.

[21] J. A. Gonzalez-Lopez, M. Gonzalez-Atienza, A. Gomez-Alanis, J. L. Perez-Cordoba, and P. D. Green, "Multi-view Temporal Alignment for Non-parallel Articulatory-to-Acoustic Speech Synthesis," in *Proc. IberSPEECH*, 2021, pp. 230–234.

[22] M. Janke and L. Diener, "EMG-to-Speech: Direct Generation of Speech From Facial Electromyographic Signals," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 12, pp. 2375–2385, dec 2017.

[23] L. Diener, G. Felsch, M. Angrick, and T. Schultz, "Session-Independent Array-Based EMG-to-Speech Conversion using Convolutional Neural Networks," in *13th ITG Conference on Speech Communication*, 2018.

[24] N. Shah, N. Shah, and H. Patil, "Effectiveness of Generative Adversarial Network for Non-Audible Murmur-to-Whisper Speech Conversion," in *Proc. Interspeech*, Hyderabad, India, 2018, pp. 3157–3161.

[25] S. Stone and P. Birkholz, "Silent-speech command word recognition using electro-optical stomatography," in *Proc. Interspeech*, San Francisco, CA, USA, 2016, pp. 2350–2351.

[26] Y. H. Shin and J. Seo, "Towards contactless silent speech recognition based on detection of active and visible articulators using IR-UWB radar," *Sensors*, vol. 16, no. 11, 2016.

[27] P. A. Digehsara, C. Wagner, P. Schaffer, M. Bärhold, S. Stone, D. Plettemeier, and P. Birkholz, "On the optimal set of features and robustness of classifiers in radar-based silent phoneme recognition," in *Proc. ESSV*, online, 2021.

[28] A. Ephrat and S. Peleg, "Vid2speech: Speech Reconstruction from Silent Video," in *Proc. ICASSP*, New Orleans, LA, USA, 2017, pp. 5095–5099.

[29] K. Sun, C. Yu, W. Shi, L. Liu, and Y. Shi, "Lip-Interact: Improving Mobile Device Interaction with Silent Speech Commands," in *UIST 2018 - Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*, Berlin, Germany, 2018, pp. 581–593.

[30] F. V. Arthur and T. G. Csapó, "Towards a practical lip-to-speech conversion system using deep neural networks and mobile application frontend," in *2nd International Conference on Artificial Intelligence and Computer Vision (AICV2021)*, 2021.

[31] T. G. Csapó, G. Németh, and M. Cernak, "Residual-Based Excitation with Continuous F0 Modeling in HMM-Based Speech Synthesis," in *Lecture Notes in Artificial Intelligence*, A.-H. Dediu, C. Martín-Vide, and K. Vicsi, Eds. Budapest, Hungary: Springer International Publishing, 2015, vol. 9449, pp. 27–38.

[32] T. G. Csapó, G. Németh, M. Cernak, and P. N. Garner, "Modeling Unvoiced Sounds In Statistical Parametric Speech Synthesis with a Continuous Vocoder," in *Proc. EUSIPCO*, Budapest, Hungary, 2016, pp. 1338–1342.

[33] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior, and K. Kavukcuoglu, "WaveNet: A Generative Model for Raw Audio," *CoRR*, vol. abs/1609.0, 2016.

[34] R. Prenger, R. Valle, and B. Catanzaro, "Waveglow: A Flow-based Generative Network for Speech Synthesis," in *Proc. ICASSP*, Brighton, UK, 2019, pp. 3617–3621.

[35] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan, R. A. Saurous, Y. Agiomvrgiannakis, and Y. Wu, "Natural TTS Synthesis by Conditioning Wavenet on MEL Spectrogram Predictions," in *Proc. ICASSP*, Calgary, Canada, 2018, pp. 4779–4783.

[36] J.-X. Zhang, K. Richmond, Zhen-Hua-Ling, and L.-R. Dai, "TaL-Net: Voice Reconstruction from Tongue and Lip Articulation with Transfer Learning from Text-to-Speech Synthesis," in *Proc. AAAI*, 2021.

[37] R. Mira, K. Vougioukas, P. Ma, S. Petridis, B. W. Schuller, and M. Pantic, "End-to-End Video-To-Speech Synthesis using Generative Adversarial Networks," apr 2021.

[38] M. S. Ribeiro, J. Sanger, J.-X. X. Zhang, A. Eshky, A. Wrench, K. Richmond, and S. Renals, "TaL: a synchronised multi-speaker corpus of ultrasound tongue imaging, audio, and lip videos," in *2021 IEEE Spoken Language Technology Workshop (SLT)*, Shenzhen, China, 2021, pp. 1109–1116.

[39] G. Olaszy, "Precíziós, párhuzamos magyar beszédadatbázis fejlesztése és szolgáltatásai [Development and services of a Hungarian precisely labeled and segmented, parallel speech database] (in Hungarian)," *Beszédkutatás 2013 [Speech Research 2013]*, pp. 261–270, 2013.

[40] G. Gosztolya, T. Grósz, L. Tóth, A. Markó, and T. G. Csapó, "Applying DNN Adaptation to Reduce the Session Dependency of Ultrasound Tongue Imaging-Based Silent Speech Interfaces," *Acta Polytechnica Hungarica*, vol. 17, no. 7, pp. 109–124, 2020.

[41] T. G. Csapó, G. Gosztolya, L. Tóth, A. H. Shandiz, and A. Markó, "Optimizing the Ultrasound Tongue Image Representation for Residual Network-based Articulatory-to-Acoustic Mapping," *submitted to Multimedia Tools and Applications*, 2021.

[42] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[43] S. Ji, W. Xu, M. Yang, and K. Yu, "3D convolutional neural networks for human action recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 1, pp. 221–231, 2012.

[44] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, nov 1997.

[45] J. Bradbury, S. Merity, C. Xiong, and R. Socher, "Quasi-recurrent neural networks," *arXiv preprint arXiv:1611.01576*, 2016.

[46] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri, "A closer look at spatiotemporal convolutions for action recognition," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2018, pp. 6450–6459.

[47] "ITU-R Recommendation BS.1534: Method for the subjective assessment of intermediate audio quality," 2001.

11th ISCA Speech Synthesis Workshop (SSW 11)
26-28 August 2021, Budapest, Hungary

Improving Emotional TTS with an Emotion Intensity Input from Unsupervised Extraction

Bastian Schnell[1,2], *Philip N. Garner*[1]

[1]Idiap Research Institute, Switzerland
[2]École Polytechnique Fédérale de Lausanne (EPFL), Switzerland
{bastian.schnell, phil.garner}@idiap.ch

Abstract

We aim to provide controls for emotion in synthetic speech. Many emotions are not displayed continuously in an otherwise emotional utterance; rather, the intensity varies with time. We show that an emotion recogniser is capable of producing a measure of emotion intensity via attention or saliency; this measure is appropriate to label utterances subsequently used to train a speech synthesiser. We evaluate novel and published means to do this showing that, whilst it is no longer state of the art for emotion recognition, attention is a good way to indicate emotion intensity for speech synthesis.

Index Terms: Emotional Speech Synthesis, TTS, Emotion Recognition, Saliency Mapping

1. Introduction

When text to speech synthesis (TTS) is used in a non-trivial application, it is desirable that the resulting synthetic speech conveys context-awareness using *affect*. For instance, in a speech to speech translation application, if the input speech (in L1) sounds, e.g., emphatic or angry, then the resulting speech (in L2) should convey the same qualities. In a dialogue application, the dialogue manager should be able to emphasise words that it wishes to clarify, and should respond to, say, frustration with empathy. Of course, this not only requires a suitably intelligent dialogue manager, but also a TTS system with controls for the appropriate affect variables. In this paper we are concerned with providing such controls for emotion.

While TTS systems have mastered human performance for neutral speech, emotional speech synthesis is still a challenge. For neutral speech large and high quality databases exist, but emotional databases are rare and mostly of low quality. It is certainly possible to record large amounts of a specific emotion and train the same systems as used for neutral speech. However, the range of emotions, varying intensities, the amount of languages, speaker variations, and the need to label each recording with the perceived emotion of multiple listeners makes recording alone a nearly infeasible task in terms of time and money. Modern emotional TTS research has identified three possible directions to solve these problems: 1) Increase the generalisability of the architectures on low data regimes; 2) increase the quantity of emotional data by voice or emotion conversion; and 3) increase the quality of the data. In the following we will highlight some recent work for each direction.

Databases with more expressive speech exist, especially audio books. Those databases cover a wider range of styles, but lack annotation of the expressed emotion or style. The lack of these annotations spawned a range of recent works focusing on increased model generalisability by utilising unsupervised methods to extract style embeddings from reference audio on a global [1, 2], clustered [3], or frame level [4, 5]. Some attempted controlling the expressiveness [1, 6]. However, controllability remains limited, especially for global embeddings.

Some work targets increasing the quantity of the expressive data. Huybrechts et al. [7] have used voice conversion to convert expressive recordings to the target speaker. In our recent work [8] we have converted neutral to emotional speech. The artificial data can then be used to train a TTS system.

We found limited work which attempts to increase the quality of the emotional data. Emotional databases usually have a single emotion label for every recording. We argue that this generalisation is misleading and that the emotion is localised within the utterance. This kind of annotation can lead to different emotion labels on words with lower emotional strength like conjunctions, while their acoustic features only marginally differ. Obviously, this impedes the learning of the model.

In this work we propose to add a frame-level emotion intensity to every sample, which is used as additional input to the TTS model. We present two methods to extract it from the recordings with pre-trained emotion recognisers. The simpler model contains a single attention layer, which allows use of the attention weights as emotion intensities. The other is a modern transformer model, where we exploit saliency maps to extract the intensity. The closest work to ours is that of Lei et al. [9]. They use relative attributes [10] to assign a level of emotional strength to each sample. In more recent work [11] they extended their method to phoneme level emotional strength.

We present our two methods for emotion intensity extraction as well as the method of attribute ranks of [11] in Section 2. We compare all three methods and a baseline without intensity input on the task of emotional TTS in Section 3. In this work we leave out the problem of generating the emotion intensity from text or extraction from a reference sample. Possible research directions to attack this problem are listed in the conclusions in Section 4.

2. Emotion intensity extraction

2.1. Attention LSTM

We use a simple emotion recogniser mostly resembling previous research [12] (Figure 1 left). It consists of a feature extraction part of 3 fully-connected layers with 256 neurons and a bidirectional LSTM (BiLSTM) with 128 neurons per direction. We apply dropout with a probability of 0.1 after each layer. Additionally, it contains an attention block with a single BiLSTM with 128 neurons per direction and a fully-connected layer without bias with a single output neuron. Its output respresents the unnormalised attention weights. As in previous work [12] we use a sigmoid activation, instead of the usual softmax, to obtain normalised attention weights. A sigmoid activation

10.21437/SSW.2021-11

ensures high activation levels over many frames, which leads to overall smoother attentions. This is especially desirable for our downstream task of emotional TTS. We use the predicted attention weights to compute a weighted sum over the outputs of the feature extraction part to create a single utterance level embedding of size 256. We pass this vector through a single fully-connected layer with as many neurons as emotion classes. All parameters are initialised using Xavier initialisation [13] with a uniform distribution, with one exception: The weights of the fully-connected layer with single output neuron in the attention block are initialised with samples from $\mathcal{N}(0, 0.1^2)$.

The openSMILE toolkit [14] is used to extract frame-level features (25 ms sliding window, 10ms shift). We use a 32-dim subset of the *IS09* feature subset composed of hand-crafted Low-Level Descriptors (pitch, energy, zero-crossing rate, voicing probability), 12 mel-frequency cepstral coefficients, and their first derivative. This subset is mean-variance normalised and forms the input to the emotion recogniser. To prevent overfitting we augment the input with random white noise with a standard deviation of 0.4. In contrast to previous research, this model accepts variable input lengths.

Training follows closely the procedure in previous work [12]. The model is trained with the Adam optimiser [15] ($\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 1\text{E}{-8}$) with a learning rate of $3\text{E}{-5}$ on mini-batches of 32 for 200 epochs with the cross-entropy loss. To account for class-imbalance we weight the cross-entropy for each class c by a factor of $w_c = \frac{N_{tot}}{N_{classes} N_c}$, where N_{tot} is the total number of training utterances, $N_{classes}$ the number of different classes/emotions, and N_c the number of utterances of class c in the training set. All but the LSTM layers are regularised with l_2-regularisation with a factor of $5\text{E}{-2}$. We select the best model based on the summed Weighted Accuracy (WA) and Unweighted Accuracy (UA).

We argue that this emotion recogniser, once trained, will attend to the emotional parts of the utterance to make a decision. Thus it is reasonable to assume that the attention weights over an utterance give a good approximation of the emotion intensity.

2.2. Transformer

The above model is a very simple emotion recogniser and does not represent the state of the art. More complex architectures exist which do not allow a straight forward extraction of attention weights. In this section we investigate a more recent transformer model [16]. It consists of multiple self-attention blocks, which do not allow the extraction of attention weights in an obvious way. We make no claim that this model is the best emotion recogniser currently available; rather, we present a technique representative of more complex models without restrictions to their architecture to extract emotion intensities.

The transformer (Figure 1 right) consists of a feature extraction block with 4 fully-connected layers with 512 neurons and SeLU activation. Afterwards a positional encoding is added in form of a sinusoid with a large period. Dropout with 0.1 probability is applied on the latent features, which is then fed to two self-attention [17] blocks with 32 heads each. The resulting attention matrix is aggregated with five 2D convolutional layers with [30, 30, 30, 10, 6] output channels, a 5×5 kernel size, and a stride of 2×2. The flattened 936-dim output is projected with a fully-connected layer with 936 neurons and a final fully-connected output layer with as many neurons as emotion classes. After each but the last layer in the aggregation step, dropout with probability 0.2 and SeLU activation is applied. All parameters are initialised using Xavier initialisation [13] with a uniform distribution.

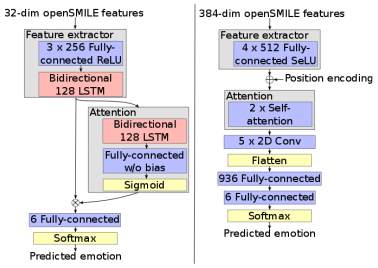


Figure 1: *Architectures of the emotion recognisers. Left: attention LSTM; right: transformer*

As before we use the openSMILE toolkit to extract frame-level features (25ms window, 10ms shift). However, we use the entire 384-dim *IS09* features subset as input to the transformer model and we do not add any noise. The transformer model requires a fixed-length input. We use a sliding window of 500 frames with a step size of 50 frames previously found to perform best [16]. At inference time the final prediction is made by applying a softmax on the predicted classes of each window and averaging the results. Sequences are zero padded to match the window and step size, no frames are dropped.

During training we randomly select 500 frames from each input in the batch. We use the Adam optimiser ($\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 1\text{E}{-8}$, no weight decay) with a learning rate of $1\text{E}{-5}$ for 170 epochs on a mini-batch size of 8 and PyTorch's *ReduceLROnPlateau* scheduler with default parameters.

To extract emotion intensities with the transformer model we propose to use saliency maps. Saliency maps are a common technique in vision-related machine learning tasks. They attempt to add interpretability to the neural network predictions. An increasing number of techniques exist with varying complexity [18, 19, 20, 21, 22]; we discuss some below. Saliency maps compute the importance of each input to the network's output, thus each openSMILE feature in each frame receives a value. To compute a scalar emotion intensity value we investigate the aggregation through *max* and *mean* operations. In the following we will give a high-level description of the techniques we use in our experiments (Section 3).

2.2.1. *Saliency Maps*

Input gradients [18] continues the backpropagation chain to the inputs and thus provides gradients of each input w.r.t. the correct class label. The idea is that the gradients indicate how much the class prediction is affected by a change in each input, thus representing its importance.

Since *input gradients* produces relatively noisy saliency maps **Smoothgrad** [19] attempts to smooth them over multiple observations. It achieves this by adding white noise to the input multiple times and computes the average input gradients for all iterations. The idea of *Smoothgrad* can be applied in many other saliency map techniques.

Input X Gradient [20] multiplies the *input gradients* with the input itself. The idea is that the gradient alone only indicates how important the feature is, but the input gives information on how strongly the feature is present. Together they provide a better abstraction of the feature importance.

Integrated Gradients [21] aggregates *input gradients* over

a linear interpolation between a baseline (the zero vector in our case) and the input. The idea is to capture *input gradients* that were steep at some of the interpolations but became flat for the input, as they are still important for the class prediction.

2.3. Attribute Rank

Recent work [9, 11] has used attribute ranks [10] to compute emotion intensities. We include this work as a competitive method here and give a brief overview. For data of two categories the ranking function computes the ranking/order of the data w.r.t. to a certain attribute, here emotion intensity. Once the ranking function is learned, it can assign an emotion intensity level to unseen emotional data. For completeness we give an example closely following that in [11].

We select all neutral N and happy H samples from the training set with acoustic features x_t with $t \in [1, \ldots, T]$ with $T = |N \cup H|$. We then form an ordered set O and an unordered set S of pairs. In the ordered set we pair an emotional sample of H with a neutral sample from N, indicating that the emotion intensity is higher in the samples of H than in those of N. In the unordered set we randomly create pairs of neutral-neutral and happy-happy samples, indicating that their rank should be similar. The goal is to learn a ranking function $r(x_t) = wx_t$ satisfying the following constraints as much as possible

$$\begin{array}{l} \forall(i,j) \in O : wx_i > wx_j \\ \forall(i,j) \in S : wx_i = wx_j \end{array} \quad (1)$$

The problem can be relaxed with slack variables ξ_{ij} and γ_{ij} and solved by Newton's method.

In [11] a single openSMILE feature vector x_t is extracted for each utterance. Then the ranking function, i.e. the ranking vector w_m with $m \in [1, \ldots, M]$, is learned for each combination of neutral with the other M emotions. To obtain phoneme-level rankings openSMILE features are extracted for the segments corresponding to each phoneme. This requires a forced-alignment step for which we use the Montreal Forced Aligner [23]. We use a Python port[1] of the original code[2] of [10] with the default parameters for the Newton algorithm.

3. Experiments

For our experiments we select the SAVEE database [24]. It is an audio-visual British English database with sentences from TIMIT phonetically-balanced for each emotion. For each emotion 3 common, 2 emotion-specific, and 10 generic sentences (different for each emotion) were taken. For neutral the 3 common and $2 * 6$ emotion-specific sentences were additionally recorded, giving 30 neutral sentences in total. 4 males acted in 7 different emotions (neutral, anger, disgust, fear, happiness, sadness, and surprise) resulting in a total of 480 utterances. The audio was recorded at 44.1 kHz and has higher quality compared to most emotional databases. We do not use the visual information of the database. To compensate for loudness differences in speaker 'KL' we use a loudness normalization technique to normalize all samples to an average root-mean squared value of $RMS = 0.1$ with $\tilde{x} = x * \sqrt{(T * RMS^2)/(\sum^T (x - x_{mean})^2)}$. We also found background noise to degrade performance in some of the recordings. To reduce the noise we use a single channel spectral enhancement scheme [25] to pre-process the entire database.

[1] https://github.com/chaitanya100100/Relative-Attributes-Zero-Shot-Learning

[2] https://www.cc.gatech.edu/ parikh/relative.html

3.1. Emotion Intensity

To train emotion recognisers, the SAVEE database is rather limited. Thus we include the IEMOCAP [26] database in all strategies for emotion intensity extraction. It splits into 5 dialogue sessions of acted and spontaneous emotions with 2 different professional actors each, totalling 10 speakers and approximately 12 hours of 48 KHz recordings. At least 3 fluent English speakers annotated the perceived emotion and the final emotion label was chosen based on majority vote. While still in the database we exclude samples where no majority label was found, additionally we exclude the 'disgusted' emotion from our experiments, as it is both very hard to express and very rare in the database. We apply the same loudness normalization and noise reduction techniques as on SAVEE.

3.1.1. Emotion Recogniser

We train the attention LSTM (Section 2.1) and transformer (Section 2.2) emotion recogniser models on IEMOCAP with the parameters and inputs as defined in their respective section, using a random split of the 5th session for the validation and test set. We then fine-tune the models on SAVEE with the same parameters for 200 epochs and select the best model based on combined WA and UA on the validation set. For each emotion we select emotion specific utterances as test and validation set. Namely we use the 4th and 5th id as test set and the 6th and 7th id as validation set. We select the same ids for all speakers so that the content is unseen during training. Table 1 shows the metrics of the trained models on IEMOCAP and SAVEE. With the trained models we extract the emotion intensity. For the attention LSTM model these are simply the attention weights.

Table 1: *Weighted (WA) and Unweighted Accuracy (UA) of the emotion recogniser models after pre-training on IEMOCAP and fine-tuning on SAVEE excluding the disgusted emotion class.*

	IEMOCAP		**SAVEE**	
	WA	UA	WA	UA
Attention LSTM	54.7	40.3	62.5	60.4
Transformer	51.2	43.1	69.6	67.7

Table 2: *MSE between saliency maps and attention weights extraction on the attention LSTM model on SAVEE. Saliency maps abbreviated as IG: Input Gradient, Sg: Smoothgrad, IxG: Input x Gradient, IntG: Integrated Gradients*

Aggr.	**Smoothed**	**IG**	**Sg**	**IxG**	**IntG**
mean	no	1.46	1.451	1.626	1.62
mean	yes	0.625	**0.621**	1.179	1.312
max	no	1.466	1.461	1.5	1.56
max	yes	0.665	0.664	0.835	0.984

For the transformer model the variety of saliency maps (Section 2.2.1) allows multiple intensity curves (Figure 2). We extract emotion intensity using Input Gradients, Smoothgrad, Input X Gradient, and Integrated Gradients with *max* and *mean* aggregation. As the saliency maps can be noisy, we also experiment with smoothed versions obtained by a simple convolution with an 11 frames wide Hanning window (Figure 3). With informal listening we cannot select a best system. However, we find that the intensity weights extracted with the attention LSTM

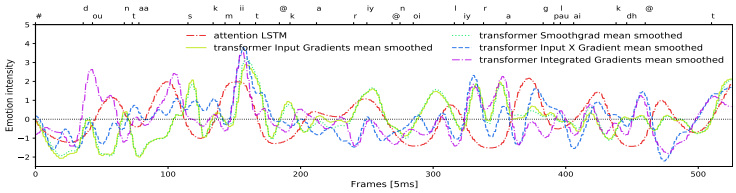


Figure 2: *Emotion intensities extracted with the attention LSTM model and different smoothed saliency maps for an angry utterance of speaker JK. For better comparison each intensity is mean-variance normalised based on its own statistics. The content is: "Don't ask me to carry an oily rag like that."*

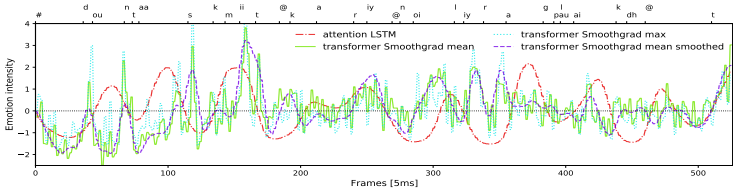


Figure 3: *Emotion intensities extracted with the attention LSTM model and with the Smoothgrad saliency map with max and mean aggregation as well as smoothed mean for the same utterance as in Figure 2*

model consistently produce more expressive speech than those extracted with saliency. Thus it is reasonable to interpret the saliency map as an approximation of the attention weights and select the saliency map which is closest to them. For that reason we extract the saliency maps on the attention LSTM model and compare them to the attention weights in terms of Mean-Squared-Error (MSE). As can be seen in Table 2 the closest saliency map is smoothgrad with smoothed *mean* aggregation.

3.1.2. Attribute Rank

While it is possible to learn the ranking just on the SAVEE database, we also include the IEMOCAP database for a fair comparison. Indeed, we found that rankings extracted on both databases outperform those learned only on SAVEE in informal listening tests. We exclude the SAVEE samples later used for validation/test set of the emotional TTS model (Section 3.2) when learning the ranking function. To form the unordered set we randomly form pairs for each sample in the neutral set of SAVEE. We then fill up the set with pairs from IEMOCAP (speaker independent selection) to reach 150 pairs. We perform the same with the respective emotion to obtain an unoredered set with 300 pairs. For the ordered set we randomly select a neutral SAVEE sample for each emotional SAVEE sample and again use IEMOCAP to fill up to 300 pairs. This procedure follows the one in [11].[3] With the learned ranking function we compute phoneme-level rankings for all SAVEE samples.

3.2. Emotional TTS

Our goal is to train an emotional TTS system with emotion intensity input on the SAVEE database. Due to the small size of SAVEE we cannot train a modern encoder-decoder network on it, as it quickly overfits before adapting the new speaking styles. Instead we rely on a classical RNN-based network, which has also been used in recent studies on emotional speech synthesis [27]. We use oracle durations in all our experiments, because duration prediction for emotional speech is a challenging problem on its own. The model consists of 2 fully-connected layers with ReLU activation and 1024 neurons, 3 BiLSTM layers with 512 neurons, and the final 97 dimensional output layer. 5% dropout is applied in all but the final layer. A 128-dim speaker and 64-dim emotion embedding is concatenated to the input of each layer. Additionally, we concatenate the mean-variance normalised emotion intensity input in all layers, which gives better results than concatenating it only to the input. For all neutral samples we set the emotion intensity to zero, indicating that there is no emotion present. We do not predict the emotion intensity internally, because we want to keep it as a tunable input.

The inputs to the model are 425 text-derived binary and numerical features normalised to $[0.01, 0.99]$, which were derived from the forced-aligned (with HTK [28]) phoneme sequence previously extracted with Festival [29]. The model predicts mean-variance normalised WORLD vocoder [30] features, consisting of linearly-interpolated log F_0 , a voiced/unvoiced flag, 30-dimensional mel-generalised cepstrum, and one Band Aperiodicity at 5 ms frame step, with their delta and double delta derivatives. The output is smoothed with the MLPG algorithm [31]. The WORLD vocoder is used to generate the waveform.

Even for our model the SAVEE database is too small to train a TTS system, so we instead pre-train on the WSJCAM0 database [32]. It is a large British English database with 92 speakers with 90 utterances each recorded at 16 kHz. We use only the head-mounted close-talking microphone recordings. We apply the same loudness normalisation and noise reduction techniques as on IEMOCAP and SAVEE (Section 3). The model is pre-trained for 35 epochs with a batch size of 16 and a learning rate of 0.001 and early stopping. We reduce the learning rate by a factor of 0.1 on validation loss plateaus. The adaptation to SAVEE is split into adaptation to the neutral subset of SAVEE first, and the entire database second. Each step is further divided into three phases. In the first phase only the speaker embedding is trained (10 epochs, lr=0.001), in the second phase

[3] We thank Shan Yang for the detailed description of the process.

the whole model is trained (10 epochs, lr=0.001), the last phase applies fine-tuning by repeating phase two with a smaller learning rate (10 epochs, lr=0.0001). The batch size in all phases is 16. In each phase early stopping is used and the best model is selected to continue with the next phase.

3.3. Subjective Results

In the subjective listening test we investigate how the TTS models compare in terms of perceived emotion and whether the audio quality is impacted. For the test we include five systems:

- **baseline**: TTS model without emotion intensity input
- **attention**: Attention weights from the attention LSTM
- **transformer**: Smoothgrad saliency map with mean aggregation and smoothing extracted with the transformer
- **rank**: Phoneme-level rankings extracted with the competitive technique [11]
- **ref**: Copy synthesis of the recordings

The test set consists of the same two utterances recorded for every emotion (7, including neutral, excluding disgusted) and every speaker (4 males). This makes 56 samples for each system. As we do not yet have a method to predict emotion intensity from text, we use the emotion intensity extracted from the reference audio by the respective technique. This gives an upper bound on the quality achievable with an emotion intensity input assuming that the prediction is perfect. We find that the emotion intensity input does not increase the expressiveness of the speech much. However, it offers an unprecedented control to tune the emotion intensity. Informal listening shows a greatly increased expressiveness, while still sounding natural, when scaling the input with a factor of **7**. The models have learned to connect certain speech properties with the intensity input, which allows scaling them in a natural way. In general higher intensities result in higher energy in the speech, which is desirable for all but the sad emotion. Thus all our tests use the scaled version except sadness.

36 listeners rated 25 randomly selected samples each in a 5-scale MOS test with 0.5 steps and also selected the emotion they perceived. Table 3 summarises the results. The *total* column includes the correct ratings on neutral. As many subtle emotions like fearful or surprised are rated as neutral, this number is biased. The *emo* column indicates the accuracy on the emotional samples only. On both metric the attention weight extracted with the attention LSTM model outperforms the other systems. It shows that an emotion intensity input increases the expressiveness of the speech, which is also perceivable by listeners. The *happy* emotion is almost never perceived. The low recognition rate of the reference samples indicates that it was not acted well enough. Providing a neutral reference during the listening test might facilitate its prediction.

It also shows that the quality of the emotion intensity matters as the phoneme level rankings perform much worse, this might be due to the phoneme-level granularity. The key benefit of the ranking function is that it requires very little training data. It might perform best when we do not include any IEMOCAP data. It outperforms the baseline system in a similar manner to that reported in the related work [11].

Interestingly, the saliency map extracted from the transformer model performs worse then the simple attention weight, even though the model is much more complex and achieves higher emotion recognition scores. All the saliency map techniques are developed for the field of vision, focusing on convolutional layers. A different type of saliency map might be necessary for speech tasks or more convolutional networks might allow better saliency maps. The benefit of the transformer model is that it will likely improve its emotion recognition performance with more training data compared to the attention LSTM model due to its small complexity. However, as long as no proper saliency map technique exists, we are limited to models that allow straight-forward extraction of emotion intensity.

Table 3: *Results of the subjective evaluation of perceived emotions in percentage. 'total' includes the neutral samples. Accuracy for each emotion is shown as well labelled as n: neutral, a: angry, f: fearful, h: happy, sa: sad, su: surprised.*

System	total	emo	n	a	f	h	sa	su
baseline	25.3	17.6	**72**	28	15	3	33	12
attention	**35.5**	**28.9**	70	**54**	13	**6**	**55**	**21**
transformer	26.7	20.6	60	33	**25**	0	41	8
rank	25.3	19.0	69	30	14	**6**	40	8
ref	45.9	40.3	75	74	23	19	31	54

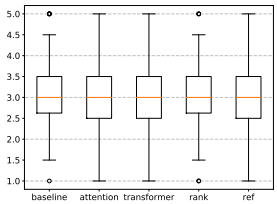


Figure 4: *Results of the 5-scale MOS test with 0.5 steps*

Figure 4 shows the results of the MOS test. None of the differences in the results are statstically significant in a two-tailed paired t-test with a p-value < 0.05. This includes the copy synthesis reference, which has other quality issues that were rated low by listeners. We can conclude that the proposed techniques do not deteriorate the audio quality.[4]

4. Conclusion and Future Work

We presented two techniques to extract an emotion intensity input from audio in an unsupervised way by utilising pre-trained emotion recognisers. We do not require emotion intensity labeling, but only emotion class labels. Thus one could also refer to it as weak supervision. From an emotion recognition network with a single attention layer we extract the attention weights as emotion intensity. From a transformer-based network we extract it using saliency maps. We show that the additional emotion intensity input improves an emotional TTS system; increasing the accuracy of which human listeners perceive the target emotion without degradation in signal quality. The simpler first method outperforms all others, including a recently published method for emotion intensity extraction by *relative attributes*.

For the tests we use oracle emotion intensity extracted from the reference. As the results show great improvements with an emotion intensity input, future research will focus on predicting it from text or conversion in speech-to-speech translation.

Ack.: This work was supported by the Swiss NSF grant number 185010: Neural Architectures for Speech Technology (NAST); http://p3.snf.ch/Project-185010

[4] Audio samples at www.idiap.ch/paper/ssw11_emotion_intensity/

5. References

[1] Y. Wang, D. Stanton, Y. Zhang, R.-S. Ryan, E. Battenberg, J. Shor, Y. Xiao, Y. Jia, F. Ren, and R. A. Saurous, "Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis," in *International Conference on Machine Learning*. PMLR, 2018, pp. 5180–5189.

[2] R. Skerry-Ryan, E. Battenberg, Y. Xiao, Y. Wang, D. Stanton, J. Shor, R. Weiss, R. Clark, and R. A. Saurous, "Towards end-to-end prosody transfer for expressive speech synthesis with tacotron," in *international conference on machine learning*. PMLR, 2018, pp. 4693–4702.

[3] V. Klimkov, S. Ronanki, J. Rohnke, and T. Drugman, "Fine-grained robust prosody transfer for single-speaker neural text-to-speech," *Proc. Interspeech 2019*, pp. 4440–4444, 2019.

[4] S. Choi, S. Han, D. Kim, and S. Ha, "Attentron: Few-Shot Text-to-Speech Utilizing Attention-Based Variable-Length Embedding," in *Proc. Interspeech 2020*, 2020, pp. 2007–2011. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2020-2096

[5] Y. Lee and T. Kim, "Robust and fine-grained prosody control of end-to-end speech synthesis," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 5911–5915.

[6] S.-Y. Um, S. Oh, K. Byun, I. Jang, C. Ahn, and H.-G. Kang, "Emotional speech synthesis with rich and granularized control," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7254–7258.

[7] G. Huybrechts, T. Merritt, G. Comini, B. Perz, R. Shah, and J. Lorenzo-Trueba, "Low-resource expressive text-to-speech using data augmentation," *arXiv preprint arXiv:2011.05707*, 2020.

[8] B. Schnell, G. Huybrechts, B. Perz, T. Drugman, and J. Lorenzo-Trueba, "EmoCat: Language-agnostic emotional voice conversion," *arXiv preprint arXiv:2101.05695*, 2021.

[9] X. Zhu, S. Yang, G. Yang, and L. Xie, "Controlling emotion strength with relative attribute for end-to-end speech synthesis," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2019, pp. 192–199.

[10] D. Parikh and K. Grauman, "Relative attributes," in *2011 International Conference on Computer Vision*. IEEE, 2011, pp. 503–510.

[11] Y. Lei, S. Yang, and L. Xie, "Fine-grained emotion strength transfer, control and prediction for emotional speech synthesis," *arXiv preprint arXiv:2011.08477*, 2020.

[12] G. Ramet, P. N. Garner, M. Baeriswyl, and A. Lazaridis, "Context-aware attention mechanism for speech emotion recognition," in *2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2018, pp. 126–131.

[13] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 2010, pp. 249–256.

[14] F. Eyben, F. Weninger, F. Gross, and B. Schuller, "Recent developments in opensmile, the munich open-source multimedia feature extractor," in *Proceedings of the 21st ACM international conference on Multimedia*, 2013, pp. 835–838.

[15] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2015. [Online]. Available: http://arxiv.org/abs/1412.6980

[16] L. Tarantino, P. N. Garner, and A. Lazaridis, "Self-attention for speech emotion recognition." in *Interspeech*, 2019, pp. 2578–2582.

[17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, pp. 6000–6010.

[18] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," *arXiv preprint arXiv:1312.6034*, 2013.

[19] D. Smilkov, N. Thorat, B. Kim, F. Viégas, and M. Wattenberg, "Smoothgrad: removing noise by adding noise," *arXiv preprint arXiv:1706.03825*, 2017.

[20] A. Shrikumar, P. Greenside, A. Shcherbina, and A. Kundaje, "Not just a black box: Learning important features through propagating activation differences," *arXiv preprint arXiv:1605.01713*, 2016.

[21] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," in *International Conference on Machine Learning*. PMLR, 2017, pp. 3319–3328.

[22] S. Srinivas and F. Fleuret, "Full-gradient representation for neural network visualization," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32. Curran Associates, Inc., 2019. [Online]. Available: https://proceedings.neurips.cc/paper/2019/file/80537a945c7aaa788ccfcdf1b99b5d8f-Paper.pdf

[23] M. McAuliffe, M. Socolof, S. Mihuc, M. Wagner, and M. Sonderegger, "Montreal forced aligner: Trainable text-speech alignment using kaldi," in *Interspeech*, vol. 2017, 2017, pp. 498–502.

[24] S. Haq, P. J. Jackson, and J. Edge, "Audio-visual feature selection and reduction for emotion classification," in *Proc. Int. Conf. on Auditory-Visual Speech Processing (AVSP08), Tangalooma, Australia*, 2008.

[25] B. Cauchi, I. Kodrasi, R. Rehr, S. Gerlach, A. Jukić, T. Gerkmann, S. Doclo, and S. Goetze, "Combination of MVDR beamforming and single-channel spectral processing for enhancing noisy and reverberant speech," *EURASIP Journal on Advances in Signal Processing*, vol. 2015, no. 1, p. 61, 2015.

[26] C. Busso, M. Bulut, C.-C. Lee, A. Kazemzadeh, E. Mower, S. Kim, J. N. Chang, S. Lee, and S. S. Narayanan, "IEMOCAP: Interactive emotional dyadic motion capture database," *Language resources and evaluation*, vol. 42, no. 4, pp. 335–359, 2008.

[27] J. Lorenzo-Trueba, G. E. Henter, S. Takaki, J. Yamagishi, Y. Morino, and Y. Ochiai, "Investigating different representations for modeling and controlling multiple emotions in DNN-based speech synthesis," *Speech Communication*, 2018.

[28] P. C. Woodland, J. J. Odell, V. Valtchev, and S. J. Young, "Large vocabulary continuous speech recognition using HTK." in *ICASSP (2)*, 1994, pp. 125–128.

[29] A. Black, P. Taylor, R. Caley, and R. Clark, "The festival speech synthesis system," 1998. [Online]. Available: http://www.cstr.ed.ac.uk/projects/festival/

[30] M. Morise, F. Yokomori, and K. Ozawa, "WORLD: a vocoder-based high-quality speech synthesis system for real-time applications," *IEICE TRANSACTIONS on Information and Systems*, vol. 99, no. 7, pp. 1877–1884, 2016.

[31] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura, "Speech parameter generation algorithms for HMM-based speech synthesis," in *Acoustics, Speech, and Signal Processing, 2000. ICASSP'00. Proceedings. 2000 IEEE International Conference on*, vol. 3. IEEE, 2000, pp. 1315–1318.

[32] J. Fransen, D. Pye, T. Robinson, P. Woodland, and S. Young, "WSJCAM0 corpus and recording description," *Cambridge University Engineering Department (CUED), Speech Group, Trumpington Street, Cambridge CB2 1PZ, UK, Tech. Rep. CUED/F-INFENG/TR*, vol. 192, 1994.

11th ISCA Speech Synthesis Workshop (SSW 11)
26-28 August 2021, Budapest, Hungary

Acquiring conversational speaking style from multi-speaker spontaneous dialog corpus for prosody-controllable sequence-to-sequence speech synthesis

Avrech Ben-David[1]*, *Slava Shechtman*[2]

[1]Technion – Israel Institute of Technology, Haifa – Israel
[2]IBM Haifa Research Lab, Haifa – Israel

avrech@campus.technion.ac.il, slava@il.ibm.com

Abstract

Sequence-to-Sequence Text-to-Speech (S2S TTS) architectures that directly generate low level acoustic features from phonetic sequence are known to produce natural and expressive speech, when provided with moderate-to-large amounts of high quality training data. When exposed to a sequence of coarse speaker-agnostic *prosodic descriptors*, such systems become prosody-controllable and can learn and transfer desired prosodic patterns (e.g. word-emphasis or speaking style) from one seen speaker to another (in multi-speaker settings).

But what if a high quality speech corpus for a desired speaking style is not available? In this work we explore the feasibility of teaching a neutral pre-trained prosody-controllable S2S TTS voice to speak with a conversational speaking style, as learnt from a low-quality multi-speaker spontaneous dialog corpus (originally intended for Automatic Speech Recognition). We have found that it is absolutely necessary to incorporate word semantics for that task. We fine-tune BERT network to predict the *prosodic descriptors* from the input text, based on that corpus, and apply them to the prosody-controllable S2S TTS at inference time. The subjective listening tests revealed that the learnt conversational style rated higher than baseline for 68% of the stimuli under test. The overall quality and naturalness rated higher than baseline in 64% of the stimuli under test. The improvement came mostly as a result of improving common conversational speech patterns, such as filler words and phrases. However, the overall MOS did not significantly improve due to less convincing realization of the rising intonation on declarative statements (*uptalk*).

Index Terms: expressive speech synthesis, sequence to sequence speech synthesis, conversational speech synthesis

1. Introduction

Sequence-to-Sequence Text-to-Speech (S2S TTS) architectures [1] [2] that directly generate low level acoustic features from phonetic sequence are known to produce natural and expressive speech, if provided with sufficient amount of high quality training data, covering a variety of speakers and speaking styles. Apparently, additional high quality expressive data is required for the S2S TTS to acquire a new speaking style for existing voices, to perform model retraining or adjustment.

But what if a high quality speech corpus for a desired speaking style is not available? Acquiring a new speaking style for existing voices in a pre-trained S2S TTS remains a hot research topic. Cross-speaker speaking style transplantation by means of style encoding of a single reference utterance, as proposed initially in [3], partially achieved that goal. However, it worked mostly when the text of the reference utterance closely matched the text to be synthesized [3], thus making this method less appropriate for standard TTS applications. Since then, several methods have been developed to apply various speaking styles, in unsupervised [4] or semi-supervised [5] configurations, when style encoding is jointly trained with S2S TTS. Such settings require high quality speech data to deduce speaking styles from the corpus. In this work, on the contrary, we explore the feasibility of acquiring an unseen speaking style from a readily available low-quality speech corpus, that seems unsuitable for high quality TTS purposes.

The style that we'd like to obtain is a conversational speaking style. Recently, a single-speaker conversational S2S TTS system has been proposed, trained on a proprietary high-quality spontaneous data corpus, recorded particularly for that purpose [6]. In this work, on the other hand, we'd like to explore a less expensive approach, making use of existing data. For that purpose we selected *Switchboard* [7], a well-known multi-speaker corpus of narrow-band spontaneous speech, originally intended for Automatic Speech Recognition (ASR) development purposes. It is a corpus of spontaneous conversations of telephone bandwidth speech. The corpus contains 2430 conversations averaging 6 minutes in length, spoken by over 500 US English speakers. The calls are manually transcribed and then submitted to an ASR system to establish approximate time alignments at the word level [7]. The recordings are truly spontaneous, with a lot of background noises, prolonged pauses, word repetitions, filler words, paralinguistics and burst-ins.

As such, this corpus cannot serve directly for high quality S2S TTS system training, but would rather be utilized for certain intermediate style representation. Considering the data set characteristics, this representation should be 1) purely prosodic, as *Switchboard*'s general spectral characteristics are very different from that of high quality wide band studio recordings used for S2S TTS voices, and 2) speaker- and gender-agnostic, as this multi-speaker dataset has only few stimuli per speaker available and we want to learn general conversational style aspects. Fortunately, our prosody-controllable S2S TTS architecture, originally proposed for unsupervised/weakly-supervised word-emphasis realization [8] is suitable for that purpose. In this architecture we condition the speech synthesis on an intermediate prosodic representation, *Hierarchical Prosodic Controls (HPC)*, comprising a sequence of hierarchical (word- and sentence- level) prosodic observations, designed to be gender- and speaker-agnostic.

In the current work we deploy an extended set of HPC parameters to better fit the desired speaking style application and design an HPC predictor model, trained on *Switchboard*. The predicted HPC sequences are utilized to condition the pre-trained prosody-controllable S2S TTS to convey the desired conversational speaking style. We explore various alternatives for conversational HPC prediction from phonetic and/or textual

*Work performed as an internship at IBM

10.21437/SSW.2021-12

input, incorporating LSTM [9] and/or BERT [10] networks. We introduce the system architecture in Sec. 2, detail on training procedure in Sec. 3 and present system evaluation in Sec. 4. Concluding remarks are provided in Sec. 5.

2. Architecture

2.1. Prosody-Controllable S2S TTS

The Sequence to Sequence Text To Speech model architecture, adopted in this work (Fig. 1), mostly follows the prosody-controllable S2S model originally proposed for unsupervised/weakly-supervised word-emphasis realization [8]. It is based on a Tacotron2 S2S acoustic model [2], augmented with Hierarchical Prosodic Controls [8]. The S2S acoustic model generates a sequence of acoustic feature vectors (composed of mel-cepstral and periodicity components [11, 12]), where each vector corresponds to a constant-length speech frame, that are then fed to an independently trained LPCNET-based neural vocoder [11] to generate high-quality samples in real time [12]. The inputs to the system are a set of symbolic sequences extracted from the input text by a rules-based TTS Front End module (adopted from a unit selection system [13]). All input sequences are aligned (by repetition) to contain the same number of symbols and are *one-hot* coded. The input sequences comprise:

- (A) phone identity (including silence phone) together with its lexical stress (primary, secondary or no stress)
- (B) phrase type (4-way: intermediate, declarative, interrogative, exclamation)

All the symbolic sequences are augmented with a special symbol for word boundary, inserted between the words with no silence between them. The *one-hot* coded input sequences are converted to a set of linear embeddings, concatenated together, and fed into Tacotron2 Encoder module (C), consisting of convolutional and bidirectional Long Short-Term Memory (Bi-LSTM) layers [2]. A global utterance-level speaker embedding (E), broadcast over the length of the sequence, is concatenated to the encoder output.

A set of Hierarchical Prosodic Controls, extended from the one introduced in [8] (and further elaborated in Sec. 2.2), is designed to enable both the sentence-level and the word-level modifications needed to realize the prosodic patterns associated with various speaking styles. They are designed to be speaker-agnostic to ease cross-speaker style transplantation. During training these prosodic controls are extracted from the target waveforms (E), while at inference time a separate predictive module (D) steps in to provide default predictions for the hierarchical prosodic trajectories.

The Decoder is an autoregressive network that largely follows the standard Tacotron2 architecture [2], but with modifications on the attention mechanism, choice of targets, and training losses. These are described in detail in [12] and briefly summarized as follows. The attention is an augmented two-stage attention where the hybrid content- and location-based attention mechanism of Tacotron2 [2] is followed by a structure-preserving mechanism encouraging monotonicity and unimodality in the alignment matrix [12]. The model is trained in a multi-task fashion to predict the end-of-sequence indicator and 80-dim mel cepstral features [2] in tandem with the parameters needed as inputs for an independently trained LPCNET neural vocoder [11]. For 22kHz signals, these features (which we denote as "LPC features") correspond to 256 waveform samples and consist of a 22-dim vector with 20 mel-cepstral coefficients, log f0 and f0 correlation. The predicted LPC features are also processed with two post-nets (one to refine the mel-cepstrum, and one to refine the pitch parameters). As opposed to [8, 12], the autoregressive feedback mechanism in the decoder is kept unmodified from the original Tacotron2 architecture.

Let y_t^M and y_t^L represent the target sequences for the mel and LPC tasks respectively, $\tilde{y}_t^M$ and $\tilde{y}_t^L$ their final predictions, and $\hat{y}_t^L$ the "intermediate" LPC-feature prediction (before the post-net). Then the combined acoustic loss function is used to train the system:

$$\mathcal{L} = MSE(\tilde{y}_t^M, y_t^M) + 0.8MSE(\hat{y}_t^L, y_t^L) + 0.4MSE(\tilde{y}_t^L, y_t^L) + 0.4MSE(\Delta\tilde{y}_t^L, \Delta y_t^L), \quad (1)$$

where the Δ operator applies the first difference in time to a sequence, and $MSE(,)$ is the mean-squared error. The above combined acoustic loss is added to the end of-sequence indicator cross-entropy loss [2] to yield the total training loss. For the sake of space, we omit some detail in this exposition, and refer the reader to [14, 12] for additional background and formulae.

The default prosodic-control predictor predicts the HPCs from the Front-End Encoder outputs of the S2S acoustic model (Fig. 1) and consists of stacked Bi-LSTMs (3x128), terminated with a linear layer. The predictor is trained separately (after the training of the main model has ended and all its weights are frozen) with *global MSE loss* of the combined HPC sequence (see section 2.2) and ADAM [15] optimizer. At inference time, the predictions of the prosodic-control subnet are rectified to be piecewise constant as the oracle values that the S2S system was trained with. To that end, a mean pooling function is applied to the prediction to be constant between the (known) sentence and word boundaries.

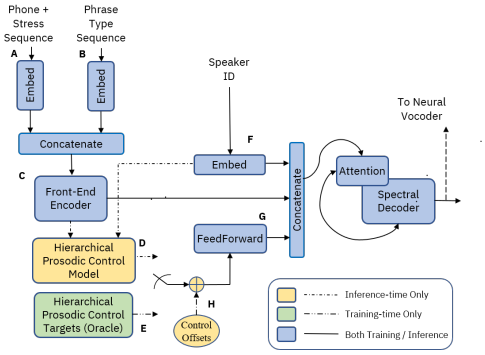


Figure 1: *Multi-speaker S2S synthesis acoustic model (phones to spectra) with Hierarchical Prosodic Controls.*

2.2. Hierarchical Prosodic-Control parameters

In this work we extend a set of four perceptually-interpretable prosodic measurements introduced in [8], evaluated over sentence and word intervals. The sentence-level components represent general speed and expressiveness [14], while the word-level components [8] represent fine-grained prosodic structure. In this work we extend the reported set of prosodic measurements [8] with two more pitch slope components to better suit

for speaking-style modeling. Altogether, we make use of the following statistics:

- S_{dur}: The log of the average per-phone durations, along a sentence (and excluding any silence).
- $S_{\Delta f_0}$: The f_0 dynamics (i.e., the difference between the 95- and 5-percentiles of log-f_0), along a sentence.
- $S_{\angle f_0}$: The log-f_0 linear regression slope along a sentence (excluding any silence).
- W_{dur}: The log of the average per-phone durations (as above), along each word.
- $W_{\Delta f_0}$: The f_0 dynamics (as above), along each word.
- $W_{\angle f_0}$: The log-f_0 linear regression slope (as above), along each word.

Note that the average per-phone durations in the above definitions are estimated as the duration (in seconds) of the relevant spans (word or sentence) divided by the number of phone symbols contained therein, and that therefore no fine-level phonetic alignment is required in the computation (only coarse word-level alignments and either phonetic transcriptions or a dictionary). The above sentence- and word-level properties are propagated down to the temporal granularity of the phonetic encoder outputs (i.e., phones) to form piecewise functions that are constant within a (sentence or word) unit. From this we define the following six-component prosodic-control target vector:

$$P = Norm_{\sigma}\{[S_{dur}, S_{\Delta f_0}, S_{\angle f_0}, W_{dur} - S_{dur}, W_{\Delta f_0} - S_{\Delta f_0}, W_{\angle f_0} - S_{\angle f_0}]\}, \quad (2)$$

where $Norm_{\sigma}\{\}$ is the linear map $[-3\sigma^2, 3\sigma^2] \rightarrow [-1, 1]$, and σ^2 is the global (multi-speaker corpus-wide) variance for each of the statistics in P. Note that all measurements in P are gender-agnostic.

2.3. Conversational HPC Prediction Model

For each utterance we predict an HPC sequence, comprising sentence- and word-level pitch- and duration-related features, as described in section 2.2. Silences are treated as special words for which only duration-related features are required. In order to adapt the prosodic controls to the conversational context, we consider combining various inputs: the input text, its phonetic sequence encoding, textual dialog context (casual) and prosodic dialog context, represented by HPCs, extracted from the past dialog audio. We use an encoder-decoder architecture, comprising the encoder (Fig. 2) that converts all the input streams into a sequence of context-aware word-embeddings, followed by a decoder (Fig.3) that contains three dedicated HPC decoder networks: word-level HPCs, sentence-level HPCs and a dedicated decoder for silence words. The encoder consists of three neural models: a BERT network for text processing and two distinct Bidirectional LSTM stacks for processing the phone-encoding and prosodic context input sequences.

2.3.1. Text encoding (with conversational context)

BERT is a widely used language model for language understanding tasks. The pre-trained model is commonly fine-tuned for a few epochs to extract high-quality task-specific word embedding [10]. BERT requires transforming the input text into tokens, e.g. with WordPiece tokenization [16]. In our work we deploy base (uncased) BERT model in its "question answering" configuration; we feed a window of the chat history, i.e. the textual *context*, into BERT's *sentence A* input, while the target utterance is fed as its *sentence B*. Attending to the conversational *context*, BERT produces token embeddings for the target utterance. We average the outputs of the 4 last hidden layers to produce token-representation, and represent each original word by its first token representation. We show in our experiments that all systems incorporating BERT text processing perform significantly better than the system that processes just phone-embeddings. We attribute this improvement to BERT's ability to capture semantic information, and to its robustness to text errors.

2.3.2. Phone sequence encoding

Following [14], we utilize the target utterance's phone sequence encoding generated by the pre-trained Tacotron2 Encoder module (Fig. 1, module C) from the phonetic sequence, to enrich the semantic word representation with its phonetic counterpart. We push the phone encoding sequence into a stack of three 32-dim Bi-LSTMs, and pool to word resolution by either averaging the output along word segments, or by taking the first output element corresponding to each word. The resulting word-level vectors are concatenated to the word-embedding produced by BERT. Note that the phone-sequence processing architecture resembles that of the baseline HPC prediction, as described in Section 2.1

2.3.3. Prosodic context encoding

We hypothesized that certain prosodic information extracted from the dialog history can help to attain better conversational prosody modeling. We extract the HPC sequence of the dialog context audio and feed it into additional stack of three 32-dim Bi-LSTMs. We average the whole output sequence into a single feature vector. This global feature vector is broadcast and concatenated to the target utterance word-vectors. We further apply a 128-dim linear layer followed by GELU [17] activation to each one of those word vectors, and output a 128-dim word embedding.

2.3.4. HPC decoder

The sentence-level, word-level and silence HPC features are decoded independently by three decoder modules, implemented as single linear layers (see Fig. 3). The decoders' outcomes are combined together to comprise the output HPC sequence. The sentence decoder is fed with sentence embeddings, where each sentence embedding is obtained by averaging over its corresponding word embeddings. Once a silence word needs to be inserted after a certain word $w1$, its embedding $we1$ (see Fig. 3) serves also as the embedding for the silence word and is fed to Silence Decoder (see Fig. 3) to obtain the silence HPC component (i.e. a silence duration estimate).

3. Conversational HPC Training

3.1. Data Preparation

The conversational HPC prediction model, excluding the pre-trained frozen parts of BERT and pre-trained S2S Front-End Encoder (see Fig. 2), is trained on Switchboard dataset that contains spontaneous conversations in a weakly controlled setup. As such, its data is challenging and noisy. Besides the speech, various paralinguistic and non-speech events occasionally happen, such as phone call quality distortions, external noise, coughing, laughter, stuttering, self-correction, un-

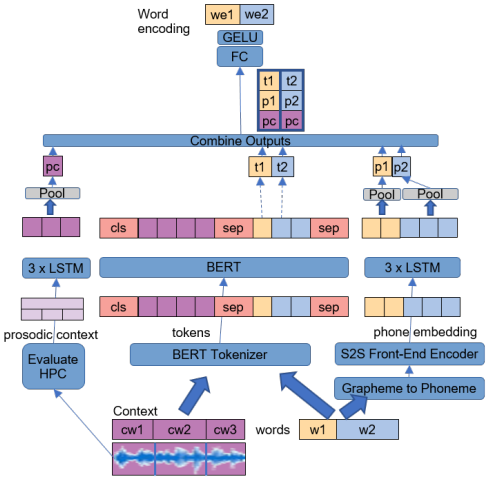


Figure 2: *Conversational HPC Predictor: Encoder. Outputs context-aware word-embedding, based on the word semantics and phonetics*

finished words, prolonged pauses, word and phrase fillers and para-linguistic particles. While the transcription is organized in turns, speakers can actually speak simultaneously. Inevitably, the transcribed text suffers from grammatical errors, incomplete sentences and weird phrasing. Fortunately, most of non-linguistic and para-linguistic events are consistently labeled in the manual transcription (with special signs). After removing all the special transcription words, we generated proper English utterances, inserted commas at each pause and obtained the phonetic sequences with word boundaries from each textual utterance, as described in Section 2.1. Then we compared the word counts in the *Switchboard* word-alignment files with the number of words in our generated phonetic sequence and left out all misaligned utterances. For each speaker turn we demanded to have a valid previous speaker turn, to be able to extract prosodic context by means of HPC. Eventually we retained $\sim$170K utterances for training, that served us to extract their phonetic sequences, target HPC sequences and prosodic (HPC-based) dialog context. We also held out a development set of 35 utterances for post-training adjustments (see Section 3.3) and a test set of 25 utterances for a listening test evaluation (see Section 4).

3.2. Dialog Context

One of the questions we wanted to explore was whether the dialog context is important for inferring the conversational style from the multi-speaker spontaneous conversational speech dataset (i.e. *Switchboard*).

3.2.1. Textual context

For the textual context (extracted from the past), we explored several context configurations:

- *NONE*: no context is used.
- *Last Turn (LT)*: The last turn and the target utterance are fed to BERT as *A*- and *B*-sequence correspondingly.

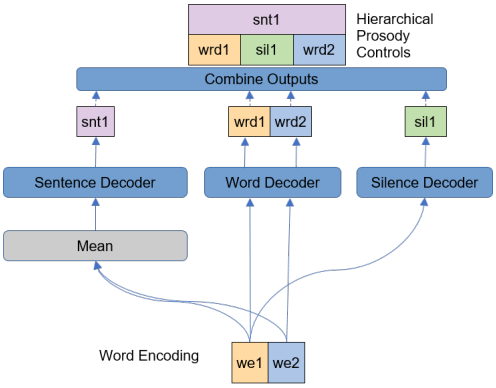


Figure 3: *Conversational HPC Predictor: Decoder*

- *Other Speaker Last Turn (OSLT)*: The context window goes back until the last turn of the other speaker. The text of the context turns is concatenated with period signs between turns, and is fed to BERT *sentence A* as in the LT case.
- *Other Speaker First Turn (OSFT)*: The context window is extended until covering the whole last burst of turns the other speaker said.

For training, we left the original text and punctuation as is, including the repeated words and the paralinguistics. The special event symbols were dropped.

3.2.2. Prosodic context

We concatenate the HPC sequences of the interlocutor's last burst of turns and set it as prosodic context. For the starting turn of each conversation we set the prosodic context to empty sequence. Examples whose prosodic context was missing due to pre-processing errors were discarded.

3.3. Training Procedure

We train our architecture with *MSE* regression loss, as follows:

$$\mathcal{L} = MSE_{word} + c_1 MSE_{sent} + c_2 MSE_{sil}, \qquad (3)$$

where the loss weights c_1, c_2 serve as hyper-parameters tuned to minimize *global MSE loss* of HPC sequence, as used in the baseline S2S TTS system, trained with the high quality speech corpora (see section 2.1).

Since BERT overfits small datasets very quickly, we freeze BERT in the first 5 training epochs while the rest of the architecture starts training. Then, we fine-tune the last two layers of BERT's encoder for 3 consequent epochs, and freeze BERT again for the rest of the training.

In addition, we used bucket batch sampler to equalize the LSTM's input sequence lengths, and let PyTorch-Lightning [18] automate the training process.

We used W&B sweeping tool [19] to apply massive Bayesian hyper parameter optimization. The optimized hyper parameters included the dialog context configuration, the sentence and silence loss coefficients, BERT fine-tuning parameters and others.

For each system we performed a separate hyper-parameter search to select several systems with the N-best losses. Among them we selected the best system by listening to 35 utterances of the held out development set.

We further post-process the predicted HPC features by adding a negative offset of -0.5 to the silence word duration HPC component to avoid too long silences that are common in the spontaneous speech, but disruptive when synthesized with neutral S2S TTS. This post-processing was also tuned on the held out development set. A list of the hyper parameters that was selected for each model under test is listed in Table 1. All the tested systems were trained with batch size of 128 and LSTM dropout of 0.3.

4. Evaluation

The training material for the pre-trained neutral multi-speaker S2S TTS system comprised corpora from three professional native speakers of US English, two females and one male, of 10-17K sentences each. A single female speaker was selected for the conversational evaluation, as it was found more suitable for the conversational speaking style than the others.

Several proposed variants of HPC models were trained separately on *Switchboard* to generate the HPC sequences from the input texts, with or without dialog contexts. Other parts of S2S TTS, besides the HPC model, where pre-trained and kept identical in all the systems.

In a subjective evaluation presented below we would like to assess how well the proposed conversational prosody models help to apply the conversational speaking style, while preserving a decent quality and naturalness of the synthesized speech. To that end, we consider the following systems:

1. **Base**: The baseline neutral S2S TTS system with the default HPC prediction, as learnt from the neutral voice corpora.
2. **Phn**: The neutral S2S TTS with the conversational HPC model, predicting HPC from phonetic sequence only (phones, lexical stress, phrase type, word boundaries).
3. **BERT-TC**: The neutral S2S TTS with the conversational HPC model based on a task-adjusted BERT, predicting HPC from textual input, enriched with textual dialog context.
4. **BERT-Phn**: The neutral S2S TTS with the conversational HPC model to predict HPC, based on phonetic and textual input, no dialog context is fed into BERT.
5. **BERT-Phn-TC**: The neutral S2S TTS with the conversational HPC model that combines that of item 2 and task-adjusted BERT, trained also with the textual dialog context. It predicts HPC, based on phonetic and textual input, enriched with textual dialog context.
6. **BERT-Phn-TPC**: The neutral S2S TTS with the conversational HPC model that combines that of item 2, but is conditioned also on the HPC dialog context, plus a task-adjusted BERT, trained with the input text and the textual dialog context. It predicts HPC, based on phonetic and textual input, enriched with both textual and prosodic dialog contexts.

To evaluate the systems defined above, we conducted a combined Mean Opinion Score (MOS) listening test for the six systems. No natural recordings were included in MOS tests, since no matched conversational utterances existed for the high quality voice. The test examples were taken from *Switchboard* with their original context.

We conducted a crowd-based evaluation (139 subjects) for a held-out test set of 25 sentences. The subjects were asked to rate 1) the overall quality and naturalness of an utterance and 2) *"how well the sound of the voice and the intonation convey the expressive character of a sentence in the context of the provided conversation"*. The subjects chose between five categorical answers (*1 - Poor, 2 - Bad, 3 - Fair, 4 - Good, 5 - Excellent*). The corresponding dialog context transcriptions were provided to the subjects so that they could assess how well the speaking style corresponds to the dialog context[1]. Each stimulus received 35 independent ratings. The raw ratings were subject to an outlier-removal procedure, after which each stimulus retained 31 independent votes on average.

Overall MOS results for 1) quality and naturalness and 2) conversational speaking style correspondence are provided in Table 2. As we requested a relatively large amount of independent votes per stimuli, it makes sense to present also a percentage of stimuli with higher than the **Base** model MOS scores, for each one of the five models under test (2-6).

5. Discussion and Conclusions

Overall MOS results have shown that the phone-only prosody prediction (**Phn**) fails to learn convincing speaking style, but rather significantly ($p < 0.01$) deteriorates the quality of the resultant speech. On the other hand, when considering the word semantics (using BERT), certain success in learning conversational style pattern from the noisy data is achieved. We observe that although the absolute MOS improvements for any of the BERT-containing models vs. the baseline in both the conversational style and the overall quality metrics are subtle (not statistically significant), the count of the better-scored stimuli is much higher for BERT (**BERT-TC**) model (68% for the conversational style and 64% for the overall quality metrics, correspondingly). The results revealed also that neither phonetic sequence, not prosodic dialog context contributed to better performance of the BERT-based HPC model (**BERT-TC**), probably due to the challenging spontaneous dataset used for training.

When exploring the best scored model's stimuli (**BERT-TC** vs. **Base**) and their corresponding MOS scores, we came to a conclusion that the perceived improvement came to some extent as a result of improving general expressiveness, but mostly due to better realization of common conversational speech patterns, such as filler words and phrases (e.g., "you know", "like", "well", etc.), that sound more fluent and natural in the proposed system. This observation aligns well with the fact that the conversational HPC models were learnt on a multi-speaker data set containing many speakers, conveying their own interpretations of a conversational speaking style, so just the most general conversational speech features could be acquired, as opposed to the previously reported setup where a large single speaker conversational data set is available [6]. This observation implies that most of the improvements came up at particular textual patterns, thus explaining why the word semantics (text-only input) seemed to be enough to gain those improvements.

Analyzing closely how the system scores change when adding the phonetic stream to the HPC predictor (e.g. **BERT-TC** vs. **BERT-Phn-TC**), we noted that more stimuli got worse perceptual prosody scores, due to some expressiveness deterio-

[1]Audio samples are available at http://ibm.biz/S2S-ConvStyle-SSW21.

System	Phn	BERT-TC	BERT-Phn-TC	BERT-Phn-TPC	BERT-Phn
bert context	-	LT	LT	OSLT	NONE
learning rate	2.892E-05	0.0002	0.0002	0.0002567	0.0002
max epochs	50	40	40	40	40
phone lstm pooling	starting phn	-	mean	mean	mean
sentence loss coef	11.929	13	13	13.002	13
silence loss coef	6.83	1	1	0.92	1
weight decay	0.0001067	0.0001	0.0001	0.0001218	0.0001

Table 1: *Hyper-parameters selected for each model.*

Table 2: *Mean opinion scores with 95% confidence interval (and percentage of stimuli with higher than **Base** MOS score) for the speaking style (Stl.) and overall quality and naturalness (Qual.)*

Cat.	Systems					
	Base	Phn	BERT-TC	BERT-Phn	BERT-Phn-TC	BERT-Phn-TPC
Stl.	3.81± 0.07 (-)	3.74± 0.07 (36%)	3.86± 0.06 (**68%**)	3.85± 0.06 (48%)	3.83± 0.06 (40%)	3.81± 0.07 (56%)
Qual.	3.82± 0.07 (-)	3.74± 0.07 (36%)	3.87± 0.06 (**64%**)	3.86± 0.07 (36%)	3.88± 0.06 (52%)	3.82± 0.07 (48%)

ration, thus obtaining lower count of stimuli with higher-than-baseline MOS scores. However, the terminal sentence prosody is improved, resulting in statistically similar overall MOS scoring.

Additional conversation style pattern, that is common in the spontaneous speech and acquired by the HPC model is *uptalk* [20], i.e. rising intonation on declarative statement end. However, we observed that our S2S TTS system (originally trained with neutral speech that had no uptalk examples) produced less convincing realizations of the uptalk pattern, that were consistently down-voted in the subjective evaluation.

Based on that findings, we are currently exploring gradual neutral and conversation HPC trajectory merging towards sentence ends, to eliminate the negative uptalk effect, while retaining other learnt conversational style effects.

6. References

[1] Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio *et al.*, "Tacotron: Towards end-to-end speech synthesis," *Proc. Interspeech 2017*, pp. 4006–4010, 2017.

[2] J. Shen, R. R. Pang, R. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerry-Ryan, R. Saurous, Y. Agiomyrgiannakis, and Y. Wu, "Natural TTS synthesis by conditioning wavenet on MEL spectrogram predictions," in *Proc. ICASSP*, Calgary, Canada, 2018, pp. 4779–4783.

[3] R. Skerry-Ryan, E. Battenberg, X. Y., Y. Wang, D. Stanton, J. Shor, R. Weiss, R. Clark, and R. Saurous, "Towards end-to-end prosody transfer for expressive speech synthesis with Tacotron," *CoRR*, vol. abs/1803.09047, 2018. [Online]. Available: http://arxiv.org/abs/1803.09047

[4] R. Valle, J. Li, and B. Catanzaro, "Mellotron: Multispeaker expressive voice synthesis by conditioning on rhythm, pitch and global style tokens," in *Proc. ICASSP*, Barcelona, Spain, 2020, pp. 6189–6193.

[5] P. Wu, Z. Ling, L. Liu, Y. Jiang, H. Wu, and L. Dai, "End-to-end emotional speech synthesis using style tokens and semi-supervised training," in *Proc. APSIPA*, Lanzhou, China, 2019, pp. 623–627.

[6] H. Guo, S. Zhang, F. K. Soong, L. He, and L. Xie, "Conversational end-to-end tts for voice agents," in *2021 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2021, pp. 403–409.

[7] J. J. Godfrey, E. C. Holliman, and J. McDaniel, "Switchboard: Telephone speech corpus for research and development," in *Acoustics, Speech, and Signal Processing, IEEE International Conference on*, vol. 1. IEEE Computer Society, 1992, pp. 517–520.

[8] S. Shechtman, R. Fernandez, and D. Haws, "Supervised and unsupervised approaches for controlling narrow lexical focus in sequence-to-sequence speech synthesis," in *2021 IEEE Spoken Language Technology Workshop (SLT)*, Shenzhen, China, January 2021, pp. 431–437.

[9] F. A. Gers, J. Schmidhuber, and F. Cummings, "Learning to forget: Continual prediction with LSTM," *Neural Computaiton*, vol. 12, no. 10, pp. 2451–2471, 2000.

[10] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *NAACL-HLT*. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 4171–4186.

[11] J. M. Valin and J. Skoglund, "LPCNET: Improving neural speech synthesis through linear prediction," in *ICASSP*, Brighton, England, 2019, pp. 5891–5895.

[12] S. Shechtman, R. Rabinovitz, A. Sorin, Z. Kons, and R. Hoory, "Controllable sequence-to-sequence neural TTS with LPCNET backend for real-time speech synthesis on CPU," *CoRR*, 2020. [Online]. Available: http://arxiv.org/abs/2002.10708

[13] J. Pitrelli, R. Bakis, E. Eide, R. Fernandez, W. Hamza, and M. Picheny, "The IBM Expressive Text-to-Speech Synthesis System for American English." *IEEE Trans. Audio, Speech and Lang. Processing*, vol. 14, no. 4, pp. 1099–1108, July 2006.

[14] S. Shechtman and A. Sorin, "Sequence to Sequence Neural Speech Synthesis with Prosody Modification Capabilities," in *Proc. SSW10*, Vienna, Austria, 2019, pp. 275–280.

[15] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, San Diego, May 2015. [Online]. Available: https://arxiv.org/abs/1412.6980

[16] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey *et al.*, "Google's neural machine translation system: Bridging the gap between human and machine translation," *arXiv preprint arXiv:1609.08144*, 2016.

[17] D. Hendrycks and K. Gimpel, "Gaussian error linear units (gelus)," *arXiv preprint arXiv:1606.08415*, 2016.

[18] e. a. Falcon, WA, "Pytorch lightning," *GitHub. Note: https://github.com/PyTorchLightning/pytorch-lightning*, vol. 3, 2019.

[19] L. Biewald, "Experiment tracking with weights and biases," 2020, software available from wandb.com. [Online]. Available: https://www.wandb.com/

[20] P. Warren, *Uptalk: The phenomenon of rising intonation*. Cambridge University Press, 2016.

11th ISCA Speech Synthesis Workshop (SSW 11)
26-28 August 2021, Budapest, Hungary

EmoCat: Language-agnostic Emotional Voice Conversion

Bastian Schnell[♯‡], Goeric Huybrechts[†], Bartek Perz[†], Thomas Drugman[†], Jaime Lorenzo-Trueba[†]*

[♯] Idiap Research Institute, Martigny, Switzerland
[‡]Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland
[†] Amazon, TTS Research, Cambridge, United Kingdom

bastian.schnell@idiap.ch, {huybrech,perzbart,drugman,truebaj}@amazon.com

Abstract

Emotional voice conversion models adapt the emotion in speech without changing the speaker identity or linguistic content. They are less data hungry than text-to-speech models and allow to generate large amounts of emotional data for downstream tasks. In this work we propose EmoCat, a language-agnostic emotional voice conversion model. It achieves high-quality emotion conversion in German with less than 45 minutes of German emotional recordings by exploiting large amounts of emotional data in US English. EmoCat is an encoder-decoder model based on CopyCat, a voice conversion system which transfers prosody. We use adversarial training to remove emotion leakage from the encoder to the decoder. The adversarial training is improved by a novel contribution to gradient reversal to truly reverse gradients. This allows to remove only the leaking information and to converge to better optima with higher conversion performance. Evaluations show that Emocat can convert to different emotions but misses on emotion intensity compared to the recordings, especially for very expressive emotions. EmoCat is able to achieve audio quality on par with the recordings for five out of six tested emotion intensities.[1]

Index Terms: Voice Conversion, Emotional Speech, Speech Synthesis, Expressive TTS, Text-to-Speech

1. Introduction

Neural Text-to-Speech (TTS) has greatly supported the advent of artificial voice assistants like Amazon Alexa, Google Assistant, or Siri. These systems are trained on tens of hours of data [1] and produce high-quality speech with close to perfect intelligibility [2]. However, their speech is mostly neutral, which prevents natural conversations and closer bounds with the user. Creating voices in more expressive speaking styles usually requires recording similarly large amounts of speech for the desired style. This is very time-consuming and costly. An alternative is the generation of synthetic data to satisfy the high data needs. The conversion of speech is generally assumed to be easier than TTS, thus has lower data needs.

Emotional voice conversion (EVC) is a subfield of voice conversion (VC) which studies the transformation of a source audio signal into a different emotion while maintaining its linguistic content and speaker identity. Techniques applied in EVC are similar to VC and differ mostly in their feature selection [3, 4]. EVC techniques working without hand-crafted features are applicable to other speaking styles as well. EVC is also applied to other tasks like film dubbing.

In this work, we aim to convert neutral to emotional speech in German. As we have only a limited amount of emotional German data available, we exploit emotional recordings in US English. We propose EmoCat, a language-agnostic EVC model trained jointly on German and US English working directly on mel-spectrograms. Compared to other works we use mel-spectrograms to leverage our high-quality universal vocoder [5] to keep a high bar on segmental quality. Our model adapts the CopyCat model [6] (which is based on AutoVC [7]) for intra-speaker emotion conversion. CopyCat is a VC model which allows to convert the speech of unseen speakers to a set of target speakers. In contrast to the global speaker identity, emotion is a continuous component of speech. We use adversarial training to explicitly remove emotion leakage from the encoder, which encodes the neutral source spectrogram, to the decoder, which generates the converted emotional spectrogram. We propose a novel improvement to gradient reversal [8] to stabilise its gradients. We further investigate fine-tuning to improve naturalness. In an ablation study, we assess the effectiveness of each of the techniques. The proposed model is able to convert neutral German to two different emotions in three intensities with the support of less than 45 minutes of German emotional data. To the best of our knowledge, no work exists on EVC with multilingual data or mel-spectrograms.

2. Related work

Emotional voice conversion methods are generally split into two categories: parallel and non-parallel training data.

In the **parallel data** scenario the database contains the same utterance spoken by the same speaker in the different target emotions. This allows the network to directly learn the conversion. However, these databases are rare and typically small. It is expensive to record all the emotions of an utterance for a large phoneme coverage. Additionally, it is challenging for voice talents to act the target emotion when the linguistic content of the utterance does not match. This can lead to errors in emotion intensity and thus either lowers the quality of the database or requires the exclusion of some recordings. Recent works, including [9, 10, 11, 12, 13, 14], are less relevant for this work.

In the **non-parallel data** scenario, the utterances for each emotion differ, meaning that the content can better match the emotion. This allows a wider variety of utterances and also simplifies acting for the voice talents. With the lack of parallel data, a model cannot be trained to do the conversion directly as the ground truth target is not available. The training can only be guided in an unsupervised way. Generative adversarial networks (GAN) and cycle consistency losses are commonly used techniques here [15, 4, 16].

* Work performed while being an intern at Amazon.

[1] The submission platform does not allow to attach samples; they will be released publicly as part of an associated blog post.

10.21437/SSW.2021-13

In [15] an encoder-decoder structure with a content and style encoder is used to convert mel-cepstrum (MCEP) extracted by WORLD [17] . The model is trained with three losses. First, the cepstrogram is auto-encoded and an L1 reconstruction loss applied. Second, a semi-cycle consistency L1 loss forces the encoder embeddings to match before and after conversion. Third, a GAN loss tries to discriminate generated from recorded samples. F_0 is converted by a linear transform to match the statistics of the target emotion domain. The band aperiodicities remain unchanged.

StarGAN is used in [4] on WORLD features with a reconstruction loss, an L1 cycle consistency loss, and a real/fake GAN loss. The model architecture is the same as in the original StarGAN-VC paper [3]. An emotion recognition model (a variant of [18]) was trained with the generated samples and evaluations show that its accuracy improved.

CycleGAN has also been used for emotion conversion [16]. It is trained with three losses: 1) a reconstruction loss, 2) a cycle-consistency loss on a sample converted to another emotion and then back to the source emotion, and 3) the GAN loss for real/fake discrimination. The experiments show that separate CycleGANs for F0 and MCEP outperform a joint model. [19] follows a very similar approach but uses an additional emotion classification loss and no reconstruction loss.

A different approach is the variational auto-encoding Wasserstein GAN (VAW-GAN) for emotion conversion [20] (originally proposed for VC in [21]). It consists of a variational auto-encoder (VAE) structure where the decoder is conditioned on an emotion embedding. The latent dimension is chosen to be small enough so that it will not contain emotion information. The model is trained with reconstruction loss, standard Kullback-Leibler (KL) divergence on the VAE latent space, and a Wasserstein GAN loss. Instead of using a binary cross-entropy loss for the real/fake prediction of the discriminator, the Wasserstein distance is used. The authors in [21] claim that the Wasserstein distance is better suited for VC as it is computed from the optimal transport corresponding to best frame alignment.

Our approach is closest to the VAW-GAN in [20] as it employs a similar encoder-decoder structure with a VAE encoder. However, the bottleneck used is temporal and drastically smaller, also we condition the decoder on the linguistic content. Our reference encoder is similar to the one in [22]. In contrast to all related work above, we operate on mel-spectrograms and train with multi-lingual data.

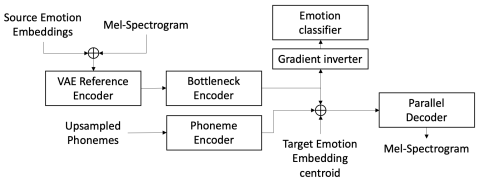


Figure 1: *Structure of the encoder-decoder EmoCat model with a gradient inverter block followed by an emotion classifier to remove emotion information in the bottleneck embeddings. The plus sign denotes a concatenation.*

3. Model description

In this section we introduce EmoCat, a language-agnostic intra-speaker emotion conversion model. It aims to convert neutral speech to emotional speech of the same speaker[2]. EmoCat is based on CopyCat [6] and inherits the same structure and hyperparameters except four differences:

1. It uses 64-dim emotion embeddings instead of 128-dim speaker embeddings (see Section 3.1).
2. It uses a gradient inverter block to remove emotion leakage from the bottleneck embeddings (see Section 3.2).
3. It operates on multi-lingual data (see Section 4.1).
4. It does not pass the phoneme embeddings to the VAE reference encoder.

Figure 1 shows the network structure. The VAE reference encoder encodes the mel-spectrograms and its emotion embedding. A dimensional and temporal bottleneck is applied by only selecting every N-th frame [7]. Each selected frame is copied N times (to restore the sequence length). The bottleneck embeddings should contain as much information as possible to generate high-quality speech but no emotion information. This is ensured by passing them through the gradient inverter to the emotion classifier, which removes any leaking emotion information. Force-aligned upsampled phonemes (procedure described in [23]) are encoded by the phoneme encoder to produce phoneme embeddings. During inference, the bottleneck and phoneme embeddings are stacked with the target-emotion embedding centroid and consumed by a parallel decoder to produce the converted mel-spectrograms. During training, the oracle utterance-level emotion embedding is used on the encoder and decoder side. Source and target spectrograms are the same as well. The parallel decoder consists of a stack of three convolutional layers followed by a uni-directional long short-term memory (LSTM). The model is trained with an L1 reconstruction loss and the KL-loss on the VAE latent space. For the detailed architecture, please refer to the CopyCat paper [6].

3.1. Utterance-level emotion embeddings

During training, utterance-level emotion embeddings are fed to the VAE reference encoder and the parallel decoder. The emotion embeddings need to be organised language-independently by their style and other latent information to be beneficial to the model. This excludes simple embeddings per emotion class and suggests a learnable approach.

We obtain the emotion embeddings from a separate TTS model, which is pre-trained to do phoneme to mel-spectrogram conversion. The TTS model has a Tacotron-like architecture [24] with the addition of two VAE reference encoders [25]. One reference encoder captures the speaker information while the other captures the emotion. We use intercross training [26] to guide each encoder to encode only the speaker/emotion information and to be language-independent. Within the reference encoder the last GRU state is projected to form the VAE parameters. The embedding is obtained by sampling from the VAE. We use the predicted embeddings from the emotion reference encoder as utterance-level emotion embeddings for the EmoCat training. We could learn the emotion embeddings in a similar fashion on-the-fly within the EmoCat model, but this would increase its training time, which is not desirable during research.

[2]We have informally verified that it also allows conversion between emotions, but this lies out of the main scope of this paper.

We could also obtain them from a simple emotion recognition model, but we hypothesised that those embeddings might be more suited for recognition than generation.

For the CopyCat model, robust speaker embeddings from a pre-trained speaker identification system are necessary, because the model also has to convert from unseen speakers. This is not the case for the EmoCat model, which only converts between seen emotions. Thus it requires less sophisticated emotion embeddings.

During inference, the utterance-level emotion embedding of the converted spectrograms is unknown. Instead we compute the centroid for each emotion over all emotion embeddings extracted from the training set and feed it to the decoder. The VAE reference encoder still uses the utterance-level emotion embedding of the input audio.

3.2. Gradient inverter

As emotion is a continuous and integral part of speech, it is necessary to explicitly prevent it from leaking from the encoder to the decoder side. With a pre-trained EmoCat with frozen weights we trained independent gated recurrent unit (GRU) emotion classifiers to predict the source emotion from the bottleneck embeddings, where the best achieved 64% overall accuracy. We found that heavy leakage resulted in low emotion intensity during conversion. Decreasing the bottleneck (as described in AutoVC [7]) led to heavy degradation in signal quality and intelligibility. With the reconstruction loss alone, we could not force the bottleneck embeddings to remove the undesired emotion information while keeping information needed for high signal quality.

Instead we used a gradient reversal block before the emotion classifier during training to actively remove emotion leakage from the bottleneck embeddings. The idea of gradient reversal is to reverse the gradients during back-propagation to remove any activation in the input that helps the following classifier. Gradient reversal achieves this by swapping the sign of the gradient Δ (Equation 1). It also applies a weight λ to control the impact of the gradient on the preceding layers. The choice of the weight greatly influences the performance of the final model.

$$\Delta' = -\lambda\Delta \tag{1}$$

We experimented with a feed-forward and a GRU based emotion classifier. Interestingly, EmoCat converged to a better model in terms of conversion ability with the feed-forward classifier than the GRU one. This suggests that with gradient reversal even a weak classifier gives sufficient gradients to lead to a better convergence point.

We again trained the same emotion classifier as above on the bottleneck embeddings of the model with gradient reversal. The classifier mainly predicted the majority class (95% of the time) showing that the majority of the emotion leakage was removed. Informal listening verified that the conversion ability of the model improved.

We argue that a simple swap of the sign (Equation 1) fulfils only half of the reversal purpose. Consider the following two scenarios:

1. Imagine there is **no leakage** in the input. As the classifier cannot rely on any information in the input, its prediction is random and the cross-entropy loss on its predictions is high. Thus the back-propagated gradients are large as well. Even though there is no leakage the preceding network receives a large reversed gradient.
2. Imagine there is **significant leakage** in the input and the classifier is already properly trained. Then its prediction is good, the cross-entropy loss is low, and the back-propagated gradients are small. Even though there is significant leakage the preceding network receives only a small reversed gradient.

The desired effect on the preceding network in both scenarios should be swapped. Without any leakage the received gradients should be small, while with significant leakage the gradients should be large.

To address this issue, we present the **gradient inverter** block. Instead of only swapping the sign of the gradient, it performs a proper inversion by also converting small gradients to large ones and vice versa. We have experimented with two gradient inverter functions.

$$\Delta' = \frac{-\lambda\Delta}{||\Delta||_2^2} \qquad \text{Inverse square norm} \tag{2}$$

$$\Delta' = \frac{-\lambda\Delta}{\exp||\Delta||_2^2} \qquad \text{Inverse exp square norm} \tag{3}$$

Equation 2 implements directly what we want to achieve by scaling the gradient by its squared norm. Gradients with a norm smaller than one will become greater than one and vice versa. However, it might lead to unstable behaviour as gradients with a norm close to zero are scaled towards infinity. Equations 3 prevents this by bounding the denominator to less than one. In this variant, gradients with a small norm remain almost unchanged while big gradients are quickly faded out. We found that depending on the target emotion one of the proposed inverter functions performs better.

3.3. Fine-tuning

While the EmoCat model with the proposed gradient inverter achieved high emotion intensities, its signal quality left room for improvement. We investigated fine-tuning on a subset of the training data. First the model was trained with all data until convergence. Then we continued training on emotional and similar amounts of neutral data. This should compensate the averaging effect in the decoder introduced by the huge amount of neutral training data. We did not change any hyper-parameters, learning rates, or losses compared to the first training step. This approach outperforms a GAN-like loss (same as used for CopyCat [6]), which strives for the generated spectrogram to be indistinguishable from the recordings.

4. Experiments

We aim at generating emotional German samples by converting from neutral using a model trained with a limited amount of emotional German data. We focused on two emotions: excited and disappointed, in three intensities: low, medium, high.

4.1. Database

We use two internal databases. For German, we use more than 20 h of neutral and 45 min of emotional single-speaker recordings of a female voice. 20 neutral samples are set aside as test set. We do not use a development set to guide the training because the L1 reconstruction loss does not match human perception. The 45 min of emotional data are split equally into excited and disappointed. 25% is low, 50% medium, and 25% high intensity. Excluding the test set, we have around 5 min

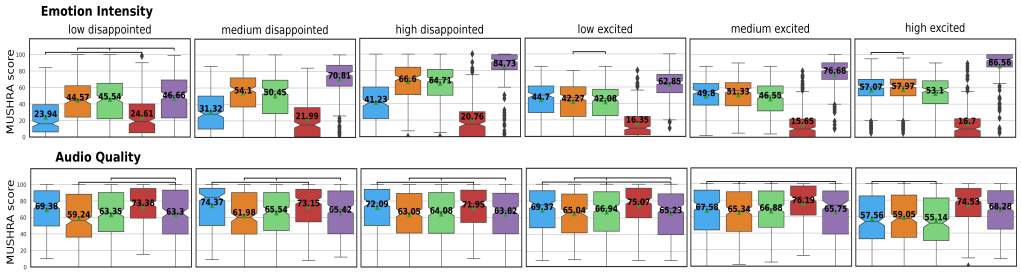


Figure 2: *System descriptions: blue: grad. reversal, orange: grad. inverter, light green: grad. inverter fine-tuned, red: neutral baseline, purple: recordings. Black horizontal bars connecting systems denote no statistically significant difference between them (p-value* < 0.05*).*

for the most challenging intensity: high. As we do not have access to more emotional German data, we use recordings of a female US English voice as supporting speaker. From this speaker, we use more than 20 h of neutral and more than 10 h of emotional recordings of the same emotion categories. We found that including US English data greatly improved the conversion abilities of our model, despite the differences in language. This suggests that the production of emotion follows a similar behaviour in English and German, which thus makes it beneficial to include the English data during training. This might hold true for other emotions as well. 24 kHz recordings are used. We trim all silences to be maximum 100 ms and extract 80-dim mel-spectrogram. We use phonemes with fully disjoint sets for English and German, thus the speaker identity can directly be inferred and explicit speaker embeddings are unnecessary.

4.2. Models

We conduct an ablation study across three models. Each is trained for 100k steps on the combined two databases. The mel-spectrogram is synthesised with our universal vocoder [5].

1. **Grad. reversal** - This model uses the vanilla gradient reversal block (Equation 1) to remove leaking emotion information. In contrast to the following two models, we used a weighted cross entropy loss for the adversarial emotion classifier to compensate for the huge class imbalance in the training data. We chose the weights inverse proportional to the amount of the emotion in the total training data. We found that this improved the grad. reversal model.
2. **Grad. inverter** - This model replaces the gradient reversal block of model 1 with the improved gradient inverter block (Section 3.2). We use two separate models for the conversion. The model to convert to the excited emotions uses the inverse exp square norm function (Equation 3), while the one to convert to disappointed uses inverse square norm (Equation 2). This was selected based on a clear performance difference in informal listening.
3. **Fine-tuning** - This is model 2 fine-tuned for 2k steps as described in Section 3.3. The best results were obtained by fine-tuning on the emotional data of the target speaker with a similar amount of neutral data as for each emotion. The neutral data requirement is probably due to the adversarial training. This simple fine-tuning outperforms GAN fine-tuning.

We wanted to include a state-of-the-art baseline, however we did not find any work on emotion conversion from spectrograms, which is required to use our high-quality neural vocoder. We adapted the work of [4] based on their StarGAN implementation[3] to use mel-spectrograms instead of WORLD vocoder features, but the quality of the synthesized speech was very low. It is likely that major adaptations to the model architecture are necessary to achieve competitive results. However, creating such a baseline system is out of scope for this work. A comparison with a WORLD vocoder-based model is superfluous [27], therefore it was impossible for us to include a competitive state-of-the-art baseline model in our benchmark.

4.3. Evaluations

We randomly selected 10 neutral German samples from the held-out test set and converted them to each of the six emotion intensities. 24 native German listeners rated the samples in terms of emotion intensity and audio quality in a MUSHRA [28] test from 0 to 100.

4.3.1. Emotion intensity

We asked listeners to rate the emotion intensity where we provided another neutral recording (different sentence) as a reference of 0. We also included another recoding of the same emotion of the target speaker as an upper anchor and the utterance generated by a neutral baseline system. We see in Figure 2 top line that our gradient inverter model outperforms vanilla gradient reversal for medium excited and is similar in high excited (no statistical difference, two-tailed t-test with *p-value* < 0.05, denoted as a horizontal bar in the plots) while it is significantly worse for low excited. The exp square norm function (Equation 3) only scales large gradients down which does not seem to be optimal for the excited intensities. For disappointed the gradient inverter model achieves more than 20 MUSHRA points higher score across all intensities, proving the improvement through the gradient inverter function. We either did not yet find a gradient inverter function which generalises to different emotions, or the function should be chosen depending on the use case. Fine-tuning lowers the emotion intensity for the medium and high emotions. This shows an averaging effect of the neutral and low intensity data. It should also be noted that we see a clear ascent from the low to the high intensity, but do not yet reach the emotion intensity of the recordings except for low dis-

[3] https://github.com/glam-imperial/EmotionalConversionStarGAN

appointed. We were only able to partially address the averaging effect in the decoder, which might reveal a general shortcoming of current decoder architectures. Highly expressive data in another language seems to improve the system only to a certain point. More high expressive German recordings, even from other speakers, might push the emotion intensity further.

4.3.2. Audio quality

We compared the same systems as above but without a reference sample and asked the listeners to rate the audio quality (Figure 2 bottom line). We do not see a statistical difference between all systems for medium and high excited. Vanilla gradient reversal outperforms both other techniques for low excited and all disappointed intensities, but at a lower emotion intensity which makes the comparison unfair. The other techniques are still at par with the recordings. We see a trade-off between emotion intensity and audio quality here. We usually found that higher emotion intensities suffer from reduced signal quality. Most likely because low intensities are close to neutral samples for which we have a lot of training data. This leads back to the averaging effect in the decoder. We suggest to explore different decoder architectures more suitable for highly expressive speaking styles. While we are not able to reach the emotion intensity of the recordings yet, we achieve high audio quality at a generally lower intensity level. Fine-tuning did not achieve the desired improvement in audio quality. Even though it increased the MUSHRA score in five out of six emotions the difference is only statistically significant for low disappointed. The increase in audio quality might be a consequence of the lower emotion intensity instead of fine-tuning. However, for low disappointed fine-tuning increased audio quality without reduced emotion intensity. Interestingly, listeners found the audio quality of the neutral baseline system to be significantly higher than the emotional recordings. Our current hypothesis is that the listeners indeed noticed that the recordings are acted emotions and thus found them slightly unnatural.

5. Conclusion

We proposed EmoCat, a novel EVC model based on CopyCat, which operates directly on mel-spectrograms. It allows to convert neutral to emotional samples in German with less than 45 minutes of German emotional recordings. It achieves this by leveraging large amounts of emotional English data with the same emotions. While we expect the technique to be language-agnostic, we only demonstrate it for the rather similar languages German and US English. Even though the model is able to generate expressive speech at different intensities, we are not yet matching the expressiveness of the recordings. Moreover, we presented the gradient inverter block, an improvement to gradient reversal. This showed statistical significant improvements in emotion intensity for four out of six emotions in subjective listening tests. We also found minor improvements in audio quality, at the cost of emotion intensity, through fine-tuning on the target emotional data. Future work is required to investigate the influence of increasing the amount of emotional German data, testing on more dissimilar languages, and further improvements to the gradient inverter functions.

6. References

[1] Vatsal Aggarwal, Marius Cotescu, Nishant Prateek, Jaime Lorenzo-Trueba, and Roberto Barra-Chicote, "Using VAEs and normalizing flows for one-shot text-to-speech synthesis of expressive speech," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6179–6183.

[2] Jonathan Shen, Ruoming Pang, Ron J Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, Rj Skerrv-Ryan, et al., "Natural TTS synthesis by conditioning wavenet on mel spectrogram predictions," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4779–4783.

[3] Hirokazu Kameoka, Takuhiro Kaneko, Kou Tanaka, and Nobukatsu Hojo, "StarGAN-VC: Non-parallel many-to-many voice conversion using star generative adversarial networks," in *2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2018, pp. 266–273.

[4] Georgios Rizos, Alice Baird, Max Elliott, and Björn Schuller, "StarGAN for emotional speech conversion: Validated by data augmentation of end-to-end emotion recognition," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 3502–3506.

[5] Jaime Lorenzo-Trueba, Thomas Drugman, Javier Latorre, Thomas Merritt, Bartosz Putrycz, Roberto Barra-Chicote, Alexis Moinet, and Vatsal Aggarwal, "Towards achieving robust universal neural vocoding," *Proc. Interspeech 2019*, pp. 181–185, 2019.

[6] Sri Karlapati, Alexis Moinet, Arnaud Joly, Viacheslav Klimkov, Daniel Sez-Trigueros, and Thomas Drugman, "CopyCat: Many-to-Many Fine-Grained Prosody Transfer for Neural Text-to-Speech," in *Proc. Interspeech 2020*, 2020, pp. 4387–4391.

[7] Kaizhi Qian, Yang Zhang, Shiyu Chang, Xuesong Yang, and Mark Hasegawa-Johnson, "AutoVC: Zero-shot voice style transfer with only autoencoder loss," Long Beach, California, USA, 09–15 Jun 2019, vol. 97 of *Proceedings of Machine Learning Research*, pp. 5210–5219, PMLR.

[8] Yaroslav Ganin and Victor Lempitsky, "Unsupervised domain adaptation by backpropagation," in *International conference on machine learning*. PMLR, 2015, pp. 1180–1189.

[9] Carl Robinson, Nicolas Obin, and Axel Roebel, "Sequence-to-sequence modelling of F0 for speech emotion conversion," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6830–6834.

[10] Ravi Shankar, Jacob Sager, and Archana Venkataraman, "A multi-speaker emotion morphing model using highway networks and maximum likelihood objective.," in *INTERSPEECH*, 2019, pp. 2848–2852.

[11] Ravi Shankar, Hsi-Wei Hsieh, Nicolas Charon, and Archana Venkataraman, "Automated emotion morphing in speech based on diffeomorphic curve registration and highway networks.," in *INTERSPEECH*, 2019, pp. 4499–4503.

[12] Huaiping Ming, Dongyan Huang, Lei Xie, Jie Wu, Minghui Dong, and Haizhou Li, "Deep bidirectional lstm modeling of timbre and prosody for emotional voice conversion," *Interspeech 2016*, pp. 2453–2457, 2016.

[13] Ryo Aihara, Ryoichi Takashima, Tetsuya Takiguchi, and Yasuo Ariki, "Gmm-based emotional voice conversion using spectrum and prosody features," *American Journal of Signal Processing*, vol. 2, no. 5, pp. 134–138, 2012.

[14] Zeynep Inanoglu and Steve Young, "A system for transforming the emotion in speech: Combining data-driven conversion techniques for prosody and voice quality," in *Eighth annual conference of the international speech communication association*, 2007.

[15] Jian Gao, Deep Chakraborty, Hamidou Tembine, and Olaitan Olaleye, "Nonparallel emotional speech conversion," *Proc. Interspeech 2019*, pp. 2858–2862, 2019.

[16] Kun Zhou, Berrak Sisman, and Haizhou Li, "Transforming spectrum and prosody for emotional voice conversion with non-parallel training data," in *Proc. Odyssey 2020 The Speaker and Language Recognition Workshop*, 2020, pp. 230–237.

[17] Masanori Morise, Fumiya Yokomori, and Kenji Ozawa, "WORLD: a vocoder-based high-quality speech synthesis system for real-time applications," *IEICE TRANSACTIONS on Information and Systems*, vol. 99, no. 7, pp. 1877–1884, 2016.

[18] Zixing Zhang, Bingwen Wu, and Björn Schuller, "Attention-augmented end-to-end multi-task learning for emotion prediction from speech," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6705–6709.

[19] Songxiang Liu, Yuewen Cao, and Helen Meng, "Emotional voice conversion with cycle-consistent adversarial network," *arXiv preprint arXiv:2004.03781*, 2020.

[20] Kun Zhou, Berrak Sisman, Mingyang Zhang, and Haizhou Li, "Converting anyone's emotion: Towards speaker-independent emotional voice conversion," *arXiv preprint arXiv:2005.07025*, 2020.

[21] Chin-Cheng Hsu, Hsin-Te Hwang, Yi-Chiao Wu, Yu Tsao, and Hsin-Min Wang, "Voice conversion from unaligned corpora using variational autoencoding Wasserstein generative adversarial networks," *arXiv preprint arXiv:1704.00849*, 2017.

[22] RJ Skerry-Ryan, Eric Battenberg, Ying Xiao, Yuxuan Wang, Daisy Stanton, Joel Shor, Ron Weiss, Rob Clark, and Rif A Saurous, "Towards end-to-end prosody transfer for expressive speech synthesis with tacotron," in *international conference on machine learning*. PMLR, 2018, pp. 4693–4702.

[23] Viacheslav Klimkov, Srikanth Ronanki, Jonas Rohnke, and Thomas Drugman, "Fine-grained robust prosody transfer for single-speaker neural text-to-speech," *Proc. Interspeech 2019*, pp. 4440–4444, 2019.

[24] Javier Latorre, Jakub Lachowicz, Jaime Lorenzo-Trueba, Thomas Merritt, Thomas Drugman, Srikanth Ronanki, and Viacheslav Klimkov, "Effect of data reduction on sequence-to-sequence neural TTS," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 7075–7079.

[25] Shubhi Tyagi, Marco Nicolis, Jonas Rohnke, Thomas Drugman, and Jaime Lorenzo-Trueba, "Dynamic prosody generation for speech synthesis using linguistics-driven acoustic embedding selection," *Proc. Interspeech 2020*, pp. 4407–4411, 2020.

[26] Yanyao Bian, Changbin Chen, Yongguo Kang, and Zhenglin Pan, "Multi-reference tacotron by intercross training for style disentangling, transfer and control in speech synthesis," *arXiv preprint arXiv:1904.02373*, 2019.

[27] Xin Wang, Jaime Lorenzo-Trueba, Shinji Takaki, Lauri Juvela, and Junichi Yamagishi, "A comparison of recent waveform generation and acoustic modeling methods for neural-network-based speech synthesis," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4804–4808.

[28] B Series, "Method for the subjective assessment of intermediate quality level of audio systems," *International Telecommunication Union Radiocommunication Assembly*, 2014.

11th ISCA Speech Synthesis Workshop (SSW 11)
26-28 August 2021, Budapest, Hungary

Enhancing audio quality for expressive Neural Text-to-Speech

Abdelhamid Ezzerg, Adam Gabrys, Bartosz Putrycz, Daniel Korzekwa, Daniel Sáez-Trigueros, David McHardy, Kamil Pokora, Jakub Lachowicz, Jaime Lorenzo-Trueba, Viacheslav Klimkov

Amazon Text-to-Speech Research

ezzerg@amazon.co.uk

Abstract

Artificial speech synthesis has made a great leap in terms of naturalness as recent Text-to-Speech (TTS) systems are capable of producing speech with similar quality to human recordings. However, not all speaking styles are easy to model: highly expressive voices are still challenging even to recent TTS architectures since there seems to be a trade-off between expressiveness in a generated audio and its signal quality. In this paper, we present a set of techniques that can be leveraged to enhance the signal quality of a highly-expressive voice without the use of additional data. The proposed techniques include: tuning the autoregressive loop's granularity during training; using Generative Adversarial Networks in acoustic modeling; and the use of Variational Auto-Encoders in both the acoustic model and the neural vocoder. We show that, when combined, these techniques greatly closed the gap in perceived naturalness between the baseline system and recordings by 39% in terms of MUSHRA scores for an expressive celebrity voice.

Index Terms: Neural Text-to-Speech, Generative Adversarial Networks, Variational Auto-Encoders.

1. Introduction

Artificial speech synthesis has seen a considerable change of paradigm: from the use of concatenative-based approaches [1, 2, 3], to leveraging modern Neural Text-to-Speech (NTTS) architectures such as Wavenet [4] and Tacotron [5]. Neural-based models are capable of synthesizing speech that rivals the real one in terms of quality while not being as constrained as concatenative methods in terms of phonetic coverage. Nonetheless, neural models are still data-hungry: training high-fidelity TTS systems using neural networks requires many hours of high-quality training data [6].

In addition to the challenge of gathering high-quality training data, we observed a tradeoff between the level of expressiveness in a voice (measured using the variance of f0, energy and phonemes' durations within the training data) and the segmental quality of produced speech: while standard architectures were able to produce high quality speech for neutral voices, their produced speech for highly-expressive voices suffered from degradations in audio quality.

In this paper, we present techniques that we applied, on top of a standard architecture such as in [5], to enhance the speech quality of highly-expressive voice. The described techniques are:

- Increasing the degree of autoregression as the training progresses
- The use of adversarial training for improving the quality of generated spectrograms
- The use of variational autoencoders (VAEs) with carefully selected latent representations at inference time

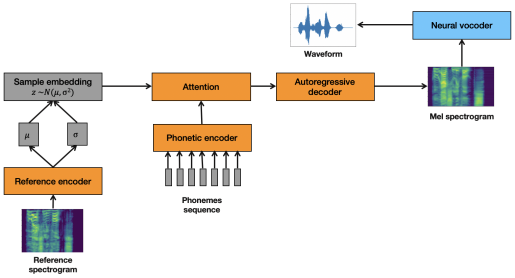


Figure 1: *Overview of model architecture. The system can be broken into two parts: an acoustic model and a neural vocoder that produces waveform. Orange blocks highlight the building neural network blocks for the acoustic model while the neural vocoder is represented by a blue box.*

- Training a neural vocoder conditioned on latent representations extracted using a pre-trained VAE

In Section 2 we will separately explain each of the applied techniques, Section 3 will present and discuss the result of applying the above-mentioned techniques on an expressive celebrity voice while Section 4 will be for conclusions.

2. Proposed approach

2.1. Model's architecture

The model we use comprises two main modules trained separately: an acoustic model which predicts a mel-spectrogram from an input sequence of phonemes, and a neural vocoder that predicts the waveform from the output of the acoustic model (see figure 1).

The acoustic model is a state-of-the-art sequence-to-sequence (seq2seq) neural network [5, 7, 6, 8] that leverages the attention mechanism [9, 10]. The model was reinforced by the use of a Variational-Auto-Encoder (VAE) [11] that takes the target mel-spectrogram as input and predicts the mean and variance of a Gaussian distribution from which a latent representation will be sampled. We use adversarial training [12] in order to shift the distribution of predicted mel-spectrograms towards the distribution of target mel-spectrograms.The acoustic module models the following probability distribution:

$$p(\mathbf{y}_{1:M}) = \int \Pi_{m=1:M} p(\mathbf{y}_m | \mathbf{y}_{<m}, \mathbf{x}_{1:N}, \mathbf{z}) p(\mathbf{z}) d\mathbf{z} \quad (1)$$

Where $\mathbf{y} = \{\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_M\}$ is a sequence of mel-spectrogram frames, $\mathbf{x} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$ is a sequence of phoneme embeddings and $\mathbf{z}$ is the VAE latent representation extracted from the

10.21437/SSW.2021-14

target mel-spectrogram.
The vocoder is a parallel-Wavenet vocoder [13] with the addition of a VAE. The VAE-component takes as input the spectrogram predicted by the acoustic model and generates a latent representation (see section 2.5 for more details). The vocoder models the following distribution:

$$p(\mathbf{w}_{1:T}) = \Pi_{t=1:T} p(w_t|\mu(\mathbf{s}_{<t},,\mathbf{y}_{1:M},\mathbf{z}),\sigma(\mathbf{s}_{<t},,\mathbf{y}_{1:M},\mathbf{z})) \quad (2)$$

Where $\mathbf{w} = \{w_1, w_2, ..., w_T\}$ is the waveform, $\mathbf{y} = \{\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_M\}$ is a sequence of mel-spectrogram frames, $\mathbf{z}$ is the latent representation extracted from the target mel-spectrogram using the acoustic model's VAE module and $\mathbf{s} = \{\mathbf{s}_1, \mathbf{s}_2, ..., \mathbf{s}_T\}$ is a sequence of noise sampled from a prior random variable that serves as input to the Inverse Autoregressive Flow (IAF) [14] blocks of the neural vocoder.

2.2. Tuning auto-regression levels

The acoustic model predicting mel-spectrograms is an encoder-decoder architecture that uses location-sensitive attention mechanism [7, 9]. The decoder is an LSTM-based autoregressive module: at each decoder step, the decoder predicts a set of mel-spectrogram frames based on frames predicted in its previous step. We observed that the interaction between the decoder and the attention mechanism led to instabilities when generating very long sequences. Such instability issues include mumbling and skipping over phonemes.
To alleviate the instability issues and help the convergence of the attention mechanism, we changed slightly the decoder's architecture enabling it to predict multiple spectrogram frames at a time instead of one. With multiple frames predicted per decoder step, the decoder needs less steps to produce the same output spectrogram, this reduction in number of steps helps prevents instabilities from accumulating during synthesis. This trick greatly improved the stability of the model; a finding that was also discussed in [5]. However, the improved stability came at the cost of a decrease in segmental quality. In order to help stabilize the attention while maintaining the same level of audio quality, we tuned the decoder's number of outputs-per-step (ops) gradually from ops 5 to ops 2 within the same training. In order to tune the ops with no change to the architecture, we made the decoder predict the maximum ops (5 in this case) at all stages of the training. The decoder's output was then sliced depending on the current ops: for example, in ops 2 we would select only the first two predicted frames.
We made the decision to stop the tuning phase before reaching ops 1 (fully autoregressive model) because we observed, in the development process, that the ops 2 model generated samples of comparable quality to the ops 1 model while being faster.
Since the autoregression's tuning approach attempts to improve on the instability issues on the decoder side, it can still be combined with orthogonal approaches that tackled the problem from the attention mechanism's side [15, 16, 17, 18].

2.3. Using adversarial training for acoustic modeling

Generative adversarial networks (GANs)[19] have been successfully used to generate high quality images. [20, 21, 22]. The adversarial loss incentivizes the generator to produce images that are indistinguishable from real ones, thus mitigating the over-smoothing effect observed when using traditional losses such as L1 or L2 alone [23]. Following the progress made in image generation, GANs have started to be applied to TTS. Kaneko et al. [24, 25] applied GANs to train the post-filter component of the acoustic model to produce sharper mel-spectrograms. Other GAN configurations were also explored, such as applying GANs on the waveform for speech enhancement [26] and the use of GANs to mitigate exposure bias [27].
We explored the use of adversarial training to reduce over-smoothing in mel-spectrogram prediction via end-to-end training of our acoustic model. By adding an adversarial loss, we aim to encourage the network to output a spectrogram distribution that matches with that of the target and not only focus on the over-smoothing L1/ L2 losses. In our configuration (figure 2), the generator is the whole acoustic model and is trained using L1 loss between predicted and target mel-spectrograms plus the adversarial loss. The discriminator is trained to distinguish predicted spectrograms from target ones and is based on self-attention blocks following the same architecture as in [28]. The following equations summarize the training losses of both the discriminator and the generator:

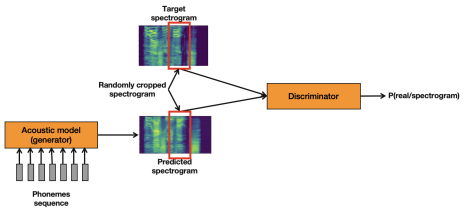


Figure 2: *GAN training setup. The whole acoustic model is the generator. The discriminator network classifies its input spectrogram as real or predicted.*

$$\begin{aligned} L_D = &- \mathop{\mathbb{E}}_{x\sim p_{data}} [min(0, -1 - D(G(x)))] \\ &- \mathop{\mathbb{E}}_{y\sim p_{data}} [min(0, -1 + D(y))] \end{aligned} \quad (3)$$

$$\begin{aligned} L_G = &\mathop{\mathbb{E}}_{x,y\sim p_{data}} [\|G(x) - y\|_1] \\ &+ \alpha \mathop{\mathbb{E}}_{x,y\sim p_{data}} [D(y) - D(G(x))] \\ &- \beta(step) KLD(p_z, p_{prior}) \end{aligned} \quad (4)$$

Where D is the discriminator network, G is the generator network, x is the phoneme input sequence, y is the target mel-spectrogram sequence, α is a weighting factor used to balance the contributions of the adversarial loss and the L1 reconstruction loss, z is the latent representation sampled from the distribution p_z whose parameters are predicted by the VAE encoder and β is a weighting factor for the Kullback-Leibler Divergence (KLD) loss used for the VAE training (see section 2.4).
We used spectral normalization [29] on the discriminator side to stabilize the discriminator's training. We also observed that feeding the whole mel-spectrogram sequence to the discriminator gave worse results than feeding a small random window of mel-spectrogram frames. We think that this approach forced the discriminator to focus on short-term transitions in the audio, thus explaining the improved audio quality.

2.4. Variational Auto-Encoders (VAEs)

The acoustic model is conditioned on phonetic input which does not account for latent (i.e. prosodical) factors in the data.

To be able to factorize these latent elements, we enhance the acoustic model via the addition of a Variational Auto-Encoder (VAE) [11] which produces a latent representation predicted from the target spectrogram. Similar approaches were used for style modelling using continuous hierarchical embeddings [30] or discrete ones [31]. The VAE module is a reference-encoder-like architecture made of a stack of convolutional neural networks followed by a BiLSTM and two projections that predict the mean and standard deviation of a 64-dimensional Gaussian distribution. The prior of the VAE latent vector is a Normal distribution with zero mean and unit variance. As such, the Kullback-Leibler Divergence (KLD) has a closed form equation.

When training with KLD loss, it is possible to observe KLD collapse: the decoder ignores the latent variable, thus keeping the posterior distribution similar to the uninformative standard Gaussian prior. To alleviate this issue, approaches such as annealing or introduction of skip connections have been proposed [32, 33, 34, 35]. We use a simple annealing scheme where the weighting factor of the KLD loss is gradually increased from 0 to 1 until a given step, after which the KLD loss is only periodically applied.

Another challenge faced while introducing VAE to our acoustic model is the selection of latent variable to use at inference time. Two main schemes can be used: sampling from the prior distribution of latent variables or providing a fixed latent representation at inference time for all utterances. For the second scheme, different variations can be used, such as using the mean of the prior distribution, using centroid (mean) computed over training data, or selecting a latent representation extracted from sampled utterances from the training set. We observed that the selected latent representation can have a big impact on the prosody and audio quality of generated samples. Furthermore, we observed that the latent representation extracted from spectrograms corresponding to utterances with flat/average prosody led to better observed segmental quality. After extensive listening, we chose a scheme where we use a latent vector extracted from an utterance with flat intonation for general speech, and a latent vector extracted from an utterance with rising intonation for yes/no questions. The acoustic model will use one of these latent representation at inference time depending on the domain.

2.5. VAE-enhanced parallel Wavenet

The vocoder is a Parallel Wavenet-like [13] architecture trained with probability density distillation and additional spectral loss term. To improve the vocoder's synthesized speech quality, we used a similar approach to [36] which conditioned both the teacher and student networks on additional VAE latent representation extracted from real-speech (section 2.4). Figure 3 shows how the VAE conditioning is performed.
The teacher model has a Wavenet-like architecture with mixture of 10 logistics where audio samples are conditioned on oracle mel-spectrograms and a 64-dimensional VAE latent representation extracted from the VAE encoder of the acoustic model. Mel-spectrogram frames are encoded by a 2-layers BiLSTM with 128 hidden size, they are then concatenated with a 64-dimensional VAE latent vector. We then apply an affine transformation, implemented as a 1x1 convolution, to the output concatenated vector. Finally, the conditioning representation is upsampled to align with audio samples. Every Residual Gated

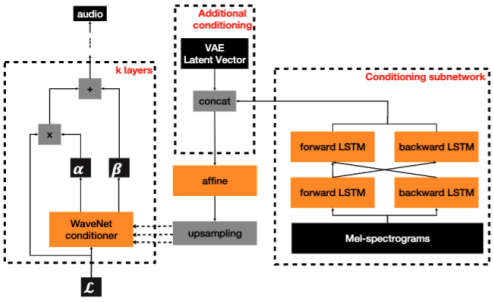


Figure 3: *The architecture of the Parallel WaveNet-like neural vocoder. The additional conditioning block concatenates the mel-spectrogram conditioning with the latent representation extracted from the VAE reference encoder of the acoustic model.*

CNN block uses 256-dimensional skip and gated channels. Filter activation is tanh and the gate activation is a sigmoid function.
The student network shares the same conditioning blocks with frozen weights with the teacher network. A logistic distribution is passed through a stack of 4 Inverse Autoregressive Flows [14] with affine transform. The parameters of the affine transform are predicted by autoregressive conditioner blocks similar in architecture to Wavenet blocks [4]. The flow conditioners contain dilated convolutions with 10, 10, 10 and 30 layers. The last block uses the same dilation value growth and reset as the teacher network. The dilated convolutions use 64-dimensional gated channels with tanh filter activation and sigmoidal gate activation.
We tested the effect of the size of the VAE latent representation by comparing a 16-dimensional representation against a 64-dimensional one and we observed that, while the 16-dimension teacher had a higher audio quality, the student conditioned on 16-dimension VAE struggled to properly learn the teacher distribution and generated noisier samples. We also explored the use of VAE conditioning on the student only, given the already high audio quality of the teacher, and found out that the student network was unable to properly match the teacher's distribution. This behavior needs more investigation and may drive future work. Another detail to take into account is which latent representation to use at inference time. Two schemes were examined: the use of a centroid (mean) computed over the training data and computing the latent representation from the conditioning spectrogram. Extensive listening led us to the choice of using the centroid conditioning.

3. Results

3.1. Experiments

To demonstrate the benefits of the techniques detailed in the previous section, we train two NTTS systems on highly-expressive data of a male, English-speaking, celebrity voice. The first system is a baseline model: an autoregressive acoustic model with 5 spectrogram frames given per each decoder step (with similar architecture to [7, 6]) followed by a parallel Wavenet vocoder. The second model is the baseline model enhanced by all four techniques described in Section 2 and we will refer to it as full-system in the remainder of the paper.

The acoustic model was trained separately from the vocoder using a batch size of 32. The training procedure for the auto-regression tuning was as follows: train the baseline with 5 output frames predicted per decoder step (denoted as ops for output-per-step), then tune the model as ops 4, followed by tuning the model as ops3 and finally as ops2. We used Adam optimizer with default parameters. Once the ops 2 tuning is finished, we tune the model using the introduced GAN module. In the GAN tuning phase, the β_1 parameter of the Adam optimizer was reduced. We train the model on 10 hours of expressive data of a male US voice.
In the vocoder's training procedure, we use a batch size of 16 and an Adam optimizer with default parameters. Learning rate decay wasn't used in student training, but it was used in the teacher network's training with a decay value of 0.95. We used Polyak averaging with a decay value of 0.999.
Our student was trained iteratively on 3 different saved snapshots of the teacher network, taken at different training steps, in order to to be able to train the student network while the teacher's training was still ongoing. We have observed through listening that the student trained this way produced better-sounding audios than the student trained only on the last snapshot of the teacher. This observation will need to be investigated, but we hypothesize that the teacher's distribution gets more complex and harder to model by the student in the teacher's later iterations and thus the iterative training provides the student with checkpoints that are easier to match during early training steps.

3.2. Evaluation

We conducted a MUSHRA (Multiple Stimuli with Hidden Reference and Anchor) evaluation test with three systems: the baseline, the full-improvements system and recordings from the speaker. The test was performed on a set of 160 utterances with varying lengths. Each utterance was evaluated by 15 native English speakers who were asked to rate the three systems based on naturalness. The results of the perceptual test can be viewed in figure 4. We report the following mean MUSHRA scores per system: baseline: 51.13, full-system: 64.04 and recordings: 84.04. These numbers translate into the full-system achieving a 39% gap-closing between the baseline and the recordings.

We also conducted VQA tests (Voice Quality Assessment) of the baseline and our proposed system where two native US English speakers were asked to report on issues they hear in audio files synthesized using both systems. The test set is comprised of 275 utterances with varying lengths and from different domains: questions, spelling, newscasting, etc. The issues were classified according to severity, from critical to almost unnoticeable (reported as minor). The reported issues covered audio quality, pronounciation issues and instabilities. Table 1 summarizes the VQA results, where we can observe a significant reduction in terms of reported issues. The biggest difference between the two systems was reported on audio buzziness issues: as the baseline system had 153 reported problems compared to 60 for the proposed system.

In addition to the above MUSHRA test, we conducted an additional ablation test in order to rank the improvement made by each change. The test has the exact same setup as the previously described MUSHRA except that we use 4 systems instead

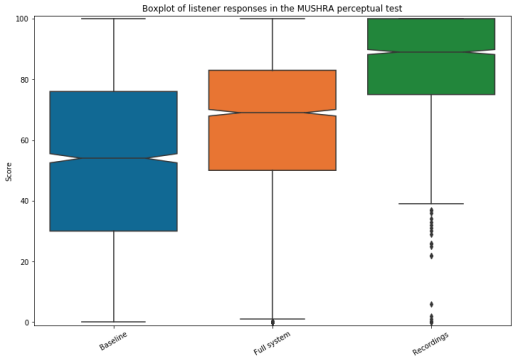


Figure 4: *Boxplot of MUSHRA test's results. Systems from left to right: baseline, full-system and recordings. Mean score per system: baseline: 51.13, full-system: 64.04 and recordings: 84.04.*

System	Number of reported **critical** issues	Number of reported **medium** issues	Number of reported **minor** issues
Baseline	7	443	252
Full-system	0	231	123

Table 1: *Summary of VQA tests performed on both the baseline and the proposed system. It is worth noting that a single issue can be reported twice as there are two testers.*

of 3. The 4 systems are: full system without ops tuning, full system without GAN training, full system without VAE in acoustic model side and full system without VAE on neural vocoder's side. Table 2 summarizes the results of the MUSHRA.

From the results of table 2, we observe that not all techniques are contributing equally to the overall improvements. Conditioning the Parallel Wavenet vocoder on VAE embeddings is the most impactful as the full system suffered the most without it. The second most important change is the tuning of ops within training. The GAN training and the VAE conditioning in the acoustic model side seem to have the least impact on the overall gains. This observation does not mean that the latter VAE should be discarded altogether since we noted, in section 2.2, that the component helped guide the question/non-question intonations.

System	MUSHRA score
w/o ops tuning	66.62 ± 1.06
w/o GAN training	67.64 ± 1.03
w/o VAE in acoustic modeling	68.19 ± 1.03
w/o VAE in vocoder	65.88 ± 1.07

Table 2: *Mean MUSHRA score per system in ablation study.*

4. Conclusions

The paper tackled the challenge of improving the audio quality of speech produced by a TTS system trained on highly-expressive data. To that end, we presented a compilation of techniques that ranged from acoustic modeling to vocoding. We then showed that, when combined, the proposed techniques improved the perceived quality which translated into a considerable increase in MUSHRA score and a reduction to the number of reported audio quality issues. We also run an ablation MUSHRA test to rank the impact of the proposed techniques.

5. References

[1] A. W. Black and N. Campbell., "Optimising selection of units from speech databases for concatenative synthesis," 1995.

[2] Y. Qian, F. K. Soong, and Z.-J. Yan, "A unified trajectory tiling approach to high quality speech rendering," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, pp. 280–290, 2013.

[3] T. Merritt, R. A. J. Clark, Z. Wu, J. Yamagishi, and S. King, "Deep neural network-guided unit selection synthesis," *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5145–5149, 2016.

[4] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *CoRR*, vol. abs/1609.03499, 2016. [Online]. Available: http://arxiv.org/abs/1609.03499

[5] Y. Wang, R. J. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio, Q. V. Le, Y. Agiomyrgiannakis, R. Clark, and R. A. Saurous, "Tacotron: A fully end-to-end text-to-speech synthesis model," *CoRR*, vol. abs/1703.10135, 2017. [Online]. Available: http://arxiv.org/abs/1703.10135

[6] J. Latorre, J. Lachowicz, J. Lorenzo-Trueba, T. Merritt, T. Drugman, S. Ronanki, and K. Viacheslav, "Effect of data reduction on sequence-to-sequence neural TTS," *CoRR*, vol. abs/1811.06315, 2018. [Online]. Available: http://arxiv.org/abs/1811.06315

[7] N. Prateek, M. Lajszczak, R. Barra-Chicote, T. Drugman, J. Lorenzo-Trueba, T. Merritt, S. Ronanki, and T. Wood, "In other news: A bi-style text-to-speech model for synthesizing newscaster voice with limited data," *CoRR*, vol. abs/1904.02790, 2019. [Online]. Available: http://arxiv.org/abs/1904.02790

[8] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. J. Skerry-Ryan, R. A. Saurous, Y. Agiomyrgiannakis, and Y. Wu, "Natural TTS synthesis by conditioning wavenet on mel spectrogram predictions," *CoRR*, vol. abs/1712.05884, 2017. [Online]. Available: http://arxiv.org/abs/1712.05884

[9] J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," *CoRR*, vol. abs/1506.07503, 2015. [Online]. Available: http://arxiv.org/abs/1506.07503

[10] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *ICLR*, 2015. [Online]. Available: https://arxiv.org/abs/1409.0473

[11] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprints*, 2013. [Online]. Available: https://arxiv.org/abs/1312.6114

[12] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *arXiv preprints*, 2014. [Online]. Available: https://arxiv.org/abs/1406.2661

[13] A. van den Oord, Y. Li, I. Babuschkin, K. Simonyan, O. Vinyals, K. Kavukcuoglu, G. van den Driessche, E. Lockhart, L. C. Cobo, F. Stimberg, N. Casagrande, D. Grewe, S. Noury, S. Dieleman, E. Elsen, N. Kalchbrenner, H. Zen, A. Graves, H. King, T. Walters, D. Belov, and D. Hassabis, "Parallel wavenet: Fast high-fidelity speech synthesis," *CoRR*, vol. abs/1711.10433, 2017. [Online]. Available: http://arxiv.org/abs/1711.10433

[14] D. P. Kingma, T. Salimans, and M. Welling, "Improving variational inference with inverse autoregressive flow," *CoRR*, vol. abs/1606.04934, 2016. [Online]. Available: http://arxiv.org/abs/1606.04934

[15] E. Battenberg, R. Skerry-Ryan, S. Mariooryad, D. Stanton, D. Kao, M. Shannon, and T. Bagby, "Location-relative attention mechanisms for robust long-form speech synthesis," 2019.

[16] J.-X. Zhang, Z.-H. Ling, and L.-R. Dai, "Forward attention in sequence- to-sequence acoustic modeling for speech synthesis," *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr 2018. [Online]. Available: http://dx.doi.org/10.1109/ICASSP.2018.8462020

[17] C. Raffel, M.-T. Luong, P. J. Liu, R. J. Weiss, and D. Eck, "Online and linear-time attention by enforcing monotonic alignments," 2017.

[18] M. He, Y. Deng, and L. He, "Robust sequence-to-sequence acoustic modeling with stepwise monotonic attention for neural tts," 2019.

[19] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," 2014.

[20] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," 2018.

[21] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther, "Autoencoding beyond pixels using a learned similarity metric," 2015.

[22] E. Denton, S. Chintala, A. Szlam, and R. Fergus, "Deep generative image models using a laplacian pyramid of adversarial networks," 2015.

[23] W. Lotter, G. Kreiman, and D. D. Cox, "Unsupervised learning of visual structure using predictive generative networks," *CoRR*, vol. abs/1511.06380, 2015. [Online]. Available: http://arxiv.org/abs/1511.06380

[24] T. Kaneko, H. Kameoka, N. Hojo, Y. Ijima, K. Hiramatsu, and K. Kashino, "Generative adversarial network-based postfilter for statistical parametric speech synthesis," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2017, pp. 4910–4914.

[25] T. Kaneko, S. Takaki, H. Kameoka, and J. Yamagishi, "Generative adversarial network-based postfilter for stft spectrograms," in *The Annual Conference of the International Speech Communication Association (Interspeech)*, August 2017.

[26] S. Pascual, A. Bonafonte, and J. Serrà, "Segan: Speech enhancement generative adversarial network," 2017.

[27] H. Guo, F. K. Soong, L. He, and L. Xie, "A new gan-based end-to-end tts training algorithm," 2019.

[28] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention generative adversarial networks," 2018.

[29] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," 2018.

[30] W.-N. Hsu, Y. Zhang, R. J. Weiss, H. Zen, Y. Wu, Y. Wang, Y. Cao, Y. Jia, Z. Chen, J. Shen, P. Nguyen, and R. Pang, "Hierarchical generative modeling for controllable speech synthesis," 2018.

[31] Y. Wang, D. Stanton, Y. Zhang, R. Skerry-Ryan, E. Battenberg, J. Shor, Y. Xiao, F. Ren, Y. Jia, and R. A. Saurous, "Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis," 2018.

[32] Z. Yang, Z. Hu, R. Salakhutdinov, and T. Berg-Kirkpatrick, "Improved variational autoencoders for text modeling using dilated convolutions," 2017.

[33] A. B. Dieng, Y. Kim, A. M. Rush, and D. M. Blei, "Avoiding latent variable collapse with generative skip models," 2018.

[34] Y. Kim, S. Wiseman, A. C. Miller, D. Sontag, and A. M. Rush, "Semi-amortized variational autoencoders," 2018.

[35] T. Zhao, R. Zhao, and M. Eskenazi, "Learning discourse-level diversity for neural dialog models using conditional variational autoencoders," 2017.

[36] J. Rohnke, T. Merritt, J. Lorenzo-Trueba, A. Gabrys, V. Aggarwal, A. Moinet, and R. Barra-Chicote, "Parallel wavenet conditioned on vae latent vectors," 2020.

11th ISCA Speech Synthesis Workshop (SSW 11)
26-28 August 2021, Budapest, Hungary

Are we truly modeling expressiveness? A study on expressive TTS in Brazilian Portuguese for real-life application styles

Lucas H. Ueda[1], *Paula D.P. Costa*[1], *Flavio O. Simoes*[2], *Mario U. Neto*[2]

[1]Department of Computer Engineering and Automation, School of Electrical and Computer Engineering, University of Campinas, Campinas, Brazil
[2]Research and Development Institute CPQD, Campinas, Brazil

l156368@dac.unicamp.br, paulad@unicamp.br, simoes@cpqd.com.br, uliani@cpqd.com.br

Abstract

This paper presents a study of expressive speech synthesis applied to real-life application styles in Brazilian Portuguese. We explore the use of data with different recording conditions in state-of-the-art architectures in expressive TTS. Our results suggest that the variability of recording conditions of the same style, combined with a guided training of the latent representation space of the Reference Encoder, assists in the modeling of non-archetypal expressivities. Additionally, we propose an alternative to evaluating the model's ability to generate expressive speech during preliminary results, based on a classifier using GeMAPS features.

Index Terms: expressive speech synthesis, tacotron2, style guided, prosodic features, gemaps

1. Introduction

The process of generating an artificial speech from a given text is named text-to-speech (*TTS*) synthesis. This artificial speech should correctly convey the message that was in the input text (intelligibility) and, ideally, sounding like a human (realism/naturalness) while having the correct prosody (expressiveness). The usage of recent deep neural networks architectures in *TTS*, neural *TTS* (*NTTS*), have established a new state of the art, being able to generate intelligible artificial speech with a naturalness close to the human voice [1, 2, 3, 4]. However, state-of-the-art *NTTS* models can still not synthesize realistic, expressive speech or modulate existing models for different styles. Because of this, the synthesis of expressive speech is still a challenge.

To modulate expressiveness in speech, we need to change the prosody. According to [5, 6] prosody are speech characteristics that are not associated with "what is said", but with "how it is said". Similarly to [7] we define prosody as the variation in speech signal that remains after accounting for variation due to phonetics, speaker identity, and channel effects. There are several parameters that can be changed to modify speech prosody, mostly related to fundamental frequency, intensity, and duration during speech. Recent approaches have proposed a global control of prosody, where a latent representation of prosody (*LRP*) is estimated in conjunction with *NTTS* models. From this latent space, these approaches showed to be able to alter the speech prosody without having to control specific acoustic parameters. Although these approaches show that modeling speech prosody without explicit specifications is feasible, such approaches do not control the expressiveness of speech itself.

Later works seek to use these architectures based on estimating the *LRP* to generate different speech styles. When estimating the *LRP* space using data with varying speech styles, the modeled space itself showed other characteristics for each of them. From the analysis of this space, such works showed that it was possible to control the speech style. However, usually, these approaches use archetypal styles of speech, clearly distinguishable from each other.

Based on the availability of lines recorded in different recording styles and conditions, this work explores the ability of state-of-the-art models to model the prosodic space (*LRP*). The present work investigates the robustness of current methods of expressive speech synthesis with a diversity of recording conditions and real-life application styles.

We explore different dataset configurations under different recording conditions, as well as different methodologies for estimating LRP space. Our experiments showed that Global Style Tokens [8] are not capable of generating separable prosodic spaces, while Reference Encoder [7] easily generates separable spaces. However, the space generated does not guarantee that different styles are modeled. The use of the guidance in the estimation of the *LRP* showed feasibility in conditioning the space to better prosodic modeling. Additionally, we propose an alternative to subjective perceptual evaluation suitable for intermediate stages of *TTS* development, based on an expressivity classifier that uses *GeMAPS* [9] set of acoustic features.

2. Related works

Three acoustic parameters are mainly related to speech prosody: fundamental frequency, intensity, and duration. Moreover, we can classify prosody in two categories: affective and augmentative [10]. The affective prosody is the expression of meaning related to emotion, mental state, and speaker attitude. On the other hand, augmentative prosody does not contain any extra information; it is used to make a verbal communication clearly by giving intonation or focus in specific parts, disambiguating one message that could be interpreted in different ways.

Silva and Barbosa [11] tried to verify how to relate prosodic acoustic features with the perception of people listening to emotional speech. Despite a good correlation of certain features, many of them did not have a significant relationship and were unstable over different emotions. The high complexity of how prosody is related to acoustic features is a big challenge when talking about controlling expressiveness in artificial speech.

In order to avoid the need of explicit annotations in prosody modeling, Skerry-Ryan and colleagues [7] proposed the Reference Encoder (*RE*), which consists of an additional module to the *NTTS Tacotron* architecture [1] that encodes the mel spectrogram of a reference audio in a lower-dimensional representation. This representation is added to the decoder input and used to control prosody at inference time. This work showed

10.21437/SSW.2021-15

that the augmented *Tacotron* with *RE* can transfer prosody from reference audio to the synthesized speech. Later, Wang and colleagues [8] proposed an attention layer augmenting the Reference Encoder, where trainable variables, named Global Style Tokens *(GST)*, are jointly estimated with the model parameters in training time. This approach shows that each token can model distinct acoustic parameters, like pitch and duration. Although these two works have shown that it is possible to use this lower-dimensional representation to control prosody in a *NTTS* systems, they didn't perform any relation between these parameters and controlling a specific speech style.

Further, other works tried to model different speech styles using these lower-dimensional representations, which we are calling Latent Representation of Prosody (*LRP*). In [12] the *GST's* were used conditioning each token to a specific emotion label. By doing that, they were able to control the synthesized speech over three different emotions by using the respective modeled token. Kwon, Jang, Ahn and Kang [13] also used the *GST* architecture, but instead of conditioning the tokens themselves, they studied the *LRP* space generated by tokens over speech samples. This work demonstrated that using the centroid's of each emotional speech in *LRP* space can lead to expressive speech control in inference time. However, those approaches used a balanced internal dataset among different styles, which is not easily feasible.

Recently, Sorin, Shechtman and Hoory [14] proposed an approach where a *NTTS* model augmented by *RE* module was used to generate expressive speech. Particularly, this work showed that it is possible to transfer expressiveness among different speakers even when just one speaker's expressive speech is available. Using a dataset consisting of speech from 3 speakers, where just one of them had a small amount of expressive data, they trained a multi-speaker *Tacotron2* [2] architecture with *RE* module to generate the *LRP* space. Using *PCA* decomposition, they were able to select a good representation for each style to generate controlled expressive speech in inference time to all speakers. Specifically, in this project, they used real-life application speech styles called "good news" and "apology", besides the neutral. This work is particularly important because real-life application styles are not so easily distinguishable from each other, unlike what happens when dealing with emotional speech..

As far as we know, the only work that applies *NTTS* approach to Brazilian Portuguese language is in [15], where several experiments were done using different *NTTS* architectures on a public dataset made by themselves. However, they don't contemplate the expressive speech synthesis problem, having only neutral samples. We understand this project as the first one considering the expressive *NTTS* problem using Brazilian Portuguese language.

3. Technical setup

3.1. Data

Our experiments were conducted using a proprietary dataset consisting of utterances recorded by three speakers identified as: *Speaker 1* (female), *Speaker 2* (female), and *Speaker 3* (male). Table 1 presents a summary of the dataset, which is characterized by a high volume of neutral recordings, and a smaller set of expressive speech samples. The expressive style is associated to real-life customer service applications and can be described as excited positively; we refer to this as *Enthusiastic* style. Also, the expressive utterances were recorded only by Speaker 1 in two different conditions: medium quality (with high reverberation) and studio quality.

In order to create different experimental setups, four dataset configurations were designed, each one characterized by a particular selection of utterances / recording conditions. The first configuration, named *DC1*, consists of all available neutral data from all three speakers, together with all expressive utterances recorded in high reverberation condition. This configuration is, theoretically, characterized by a more easily separable latent subspace, since utterances differ both in terms of expressiveness and recording condition at the same time.

The other three settings are more challenging and have been used to cover three possible combinations of recording conditions for the expressive data. In the neutral partition of these settings only phonetically rich sentences are present, which represents 30% of the neutral speech contained in *DC1* (approximately 2 hours of speech per speaker). The main motivations to reduce the amount of neutral data were: (1) to be able to train a good *NTTS* model with less data (since the selected sentences guarantee a good phonetic coverage) and (2) to assure a better balance between the neutral data and the expressive data.

For each selection of the expressive data, a different dataset configuration was designed. Therefore, *DC2* contains only expressive data with high reverberation; *DC3* contains expressive data recorded in studio quality; and finally, *DC4* contains expressive data collected in both recording conditions. A detailed view of data configurations and audio samples are presented on our demo webpage[1].

3.2. Proposed Approaches

Our *NTTS* architecture is based on Tacotron2 [2] from MozillaTTS implementation. Tacotron2 is a state-of-the-art *NTTS* model that maps grapheme or phoneme sequences into mel spectrograms. The predicted mel spectrograms are synthesized using Griffin-Lim vocoder [16]. To accomplish the multi-speaker modeling, we added a speaker identity embedding layer. The speaker embedding output is then broadcast-concatenated to the decoder input.

To generate *LRP* space, we augmented the Tacotron2 architecture with different style encoders. The first architecture was based on a *GST* module with six tokens and four heads, and we name this as simple GST-Tacotron2 architecture. The choice for six tokens and four heads was an attempt to avoid the high degree of freedom that *GST* has. In the second one, we removed GST module and used only the Reference Encoder (*RE-Tacotron2*) without the last fully connected layer, similarly to [14]. Finally, the third architecture is based on Reference Encoder with an additional layer that receives as input the *LRP* generated by *RE* and classifies the expressivity of the reference audio in a supervised manner. In order to accomplish the style-guided modeling, we added Cross-Entropy Loss to the Tacotron2 loss function that measures the error of the classifier layer. We call this last one Style Guided RE-Tacotron2 (*SGRE-Tacotron2*). For all architectures, the style encoder layer's output is broadcast-concatenated with the decoder input, similarly to the speaker embedding.

In order to generate controlled expressive speech using the *LRP* space, we used the centroid's of each expressivity and, additionally, the expressive point with maximum distance in comparison with the neutral centroid. This last point is an attempt to generate an utterance "as expressive as possible" at inference

[1] https://bit.ly/3qJfwm5

Table 1: *Proprietary Dataset Available*

Speaker ID	Gender	Style	Recording Conditions	Data Partition	#Utterances	#Hours
Speaker 1	Female	Expressive	Medium Quality	All Available	381	0.45
		Expressive	Studio Quality	All Available	265	0.29
		Neutral	Studio Quality	All Available	7209	6.38
		Neutral	Studio Quality	Phonetically rich	2389	2.24
Speaker 2	Female	Neutral	Studio Quality	All Available	17992	15.00
		Neutral	Studio Quality	Phonetically rich	2543	2.61
Speaker 3	Male	Neutral	Studio Quality	All Available	5173	5.1
		Neutral	Studio Quality	Phonetically rich	1828	2.04

time; we refer to this point as Expressive Maximum Distance Point (*EMDP*).

3.3. Model Evaluation

A fundamental problem in the development of expressive speech synthesis models is how to evaluate the contribution of parameter or architectural changes to the final naturalness of speech. Typically, subjective perceptual evaluation is needed to assess the realism of synthesized speeches as well as the capability of the model in generating consistent speech styles. However, subjective perceptual evaluation is a laborious and time-consuming task, and it is not affordable for intermediate stages of development.

As an alternative to the subjective evaluation, we propose two objective metrics to evaluate how well our model fitted our target expressive style.

The ROC-AUC metric of a Logistic Regression trained on the learned *LPR* space can measure how well a linear model can classify the space. It is a proxy variable to show how separable are the expressive styles (neutral and expressive) in the *LPR* space. However, not only prosodic information is observed by the model. Since we have no control on what exactly is being modeled, it is not possible to state that a well separable space represents a model learned to distinguish only different prosodic styles.

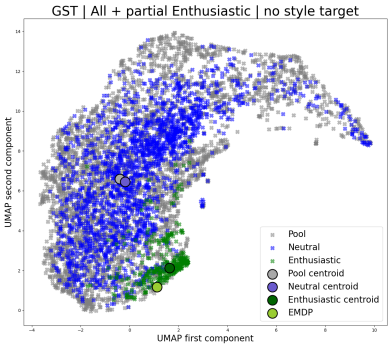
As a second objective evaluation approach, we trained a robust classifier based on the Geneva Minimalistic Acoustic Parameter Set (GeMAPS) features extracted from only Studio Quality data (*DC3*) and use this model to classify whether synthesized audio is expressive or not based only on acoustic features [9]. This set of features was elaborated aiming at a set of acoustic parameters that shows a good performance in affective computing tasks applied to speech. The GeMAPS features are characterized by: (1) their potential to index affective physiological changes in voice production, (2) their proven value in former studies as well as their automatic extractability, and (3) their theoretical significance in affect theory. The minimalistic set of features consists of 62 parameters extracted from 18 Low-level descriptors (*LLD*) based on frequency (pitch, jitter, formant's frequency), energy/amplitude (shimmer, loudness, harmonics-to-noise ratio), and spectral (alpha ratio, hammarberg index, spectral slope, formant's relative energies, and harmonic differences). Particularly, we used the extended version, called eGeMAPS, that has seven *LDD* added: spectral (MFCC and spectral flux) and frequency (Formant's). As discussed in [9], cepstral parameters have proven to be highly successful in modeling affective states. In total, the extended set of features, eGeMAPS, contains 88 parameters. In our experiments We trained a 5-fold Random Forest classifier that recognizes whether an audio is expressive or not with an accuracy of 98%, based on eGeMAPS features, the eGeMAPS classifier. To assess the ability of our *NTTS* model to generate consistent expressiveness, we select 150 utterances, never seen in training, and classify the synthesized speech samples using the classifier. The synthesized speech samples are conditioned to the *EMDP* point, the eGeMAPS features are extracted from the output audio and then classified by the classifier. It is therefore expected that the classifier will be able to focus on prosodic features and indicate whether the synthesized speech is expressive or not.

4. Results

4.1. Low Dimensional Representations

Initially, we seek to evaluate different approaches in building the *LRP* space applied to the configuration of *DC1* data. The *GST-Tacotron2* architecture was the first to be evaluated, where some works have already used the approach to archetypal styles (such as Ekman's emotions). We used the *UMAP* [17] technique to project the *LRP* space to a 2D representation and we analyze the distribution of styles in this lower space, Figure 1 shows the 2D-dimensional projection given by *UMAP*. The green points represent the representations of the expressive data and the blue points the neutral ones, both from Speaker 1, while the gray (Pool) points are the representations of the other speakers' neutral data.

Figure 1: *UMAP projection of* LRP *space generated by GST-Tacotron trained on* DC1.



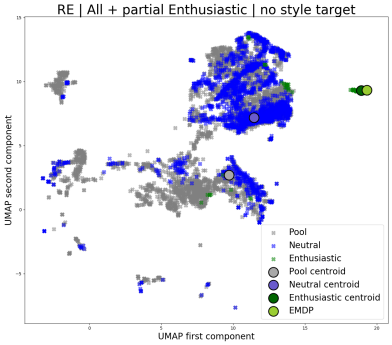
Although the architecture managed to concentrate the data labeled as expressive in a specific region of space, that same re-

gion still consisted of neutral data from all speakers. It was not possible to achieve an adequate modeling of the speech style in when listening to the artificial audios generated by conditioning the synthesis to several regions of the space (centroid's or even the *EMDP* point), as well as the points described in the proposed approaches.

Similar to what was observed in [14], *GST* was not able to model different speech styles under the present conditions of experimentation. Unlike [13] for example, we don't have a balanced dataset between different styles, and also we don't use archetypal styles. Because of this, we have chosen to use only the Reference Encoder as our speech style encoder.

We note that, the *RE-Tacotron2* architecture trained on *DC1* was able to generate a separable space between the expressive and neutral data for all speakers, Figure 2. However, it is also noted that the generated *LRP* space is more sparse, having several agglomerations along with data of the same style. When listening to the generated audios, this architecture showed to be more effective to modify acoustic parameters in the synthesized speech by conditioning it to regions of neutral or expressive data. However, when conditioning to the *EMDP* point, the hearing quality of the synthesized speech seemed more associated with the recording condition of the expressive subset than with the speaking style itself.

Figure 2: *UMAP projection of* LRP *space generated by RE-Tacotron trained on* DC1.



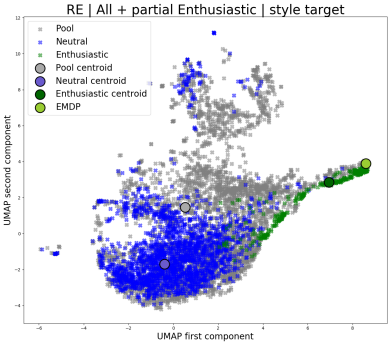
We then performed the same experiment, using the SGRE-Tacotron2 architecture. We noticed that the generated *LRP* space guided in a supervised way is less separable than in the second experiment, but still more separable than in the first one, as illustrated by Figure 3.

In this experiment, we noticed that with the *EMDP* point, the artificial speech generated is closer to the expressive style while still modeling the different recording conditions. When using the expressive centroid for inference, there was no explicit modeling of the style. We, therefore, chose to follow the *SGRE-Tacotron2* architecture in the subsequent experiments. Table 2 summarizes the conclusions of these three experiments.

4.2. Data diversity experiments

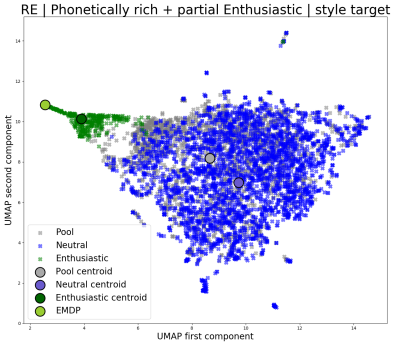
Based on the experiments in the *LRP* space described in the previous topic, we choose to continue using the *SGRE-Tacotron2*

Figure 3: *UMAP projection of* SGRE-Tacotron2 *space generated by SGRE-Tacotron trained on* DC1.



architecture. Next, we tried to assess how different data conditions influence the technique's ability to generate expressive artificial speech. First of all, we noticed that a large amount of data is not necessary if you have a smaller amount of phonetic richness data. In addition, a smaller amount of neutral data allows a better balance of the dataset in relation to expressive data. The architecture was then trained with the *DC2* data. The generated *LRP* space is clearly separable, as shown in Figure 4. Apparently, a more balanced amount of expressive data helps the model to generate a more separable *LRP* space. In addition, as previously noted, the model is able to generate expressive speech by conditioning to the *EMDP* point but still jointly models the recording condition.

Figure 4: *UMAP projection of* LRP *space generated by SGRE-Tacotron trained on* DC2.



The *LRP* space generated by this approach allows a linear model to reach an 87.33% ROC-AUC in the validation set, showing itself to be a highly separable space. However, only 48% of the synthetic utterances conditioned on the *EMDP* point were classified as expressive by the eGeMAPS classifier, which

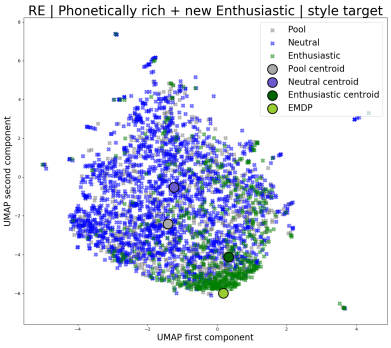
Table 2: *Summary of Low Dimensional Representation experiments*

Model Architecture	Data Configuration	Comment
GST-Tacotron2	DC1	Not able control acoustic features with Tokens or generate a separable space
RE-Tacotron2	DC1	Can generate a separable space, but no certain of modeling prosodic information
SGRE-Tacotron2	DC1	Can guide *LRP* space to a better prosodic modeling but with no clear consistency in controlling expressivity

supports our hypothesis that the architecture models not only the style, but also the recording condition.

DC3 was used in our next test setup. In this case, both neutral and expressive speech have the same recording condition. This approach leads to a less separable *LRP* space, with a lower ROC-AUC value (81.74). Moreover, when listening to the synthetic utterances they sound less distinguishable with regard to the target style, Figure 5. In practice, this experiment resulted in audios close to the neutral style, even when conditioned to the *EMDP* point. Despite this, our eGeMAPS classifier model recognized 97.33% of the synthesized audios as expressive.

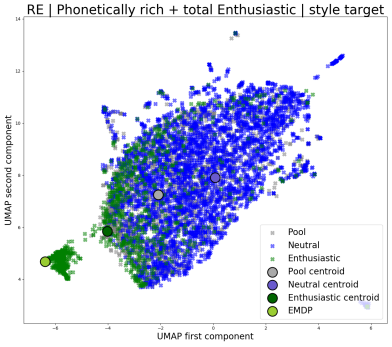
Figure 5: *UMAP projection of* LRP *space generated by SGRE-Tacotron trained on* DC3.



Our last experiment used all the available expressive data (*DC4*), maintaining the same *SGRE-Tacotron2* architecture. The generated *LRP* space remains separable, but the studio quality expressive samples are clearly closer to the neutral samples than the medium quality ones, as shown in Figure 6. In this space, the linear model achieves a ROC-AUC of 85.52, and the eGeMAPS classifier model was able to identify 87.33% of the artificial utterances generated using the *EMDP* point. With this configuration, the artificial speech generated by the model sounds more expressive and with less recording condition, as if the recording condition highlighted the expressive data allowing the model to capture the prosody of expressiveness. Table 3 summarize all experiments reported in this topic.

Even though the first experiment resulted in a more separable space, it mostly models the recording condition itself. As a result, even when conditioning the model to the *EMDP* point, our eGeMAPS classifier could not recognize such synthesized speeches as actually expressive. On the other hand,

Figure 6: *UMAP projection of* LRP *space generated by SGRE-Tacotron trained on* DC4.



when we train the model only on studio quality data, the generated space does not seem so separable, but the eGeMAPS model can identify the synthesized speeches as expressive. However, when listening to such audios, they are very similar to neutral ones, indicating a possible instability of the eGeMAPS classifier model. On the other hand, when we used all the expressive data in the third experiment, the SGRE-Tacotron2 was able to model less recording condition while still having a separable *LRP* space. It indicates that the variability of recording conditions for the same speech style can prevent only the recording condition from being modeled while still modeling prosody. The results suggest that not even guided training can guarantee that prosodic features will be properly modeled. Finally, the presence of a second distinguishing factor among utterances, as in the first experiment, reinforces the architecture's ability to model all speech aspects present in that group (prosodic characteristics and recording condition).

5. Conclusions and Discussions

The use of latent representations of audios (*LRP*) to model acoustic parameters of speech has been widely used to deal with expressive speech synthesis. Although this approach does manage to model speech characteristics, it did not explicitly guarantee the modeling of prosodic parameters in isolation under experimentation conditions of this work. Archetypal speech styles are associated with striking features in speech, allowing the model to separate its characteristics from one another easily. However, when dealing with styles typically used in real

Table 3: *Summary of Data Diversity experiments*

Model Architecture	Data Configuration	ROC-AUC	eGeMAPS expressive accuracy
SGRE-Tacotron2	DC2	87.33	48.00%
SGRE-Tacotron2	DC3	81.00	97.33%
SGRE-Tacotron2	DC4	84.46	87.33%

life applications, the task becomes more challenging. Our experiments suggest that the model tends to model the audios' most striking features, be them recording conditions or prosody. The variability of recording conditions in utterances of the same style, combined with a guided training of the *LRP* space, proved to be a promising approach to deal with styles that are not so easily distinguishable from each other.

Additionally, the present work is innovative in terms of expressive speech synthesis based on *NTTS* models applied to Brazilian Portuguese, and it can serve as a basis for future studies along the same line. Moreover, we have proposed the eGeMAPS Classifier as preliminary objective metric to evaluate expressive TTS models.

As a future work, we intend to continue studying the influence of different recording conditions on different architectures to help model real-life application speaking styles. We also intend to evaluate the use of prosodic variables to highlight such characteristics within the *LRP* spaces. Additionally, we intend to enrich our dataset with additional styles in order to perform similar experiments using two different style besides neutral one.

6. Acknowledgment

The authors would like to thank the Research and Development Institute CPQD and the Ministry of Science, Technology and Innovations for supporting and funding this project.

7. References

[1] Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio, Q. Le, Y. Agiomyrgiannakis, R. Clark, and R. A. Saurous, "Tacotron: Towards End-to-End Speech Synthesis," pp. 4006–4010, Aug. 2017. [Online]. Available: http://www.isca-speech.org/archive/Interspeech_2017/abstracts/1452

[2] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan, R. A. Saurous, Y. Agiomvrgiannakis, and Y. Wu, "Natural TTS Synthesis by Conditioning Wavenet on MEL Spectrogram Predictions," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 2018, pp. 4779–4783, iSSN: 2379-190X.

[3] W. Ping, K. Peng, A. Gibiansky, S. O. Arık, A. Kannan, S. Narang, J. Raiman, and J. Miller, "Deep Voice 3: Scaling Text-To-Speech with convolutional sequence learning," p. 16, 2018.

[4] N. Li, S. Liu, Y. Liu, S. Zhao, and M. Liu, "Neural Speech Synthesis with Transformer Network," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, pp. 6706–6713, Jul. 2019, number: 01. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/4642

[5] G. O'Grady, "D. Robert Ladd. 2008. Intonational phonology," *Functions of Language*, vol. 17, no. 2, pp. 276–294, Jan. 2008, publisher: John Benjamins. [Online]. Available: https://www.jbe-platform.com/content/journals/10.1075/fol.17.2.08ogr

[6] P. A. Barbosa, *Prosódia*. Parábola, May 2019.

[7] R. Skerry-Ryan, E. Battenberg, Y. Xiao, Y. Wang, D. Stanton, J. Shor, R. Weiss, R. Clark, and R. Saurous, "Towards End-to-End Prosody Transfer for Expressive Speech Synthesis with Tacotron," Mar. 2018.

[8] Y. Wang, D. Stanton, Y. Zhang, R. Skerry-Ryan, E. Battenberg, J. Shor, Y. Xiao, F. Ren, Y. Jia, and R. A. Saurous, "Style Tokens: Unsupervised Style Modeling, Control and Transfer in End-to-End Speech Synthesis," p. 10, 2018.

[9] F. Eyben, K. R. Scherer, B. W. Schuller, J. Sundberg, E. André, C. Busso, L. Y. Devillers, J. Epps, P. Laukka, S. S. Narayanan, and K. P. Truong, "The Geneva Minimalistic Acoustic Parameter Set (GeMAPS) for Voice Research and Affective Computing," *IEEE Transactions on Affective Computing*, vol. 7, no. 2, pp. 190–202, Apr. 2016, conference Name: IEEE Transactions on Affective Computing.

[10] P. Taylor, *Text-to-Speech Synthesis*. Cambridge: Cambridge University Press, 2009. [Online]. Available: https://www.cambridge.org/core/books/texttospeech-synthesis/D2C567CEF939C7D15B2F1232992C7836

[11] W. d. Silva and P. A. Barbosa, "Perception of emotional prosody: investigating the relation between the discrete and dimensional approaches to emotions," *REVISTA DE ESTUDOS DA LINGUAGEM*, vol. 25, no. 3, pp. 1075–1103, Jun. 2017, number: 3. [Online]. Available: http://periodicos.letras.ufmg.br/index.php/relin/article/view/10972

[12] P. Wu, Z. Ling, L. Liu, Y. Jiang, H. Wu, and L. Dai, "End-to-End Emotional Speech Synthesis Using Style Tokens and Semi-Supervised Training," in *2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. Lanzhou, China: IEEE, Nov. 2019, pp. 623–627. [Online]. Available: https://ieeexplore.ieee.org/document/9023186/

[13] O. Kwon, I. Jang, C. Ahn, and H.-G. Kang, "An Effective Style Token Weight Control Technique for End-to-End Emotional Speech Synthesis," *IEEE Signal Processing Letters*, vol. 26, no. 9, pp. 1383–1387, Sep. 2019, conference Name: IEEE Signal Processing Letters.

[14] A. Sorin, S. Shechtman, and R. Hoory, "Principal Style Components: Expressive Style Control and Cross-Speaker Transfer in Neural TTS," in *Interspeech 2020*. ISCA, Oct. 2020, pp. 3411–3415. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2020-1854

[15] E. Casanova, A. C. Junior, C. Shulby, F. S. de Oliveira, J. P. Teixeira, M. A. Ponti, and S. M. Aluisio, "End-To-End Speech Synthesis Applied to Brazilian Portuguese," *arXiv:2005.05144 [cs, eess]*, Jul. 2020, arXiv: 2005.05144. [Online]. Available: http://arxiv.org/abs/2005.05144

[16] D. Griffin and Jae Lim, "Signal estimation from modified short-time Fourier transform," in *ICASSP '83. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 8, Apr. 1983, pp. 804–807.

[17] L. McInnes, J. Healy, and J. Melville, "UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction," *arXiv:1802.03426 [cs, stat]*, Sep. 2020, arXiv: 1802.03426. [Online]. Available: http://arxiv.org/abs/1802.03426

11th ISCA Speech Synthesis Workshop (SSW 11)
26-28 August 2021, Budapest, Hungary

Vocal tract area function extraction using ultrasound for articulatory speech synthesis

Debasish Ray Mohapatra[1], *Pramit Saha*[1], *Yadong Liu*[2], *Bryan Gick*[2], *Sidney Fels*[1]

[1]Department of Electrical and Computer Engineering, University of British Columbia, Canada
[2]Department of Linguistics, University of British Columbia, Canada

debasishray@ece.ubc.ca, pramit@ece.ubc.ca

Abstract

This paper studies the feasibility of an articulatory speech synthesizer by extracting the mid-sagittal tongue and palate contours using the ultrasound (US) imaging modality. The extracted contours are then used to compute the vocal tract cross-sectional areas (i.e., area function) during phonation, which then drives an articulary speech synthesizer. Using this approach, we synthesized four phonetic vowel sounds (/a/, /i/, /e/ and /o/). The derived vocal tract (VT) transfer functions are shown to match over multiple utterances for a single vowel, thereby confirming reliable and accurate area function derivation using the US. The acoustic formants of simulated vowels using the proposed method show a modest deviation from the speaker's recorded speech signal since the current articulatory model does not include the mouth radiation mechanism. Furthermore, the higher formants' positions (F5-F8) are approximately equivalent to the high-quality standard MRI-based acoustic results and have an average error of 3.90%, 4.14%, 1.26% and 2.99% for vowel sounds /a/, /i/, /e/ and /o/, respectively. Our approach provides a step towards developing a US-based speech synthesizer for precise extraction of the upper VT geometry and enabling speakers to drive an articulatory model directly by their tongue movements without the necessity of vocalization.

Index Terms: speech synthesis, computational paralinguistics, ultrasound image, silent speech, human-computer interaction

1. Introduction

Speech interfaces follow the sophisticated speech production mechanism for synthesizing voices to aid the speech-handicapped and support communication systems [2, 3, 4]. The articulatory data are of paramount importance for such interfaces, as the speech production task demands unified movements of a set of articulators (e.g., pharynx, tongue, hard palate, lips, etc.). The existing challenges in these speech synthesizers can be highlighted by analyzing their underlying assisting tools (e.g., sensors and imaging devices) and core functionality. Fundamentally, they build upon two principles:

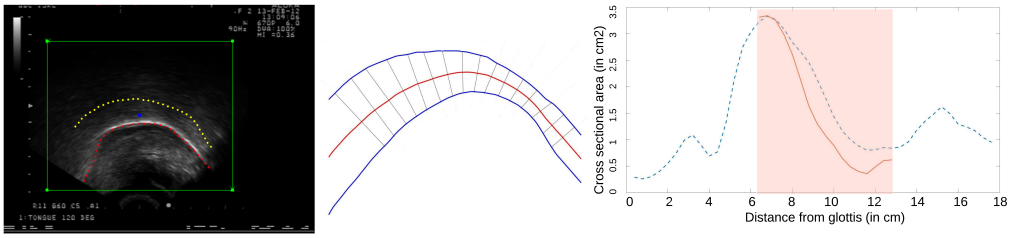
- Approximating the upper VT geometry (i.e., area function) using several invasive or non-invasive sensors and imaging modalities.
- Mapping the VT geometrical information to an acoustic space that can generate speech sounds with precise acoustic characteristics.

The earlier synthesizers took advantage of different optical and magnetic sensing technologies to estimate VT shapes by tracking lip movements [5, 6] and tongue shapes [7, 8]. They consider each articulatory movement as an individual entity and then unify all articulatory information to determine the final VT shape. Though it is hard to determine the influence of an individual articulator, studies [9, 10] show that the tongue is a significant determinant for the upper VT geometry. Due to its agility and higher degrees of freedom, the tongue dynamically creates an irregular tube-like constriction between the larynx and mouth opening, which governs the acoustic features of speech. Hence, the accurate description of the VT shape through its area function is a primary component for an articulatory model, which uses the VT area function to produce synthesized speech. In contrast to other approaches, the area function approximates the variation in VT cross-sections as a function of distance from the glottis and offers a much straightforward geometrical representation of the intricate VT shape.

Lately, the advance in high-quality non-invasive medical imaging techniques has enabled the visualization of tongue movements from different orientations at once. Earlier, the sagittal x-ray projection [10], and computed tomography [11] were used to image the upper VT for various vowel and consonant sounds. Nevertheless, due to radiation risks to the human subject being imaged, these imaging techniques are not clinically safe. Lately, MRI is widely adopted [1, 12] to capture the mid-sagittal upper VT images at any desired orientation without any harmful effects. The captured upper airways are then segmented from surrounding tissues before computing their area functions, which assist in building speaker-specific articulatory models [13, 14, 15]. However, the image acquisition time using MRI is on the order of several minutes, and the method is unsuitable for extracting VT cross-sections during continuous speech production due to the low temporal resolution of rt-MRI. Lim et al. [16] have recently produced VT rt-MRI images of 75 speakers with a high spatio-temporal resolution, though such imaging methods are expensive and not always readily available. Alternatively, the multimodal compact US imaging device allows real-time visualization of the tongue contour with better temporal resolution. There are several models in literature [17, 18, 19, 20, 21, 22, 23] which use US to design silent-speech interfaces. Such a system characterizes tongue shapes, lips and jaw movements [24] as feature vectors instead of area functions and matches their corresponding speech tokens through a mapping algorithm (i.e., articulatory-to-acoustic mapping). The system efficiency relies upon a fixed-size database with a broad set of prerecorded speech tokens at the expense of computational run-time and memory space. Therefore, the computational overhead of the system increases as the acoustic space grows.

This paper proposes a novel US-based articulatory model via tracing a section of the oral cavity during phonation and determining its cross-sectional areas from the US imaging modality. A physics-based articulatory synthesizer (2.5D FDTD VT [13]) then directly utilizes these area functions to produce intel-

10.21437/SSW.2021-16



(a) (b) (c)

Figure 1: *The area function extraction procedure for the vowel /e/. (a) shows the tracked palate (dotted yellow line) and tongue surface (dotted red line) in an ultrasound image of a tongue, (b) shows the extracted vocal tract boundaries (in blue), the estimated midline (in red) and the equidistant grid lines (in black), (c) shows our computed US-based area functions (in red) in comparison to standard MRI based area functions (blue dotted) [1]. The shaded portion shows the vocal tract area region from the base of the tongue to the tip.*

ligible static English vowel sounds without having any dependence on prerecorded speech segments. Therefore, our model does not require extra memory space to accommodate acoustic data. Moreover, as a realistic approach, it only relies upon articulatory information in terms of area function rather than a deep learning-based mapping algorithm to synthesize speech sound. At the core of this method is the extraction of the mid-sagittal cross-sectional areas between tongue and palate using the US, which drives a computationally affordable acoustic wave solver for synthesizing speech sounds. We implement the adaptive grid strategy (AG) [25] that first determines the vocal tract centerline and subsequently extracts its cross-sections perpendicular to the centerline. For the articulatory model, we adopted a 2D acoustic wave solver (i.e., 2.5D FDTD VT) due to its better geometrical flexibility and time complexity than the 1D and 3D articulatory models, respectively. The following section discusses the development of the proposed model in detail.

2. Method

2.1. Ultrasound Imaging & Audio Recording

At present, the coupled articulatory model simulates synthetic speech sounds for a fixed VT geometry. Hence, we performed experimental studies with phonetic vowel sounds for a preliminary assessment of our synthesizer. Evaluation of the articulatory synthesizer with dynamic boundary conditions is being investigated for future work. During this study, a healthy male speaker (26 years old) was asked to repeat each of the following vowel sounds five times with intervals in between: /a/, /i/, /o/ and /e/ — the resting time before every utterance was approximately 10 seconds. We have mainly analyzed the VT shape of cardinal vowels in our study as these vowel sounds represent the extreme points of articulation. However, we did not consider vowel sound /u/ as it involves the extension of VT due to lips protrusion. Since ultrasound only captures tongue contour, the acoustic simulation of vowel /u/ may produce inaccurate results. Nevertheless, this can be addressed in future work by estimating lips position using different sensing technology.

For imaging the tongue and upper palate, we used ALOKA SSD-5000 ultrasound system at 30 fps in conjunction with a 9mm radius UST-9118 3.5 MHz convex ultrasound transducer (120-degree scan angle) and Echo Wave II ultrasound imaging software. Since the articulatory model requires the cross-sectional area profile of VT contour along the mid-sagittal plane, the US transducer was locked into a fixed position and placed beneath the speaker's chin while imaging a single section of the tongue. Usually, the palate is not visible as the US beam reflects off the air in the vocal tract region. However, while swallowing, the tongue surface makes complete contact with the palate, leaving no air in between and thereby allows the imaging of the palate [26]. Since the palate movement is negligible compared to the tongue during speech production tasks, we imaged the speaker's upper palate only once through a swallowing action. Throughout the experiment, we asked the speaker to sit in a chair and put his head supported by firmly attached pads to the chair. The ultrasound probe was held securely by the arm attached to the same chair. Using this approach, we stabilized the speaker's head and eliminated any unwanted palate movements.

Simultaneously, the audio data from the speaker were recorded using a Sennheiser MKH 416 P48 shotgun microphone and a Focusrite Scalet 2i2 preamplifier to enable a comparative study between the original speech signal and the simulated speech output, as illustrated in Section 3.3. The recorded data are available here[1]. We used Audacity, a digital audio editor, to cancel the US ventilation noise from the recorded speech signal. A time lag exists between the audio recording and video of the US imaging of the tongue that needs to be calculated. Therefore, the participant was asked to produce /ka/ 5-7 times at the beginning of the recording session. The release of /k/ involves a burst and a quick tongue movement; hence, it makes both acoustic and articulatory landmarks of release easy to identify and determine the time differences between them.

2.2. Area Function Extraction

As the next step, we traced the tongue shape and palate contour in a semi-automatic fashion from the US video stream and extracted the corresponding cross-sectional areas. The area function usage simplifies acoustic analysis of the vocal tract as it approximates the upper airway as an irregular acoustic tube which allows planar wave propagation. From an acoustic perspective, this means the variation of acoustic pressure remains constant over a planar wavefront, and the wavefronts are perpendicular to the tube centerline (i.e., along the direction of sound wave propagation) in space. It is to be noted that such an assumption limits the characterization of the vocal tract model for higher-order modes. However, as per the assumption, the vocal tract's cross-sections that coincide with planar wavefronts supervise its acoustic characteristic. Thus, we implemented the adaptive grid strategy [25, 27], which first derives the centreline between

[1] https://github.com/Debasishray19/Talking-Tube/tree/SSW11/audioData

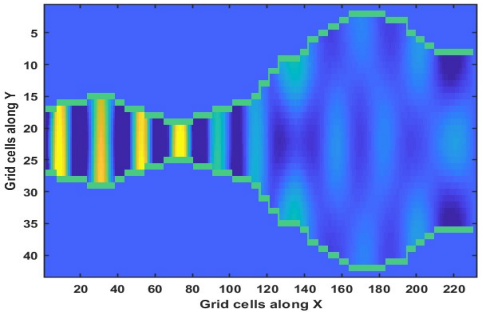


Figure 2: *Illustration of vowel simulation using 2.5D FDTD vocal tract model. The colormap shows the acoustic pressure wave propagation inside the vocal tract.*

the traced tongue and palate contours. Next, it approximates a set of grid planes perpendicular to the centerline for estimating the corresponding area function. Unfortunately, if there is an abrupt VT shape change and the number of grid planes is high, there is a possibility of collision between the grid planes. However, this could be prevented by positioning them at an equal interval while reducing their counts. Our solution procedure is discussed as follows:

1. Trace the tongue and upper palate (see Figure 1a) by fitting a smooth spline to the lower edge of the visual lingual contour using the EdgeTrack system [28]. The EdgeTrack is an automation tool for extracting tongue surfaces from the surrounding noise and unrelated high-contrast edges in ultrasound images.
2. Draw the traced contours on a 2D cartesian coordinate system and sample the coordinates for both the tongue and the upper palate.
3. Derive the centerline coordinates by averaging the ordinates of the tongue and palate corresponding to each abscissa in the set of sample points. Then, connect the derived coordinates using piecewise line segments to determine the centerline.
4. Draw a set of equidistant grid lines perpendicular to the centerline. Each grid line intercepts the posterior-superior palate contour and the anterior-inferior tongue contour (see Figure 1b).
5. Calculate the cross-sectional area of the vocal tract tube from the length of intercepts at the sampled points.
6. Normalize the cross-sectional areas starting from the base to the tip of the tongue to retrieve the 1D area functions between tongue and palate (see Figure 1c).

2.3. 2.5D FDTD Acoustic Wave Solver

We implement a 2.5D wave-guide-based articulatory model [13] to synthesize acoustic outputs. The articulatory model extends the rationale of the 2D finite-difference time-domain (2D FDTD) numerical scheme [14] and improves upon it by lumping wave propagation effects for off-plane waves. Mohapatra and Zappi have already shown the potential of the lightweight 2.5D FDTD VT model as its acoustic features are comparable to a complex 3D FEM (i.e., finite element method) VT model [15] for static vocal tract shapes. Moreover, unlike 1D articulatory models, the 2.5D model captures the effect of non-planar wave propagation inside the VT geometry, which heavily influence speech acoustics, especially higher-order modes that typically appear above 5 kHz [25].

Algorithm 1 FDTD Time-Marching algorithm

Input: VT area function $a(x)$, audio time t

Initialize the physical constants: air density ρ, sound speed c.
Initialize grid size ($M \times N$) and simulation sampling rate R.
Set the temporal resolution (Δt) and grid resolution (Δx) with R and CFL condition.
Set the simulation step size T with t and Δt.
Define boundary cells with $a(x)$ and normal acoustic impedance Z.
Set depth values ($\bar{D}$, $D_{(x)}$ and $D_{(y)}$) for each grid cell derived from $a(x)$.
Define source excitation cells
Initialize source excitation velocity v_e.
Initialize acoustic components (p, v_x and v_y) for each grid cell.
for $n = 1 \ldots T$ **do**
  for $i = 1 \ldots M$ **do**
    for $j = 1 \ldots N$ **do**
      Update $p^{n+1}(i,j)$ with $v_x^n(i,j)$, $v_y^n(i,j)$ and $D(i,j)$ (Eq. 1 and Eq.3)
      if $(i,j) =$ excitation cell **then**
        $v_x^n \leftarrow v_x^n + v_e^n$ and $v_y^n \leftarrow v_y^n + v_e^n$
      end if
      if neighbouring cells = boundary cell **then**
        $v_b^n \leftarrow p^n(i,j)/Z$
      end if
      Update $v_x^{n+1}(i,j)$ with $p^{n+1}(i,j)$ and v_b^n (Eq. 2)
      Update $v_y^{n+1}(i,j)$ with $p^{n+1}(i,j)$ and v_b^n (Eq. 2)
    end for
  end for
end for

Before simulation starts, we first create the 2D VT contour inside a staggered rectangular grid using the area function retrieved from the US and then define the acoustic components (p as pressure, v_x and v_y as velocity along x and y axis) for each grid cell. In a similar fashion, the tube depths($\bar{D}$, $D_{(x)}$ and $D_{(y)}$) are derived and mapped to each grid cell as described here [13] in detail. The spatial resolution (Δs) and temporal resolution (Δt) of the simulation are restricted using the Courant-Friedrichs-Lewy condition($\Delta t \leq \Delta s/\sqrt{2}c$, $c =$ speed of sound). A boundary condition is enforced via $\boldsymbol{v}_b$ similar to what proposed here [14, 29] to include the time-dependent VT wall losses. A source excitation function is coupled at the glottal end to induce the acoustic energy into the VT tube.

During the simulation, the 2D wave solver updates each acoustic component by solving the discretized mixed-form wave Equations (1) and (2) across the entire grid using a time-marching algorithm (see Algorithm 1). The time-domain analysis fits well for the articulatory model, as the vocal tract and articulatory processes evolve with time to produce speech. We denote the standard discrete spatial derivatives with $\widetilde{\boldsymbol{\nabla}}$ as performed in 2D FDTD:

$$p^{(n+1)} = \frac{\bar{D}p^{(n)} - \rho c^2 \Delta t \, \widetilde{\boldsymbol{\nabla}} \cdot \boldsymbol{V}^{(n)}}{\bar{D}} \tag{1}$$

$$\boldsymbol{v}^{(n+1)} = \frac{\beta \boldsymbol{v}^{(n)} - \beta^2 \Delta t \, \widetilde{\nabla} p^{(n+1)}/\rho \; + \; \Delta t(1-\beta)\boldsymbol{v_b}}{\beta + \Delta t(1-\beta)} \tag{2}$$

where,

$$\boldsymbol{V} = \left(D_{(x)} v_x, \; D_{(y)} v_y \right) \tag{3}$$

$\rho =$ air density and $n = n^{th}$ time step of the simulation.
$\beta =$ A scalar parameter to distinguish between air ($\beta = 1$) and the VT boundary ($\beta = 0$) in the computational domain.

3. Experiments and Results

3.1. Experimental Setup

We captured the pressure propagation $p^{(n)}$ (as shown in Figure 2) by placing a virtual microphone 3 mm inside the mouth opening to simulate audio time events of 50 ms. The microphone essentially collects discretized pressure samples at each time step during the simulation. Since the 2.5D FDTD VT model does not yet employ radiation losses, we impose the Dirichlet boundary condition (i.e., open-end boundary condition) at the tube end as proposed here [30]. However, it is to be noted that the mouth radiation is an important dissipation mechanism [31] in determining the actual acoustic characteristics (i.e., formants' positions, bandwidths and amplitudes, voice naturalness, etc.) during speech production. Therefore, we expect some deviations in the acoustic properties between the synthesized vowel sounds and the recorded speech signal. A Gaussian volume velocity pulse $v_e^{(n)}$ with a maximum frequency range up to 10kHz was used as the glottal excitation to compute the VT transfer function and their formants. We followed the transfer function analysis method to extract formants as it is a standard approach to characterize the synthesized audio outputs generated from the computational vocal tract model.

$$v_e^{(n)} = e^{-\{(\Delta t n - T)/0.29T\}^2} (m^3/s) \tag{4}$$

where, $T = 0.646/f_0$ and $f_0 = 10$kHz

However, the articulatory model needs to be coupled with a self-oscillatory vocal fold (VF) model, a source excitation function to generate synthetic speech output. Hence, during the vowel sound production, we coupled the 2.5D vocal tract to a self-oscillating biomechanical vocal fold model (i.e., lumped-element two-mass VF [32]). We chose a biomechanical VF model over the parametric (e.g., [33]) and kinematic models (e.g., [34]), as it simulates the flow-induced oscillations of the vocal folds. Thereby, it naturally reproduces effects that result from the interaction between the vocal folds, the glottal flow, and the vocal tract. In the past, multiple articulatory models have been successfully demonstrated using the two-mass VF model [35, 36]. The two-mass VF model is computationally lightweight, and it uses two point-masses connected by a spring-damper system to emulate the self-oscillatory characteristics of the human vocal folds.

A numerical simulation using the finite-difference scheme was then carried out, setting the physical constants as follows, speed of sound of $c = 350$m/s and air density of $\rho = 1.14$ kg/m^3. We fixed the sampling rate of the simulation to $661,500$ Hz, thereby having the temporal resolution of $\Delta t = 1.51 \times 10^{-6}$s. We lowpass the output pressure samples above 22kHz using a second-order Butterworth filter to generate vowel sounds. The feed-forward and feedback coefficients of the filter are $0.0754 \times \{1, 2, 1\}$ and $\{-1.0875, 0.3890\}$, respectively. We implemented the VT area functions extraction algorithm and the physics-based articulatory model in the MATLAB environment. The custom code of our model is publicly available here[2].

3.2. Speech Output Evaluation

Our current model supports investigation with different speech sounds by directly modifying the VT tube geometry, hence controlling the acoustic output. Though several acoustic features (e.g., formant position and bandwidth, pitch, etc.) contribute to the simulated-voice quality, we solely focus on formant positions as they play a vital role in distinguishing the vowel categories. First, we characterize the acoustic behaviour (i.e., transfer function curves) of a simulated vowel sound, phonated at separate time instants (i.e., three different trials) by the same speaker to evaluate the reliability of the US imaging modality for vocal tract area function extraction. Second, we extract the formants [37] from the recorded speech signals and compare them against the US-based VT area functions acoustic simulation. Next, we analyze the higher formants' positions (F5-F8) using the proposed method as higher formants in VT frequency response contribute to the perception of voice quality [38]. For this, we match the results against a standard high-quality MRI dataset [1]. Since the US does not capture the complete VT cross-sectional areas, we generate a new area function dataset, called *US+MRI*, to study the impact of missing geometry. The new area function dataset includes the tongue contour extracted from the US and supplements the absent area functions using the standard MRI data directly. Currently, we do not have the VT area function dataset using the MRI and US imaging techniques for the same subject. However, the human upper airway geometry approximately remains identical for vowel sounds across different subjects. Therefore we directly use Story's MRI data [1] of VT cross-sections to compensate for the missing geometrical information.

As discussed earlier, this study does not characterize the acoustic properties of speech sounds for dynamic vocal tract shapes. The dynamic boundary condition requires attention to other additional details such as continuous extraction of vocal tract cross-sections in a fixed-time interval as it changes, a smooth transition of vocal tract walls in the FDTD computational domain [39], etc. Therefore, it is essential that we first characterize our US-based articulatory model for static vocal tract shapes.

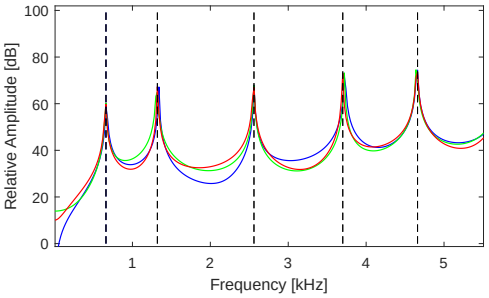


Figure 3: *Illustration of spectral envelops for vowel /a/ and their formants (dotted vertical line), derived from the simulation of the US-based area functions samples for three different trials - Trial1 (Red), Trial2(Green), Trial3 (Blue).*

3.3. Results

The VT transfer function $H(f)$ was obtained by applying the fast Fourier transform (FFT) to the output-pressure samples p_0 and Gaussian volume velocity pulse v_e as follows,

$$H(f) = \frac{p(f)}{v_e(f)} \tag{5}$$

where $p(f)$ and $v_e(f)$, respectively stands for the FFT of $p^{(n)}$ and $v_e^{(n)}$. Figure 3 shows three nearly equivalent transfer func-

[2] https://github.com/Debasishray19/Talking-Tube/tree/SSW11

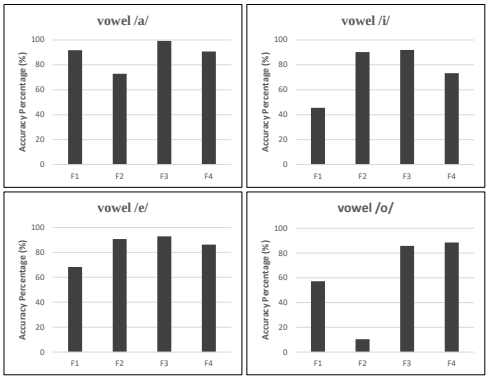


Figure 4: *Accuracy percentage of the first 4 formants in the US-based area function simulation for vowels /a/, /i/, /e/ and /o/, with respect to the speaker's original speech signal.*

Table 1: *Absolute positional errors (in percentage) of the higher formants (F5-F8) with only US (denoted as U) vs US+MRI (denoted as U+M) based area function simulation*

V	F5		F6		F7		F8	
	U	U+M	U	U+M	U	U+M	U	U+M
/a/	6.4	0.4	1.4	0.7	5.9	1.1	1.8	0.5
/i/	3.1	4.7	0.5	0	7.4	4.9	5.4	5.9
/e/	2.0	3.6	0.3	0	0.9	2.4	1.7	2.3
/o/	6.3	0.4	3.6	0.7	0.3	0	1.6	1.3

tions having similar formants, derived from the simulation of vowel /a/ with different US-based VT area functions samples.

The speaker's original vowel utterances were considered ground truth to analyze the acoustic features of the model. We used the PRAAT application [40] to approximate the first four formants for each vowel sound from its recorded speech waveform. Figure 4 demonstrates that the accuracy percentage of the first four formants for static vocal tract shapes generated using the US-based area functions and synthesized using 2.5D VT model, is above 80% in most cases.

The PRAAT application can not identify higher-order formant frequencies accurately. Hence, we used Story's VT area function [1] to simulate and retrieve the higher formants (F5-F8), and considered them as baseline for a comparative study. This MRI dataset has been widely used for the acoustic analysis of many articulatory models [14, 30]. The comparative study shows that the average absolute positional errors of these formants were 3.90%, 4.14%, 1.26% and 2.99% for vowel /a/, /i/, /e/ and /o/, respectively (Table 1). However, there was a significant improvement in the formant position with the *US+MRI* area functions, which compensate for the missing cross-sections in the US-based VT areas. The synthesized vowel sounds using different VT area functions are provided here[1] as audio files.

4. Discussion and Conclusion

The VT transfer function for a specific phoneme is unique, subject-specific and depends upon area functions. Since the derived transfer functions for vowel /a/ across multiple samples remain consistent, it is evident that the US can be used to extract VT cross-sections accurately by restricting subjects' head movements while imaging. Besides a few exceptions (e.g., F2 in vowel /o/), most formants resulting from VT acoustic simulation using US images match the speaker's recorded speech signal well. However, the inclusion of free radiation effects into the articulatory model will allow an objective analysis of the existing errors. In order to do that, we have to include a simplified head geometry to the existing VT model and allow the outward pressure wave propagation at the mouth end to emulate free space[41]. This feature will also offer realistic simulation of a VT shapes extracted with US imaging techniques.

As an imaging device, the US is suitable for tracing tongue shape from its blade to root, thereby allowing the accurate representation of a section of the VT shape as an area function inventory. This approach is incapable of estimating the complexity of realistic VT geometries (e.g., piriform fossae, subglottal tract, and lip's area). Though such a level of geometrical details might not be necessary for categorizing vowel sounds, they have significance at the higher end of the speech spectrum. Therefore, the simulation output of *US+MRI* VT area function provides better acoustic results for the higher formants. As a future research direction, it motivates an investigation into the possible ways of augmenting US-based area functions to generate rich VT geometrical information.

The results from this study provide insights into the future development of a silent-speech interface. This could be achieved by retrieving VT area functions using the US and other sensing technologies and passing them to the articulatory speech synthesizer to simulate synthetic speech sounds. The current off-line platforms: (1)VT cross-sectional area function extraction from the US (2) 2.5D FDTD articulatory speech synthesis model, needs to be connected to generate acoustic output in real-time. We are currently working towards implementing the dynamic VT geometry in the articulatory model, which will finally allow the synthesis of continuous speech.

5. Acknowledgements

This work is supported by the Natural Sciences and Engineering Research Council (NSERC; STPGP 506576-17) of Canada and Canadian Institutes for Health Research (CIHR) .

6. References

[1] B. H. Story, "Comparison of magnetic resonance imaging-based vocal tract area functions obtained from the same speaker in 1994 and 2002," *The Journal of the Acoustical Society of America*, vol. 123, no. 1, pp. 327–335, 2008.

[2] B. Denby, T. Schultz, K. Honda, T. Hueber, J. M. Gilbert, and J. S. Brumberg, "Silent speech interfaces," *Speech Communication*, vol. 52, no. 4, pp. 270–287, 2010.

[3] J. Gonzalez Lopez, L. A. Cheah, P. D. Green, J. M. Gilbert, S. R. Ell, R. K. Moore, and E. Holdsworth, "Evaluation of a silent speech interface based on magnetic sensing and deep learning for a phonetically rich vocabulary," in *Proceedings of the Annual Conference of the International Speech Communication Association, Interspeech*. ISCA, 2017, pp. 3986–3990.

[4] G. S. Meltzner, J. T. Heaton, Y. Deng, G. De Luca, S. H. Roy, and J. C. Kline, "Silent speech recognition as an alternative communication device for persons with laryngectomy," *IEEE/ACM transactions on audio, speech, and language processing*, vol. 25, no. 12, pp. 2386–2398, 2017.

[5] H. Nakamura, "Method of recognizing speech using a lip image," Sep. 6 1988, uS Patent 4,769,845.

[6] T. Hasegawa and K. Ohtani, "Oral image to voice converter-image input microphone," in *Proceedings Singapore ICCS/ISITA92*. IEEE, 1992, pp. 617–620.

[7] J. S. Perkell, M. H. Cohen, M. A. Svirsky, M. L. Matthies, I. Garabieta, and M. T. Jackson, "Electromagnetic midsagittal articulometer systems for transducing speech articulatory movements," *The Journal of the Acoustical Society of America*, vol. 92, no. 6, pp. 3078–3096, 1992.

[8] M. J. Fagan, S. R. Ell, J. M. Gilbert, E. Sarrazin, and P. M. Chapman, "Development of a (silent) speech recognition system for patients following laryngectomy," *Medical engineering & physics*, vol. 30, no. 4, pp. 419–425, 2008.

[9] J. Lee, S. Shaiman, and G. Weismer, "Relationship between tongue positions and formant frequencies in female speakers," *The Journal of the Acoustical Society of America*, vol. 139, no. 1, pp. 426–440, 2016.

[10] G. Fant, "Acoustic theory of speech production, s'-gravenhage," *Mouton and Co*, 1960.

[11] C. Johansson, J. Sundberg, and H. Wilbrand, "From sagittal distance to area. a study of pharyngeal cross-sectional area of computer tomography," *The Journal of the Acoustical Society of America*, vol. 75, no. S1, pp. S22–S22, 1984.

[12] H. Takemoto, K. Honda, S. Masaki, Y. Shimada, and I. Fujimoto, "Measurement of temporal changes in vocal tract area function from 3d cine-mri data," *The Journal of the Acoustical Society of America*, vol. 119, no. 2, pp. 1037–1049, 2006.

[13] D. R. Mohapatra, V. Zappi, and S. Fels, "An extended two-dimensional vocal tract model for fast acoustic simulation of single-axis symmetric three-dimensional tubes," *Proc. Interspeech 2019*, pp. 3760–3764, 2019.

[14] V. Zappi, A. Vasuvedan, A. Allen, N. Raghuvanshi, and S. Fels, "Towards real-time two-dimensional wave propagation for articulatory speech synthesis," in *Proceedings of Meetings on Acoustics 171ASA*, vol. 26, no. 1. Acoustical Society of America, 2016, p. 045005.

[15] M. Arnela Coll, "Numerical production of vowels and diphthongs using finite element methods," Ph.D. dissertation, Universitat Ramon Llull, 2015.

[16] Y. Lim, A. Toutios, Y. Bliesener, Y. Tian, S. G. Lingala, C. Vaz, T. Sorensen, M. Oh, S. Harper, W. Chen *et al.*, "A multispeaker dataset of raw and reconstructed speech production real-time mri video and 3d volumetric images," *arXiv preprint arXiv:2102.07896*, 2021.

[17] B. Denby and M. Stone, "Speech synthesis from real time ultrasound images of the tongue," in *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1. IEEE, 2004, pp. I–685.

[18] B. Denby, Y. Oussar, G. Dreyfus, and M. Stone, "Prospects for a silent speech interface using ultrasound imaging," in *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, vol. 1. IEEE, 2006, pp. I–I.

[19] T. Hueber, G. Aversano, G. Cholle, B. Denby, G. Dreyfus, Y. Oussar, P. Roussel, and M. Stone, "Eigentongue feature extraction for an ultrasound-based silent speech interface," in *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07*, vol. 1. IEEE, 2007, pp. I–1245.

[20] T. Hueber, E.-L. Benaroya, G. Chollet, B. Denby, G. Dreyfus, and M. Stone, "Development of a silent speech interface driven by ultrasound and optical images of the tongue and lips," *Speech Communication*, vol. 52, no. 4, pp. 288–300, 2010.

[21] P. Saha, Y. Liu, B. Gick, and S. Fels, "Ultra2speech-a deep learning framework for formant frequency estimation and tracking from ultrasound tongue images," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2020, pp. 473–482.

[22] T. G. Csapó, T. Grósz, G. Gosztolya, L. Tóth, and A. Markó, "Dnn-based ultrasound-to-speech conversion for a silent speech interface," 2017.

[23] N. Kimura, M. Kono, and J. Rekimoto, "Sottovoce: an ultrasound imaging-based silent speech interaction using deep neural networks," in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 2019, pp. 1–11.

[24] T. Hueber and G. Bailly, "Statistical conversion of silent articulation into audible speech using full-covariance hmm," *Computer Speech & Language*, vol. 36, pp. 274–293, 2016.

[25] M. Arnela, S. Dabbaghchian, R. Blandin, O. Guasch, O. Engwall, A. Van Hirtum, and X. Pelorson, "Influence of vocal tract geometry simplifications on the numerical simulation of vowel sounds," *The Journal of the Acoustical Society of America*, vol. 140, no. 3, pp. 1707–1718, 2016.

[26] M. A. Epstein and M. Stone, "The tongue stops here: Ultrasound imaging of the palate," *The Journal of the Acoustical Society of America*, vol. 118, no. 4, pp. 2128–2131, 2005.

[27] M. Arnela, S. Dabbaghchian, O. Guasch, and O. Engwall, "A semi-polar grid strategy for the three-dimensional finite element simulation of vowel-vowel sequences," in *18th Annual Conference of the International Speech Communication Association, INTERSPEECH 2017, Stockholm, Sweden, 20 August 2017 through 24 August 2017*, vol. 2017. The International Speech Communication Association (ISCA), 2017, pp. 3477–3481.

[28] M. Li, C. Kambhamettu, and M. Stone, "Automatic contour tracking in ultrasound images," *Clinical linguistics & phonetics*, vol. 19, no. 6-7, pp. 545–554, 2005.

[29] H. Takemoto, P. Mokhtari, and T. Kitamura, "Acoustic analysis of the vocal tract during vowel production by finite-difference time-domain method," *The Journal of the Acoustical Society of America*, vol. 128, no. 6, pp. 3724–3738, 2010.

[30] M. Arnela and O. Guasch, "Two-dimensional vocal tracts with three-dimensional behavior in the numerical generation of vowels," *The Journal of the Acoustical Society of America*, vol. 135, no. 1, pp. 369–379, 2014.

[31] M. Arnela, O. Guasch, and F. Alías, "Effects of head geometry simplifications on acoustic radiation of vowel sounds based on time-domain finite-element simulations," *The Journal of the Acoustical Society of America*, vol. 134, no. 4, pp. 2946–2954, 2013.

[32] K. Ishizaka and J. L. Flanagan, "Synthesis of voiced sounds from a two-mass model of the vocal cords," *Bell system technical journal*, vol. 51, no. 6, pp. 1233–1268, 1972.

[33] G. Fant, J. Liljencrants, and Q.-g. Lin, "A four-parameter model of glottal flow," *STL-QPSR*, vol. 4, no. 1985, pp. 1–13, 1985.

[34] I. R. Titze, "A four-parameter model of the glottis and vocal fold contact area," *Speech communication*, vol. 8, no. 3, pp. 191–201, 1989.

[35] M. Sondhi and J. Schroeter, "A hybrid time-frequency domain articulatory speech synthesizer," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 35, no. 7, pp. 955–967, 1987.

[36] K. van den Doel and U. M. Ascher, "Real-time numerical solution of webster's equation on a nonuniform grid," *IEEE transactions on audio, speech, and language processing*, vol. 16, no. 6, pp. 1163–1172, 2008.

[37] T. Vampola, J. Horáček, A.-M. Laukkanen, and J. G. Švec, "Human vocal tract resonances and the corresponding mode shapes investigated by three-dimensional finite-element modelling based on ct measurement," *Logopedics Phoniatrics Vocology*, vol. 40, no. 1, pp. 14–23, 2015.

[38] B. B. Monson, A. J. Lotto, and S. Ternström, "Detection of high-frequency energy changes in sustained vowels produced by singers," *The Journal of the Acoustical Society of America*, vol. 129, no. 4, pp. 2263–2268, 2011.

[39] M. Arnela, S. Dabbaghchian, O. Guasch, and O. Engwall, "Mri-based vocal tract representations for the three-dimensional finite element synthesis of diphthongs," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 12, pp. 2173–2182, 2019.

[40] P. Boersma, "Praat: doing phonetics by computer [computer program]," *http://www. praat. org/*, 2011.

[41] D. R. Mohapatra, V. Zappi, and S. Fels, "A comparative study of two-dimensional vocal tract acoustic modeling based on finite-difference time-domain methods," *arXiv preprint arXiv:2102.04588*, 2021.

11th ISCA Speech Synthesis Workshop (SSW 11)
26-28 August 2021, Budapest, Hungary

Non-Autoregressive TTS with Explicit Duration Modelling for Low-Resource Highly Expressive Speech

Raahil Shah[*]*, Kamil Pokora*[*]*, Abdelhamid Ezzerg, Viacheslav Klimkov, Goeric Huybrechts, Bartosz Putrycz, Daniel Korzekwa, Thomas Merritt*

Amazon Text-to-Speech Research

{raahshah, kamipoko}@amazon.com

Abstract

Whilst recent neural text-to-speech (TTS) approaches produce high-quality speech, they typically require a large amount of recordings from the target speaker. In previous work [1], a 3-step method was proposed to generate high-quality TTS while greatly reducing the amount of data required for training. However, we have observed a ceiling effect in the level of naturalness achievable for highly expressive voices when using this approach. In this paper, we present a method for building highly expressive TTS voices with as little as 15 minutes of speech data from the target speaker. Compared to the current state-of-the-art approach, our proposed improvements close the gap to recordings by 23.3% for naturalness of speech and by 16.3% for speaker similarity. Further, we match the naturalness and speaker similarity of a Tacotron2-based full-data ($\approx$ 10 hours) model using only 15 minutes of target speaker data, whereas with 30 minutes or more, we significantly outperform it. The following improvements are proposed: 1) changing from an autoregressive, attention-based TTS model to a non-autoregressive model replacing attention with an external duration model and 2) an additional Conditional Generative Adversarial Network (cGAN) based fine-tuning step.

Index Terms: Text-to-speech, low-resource, expressive speech

1. Introduction

Recent advancements in the TTS domain have demonstrated highly natural speech generated by neural text-to-speech (NTTS) models [2, 3, 4, 5]. However, these models often require large amounts ($\approx$ 10 hours) of recordings [6] to achieve high levels of naturalness without degradation.

Data collection for TTS is an expensive and time-consuming task. The problem is magnified for highly expressive voices, because it requires higher vocal effort from the voice talent as compared to neutral speech. This amplifies the need for a scalable solution to be able to build highly expressive voices with smaller amounts of data and without substantial cost (i.e. low-resource TTS).

Previous research around low-resource TTS attempts to address this problem with multi-speaker modelling and transfer learning. Transferring knowledge from full-resource speakers to a low-resource one improves the synthesis quality of the low-resource speaker [6, 7, 8, 9, 10, 11, 12].

Recent work in Huybrechts et al. [1] brings significant improvements to naturalness by combining multi-speaker modelling with data augmentation for the low-resource speaker. This approach uses a Voice Conversion (VC) model [13, 14, 15, 16, 17] to transform speech from one speaker to sound like speech from another, while preserving the content and prosody of the source speaker. This artificially boosts the training data available for the resource-scarce target speaker by leveraging readily available source speaker data. However, we have observed that this solution does not scale to achieve naturalness on par with a full-data model for more expressive voices than those presented in [1].

To address this limitation, we investigate the most expressive voice in our catalog and propose changes to the model architecture that consistently outperform the approach presented in [1] and achieve naturalness on par or better than a full-data Tacotron2-based [2] model.

First, we propose to switch from a Tacotron2-based (autoregressive) TTS model to a non-autoregressive mel-spectrogram prediction model and to replace the attention mechanism in Tacotron2 with an external duration model. To the authors' knowledge, this work is the first to investigate such NTTS architectures in a reduced data scenario. In the literature, so far mainly attention-based or autoregressive models have been explored in the context of expressive low-resource TTS [7, 18, 19, 8]. Such models suffer from stability issues exhibited in synthesised speech, such as babbling, early cut-off, word repetition, and word skipping [20, 21, 22, 23]. These problems, attributed to teacher-forcing and attention, are even more prevalent in the reduced data scenario. Recent research in the field [24, 25, 26], inspired by traditional parametric speech synthesis [27, 28] mitigates these issues by explicitly modelling the durations of phonemes. In addition to improving speech stability, we posit that explicit duration modelling significantly improves the overall naturalness of highly expressive voices by making it easier to model variability in phoneme durations than in the baseline attention-based systems.

Second, we investigate an application of Conditional Generative Adversarial Networks (cGAN) [29] as an additional fine-tuning step aimed at improving the signal quality of low-resource synthesis. The less data we have, the harder it is to maintain good segmental quality and speaker similarity. GANs [30], known for generating high quality images, have also been applied in the speech domain to improve the segmental quality of predicted mel-spectrograms [31, 32]. We extend the standard GAN recipe to pass conditioning in addition to the typical mel-spectrogram input to the discriminator. This better informs the discriminator network when making a classification, allowing for more insightful information to flow to the generator.

2. Proposed Method

As in Huybrechts et al. [1], the method presented in this paper is based on three main steps: 1) data augmentation, 2) multi-speaker TTS and 3) fine-tuning. In this work, we also investigate the addition of a fourth step where we fine-tune the model with a cGAN approach to further improve the audio quality.

[*]The first two authors have equal contribution.

10.21437/SSW.2021-17

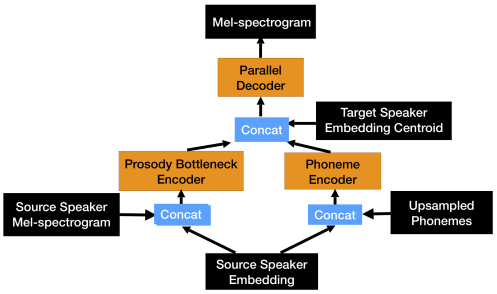


Figure 1: *Schematic diagram of the voice conversion model used in Step 1 of the proposed method.*

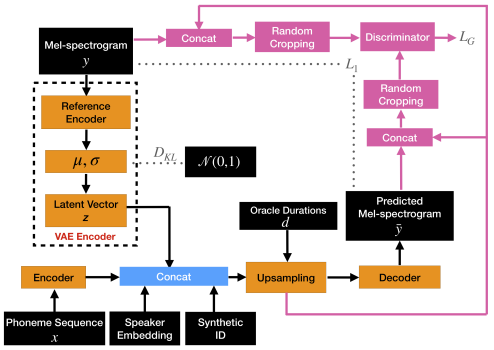


Figure 2: *Schematic diagram of the acoustic model used in Steps 2-4 of the proposed method. Components in pink are used only in Step 4 of the method.*

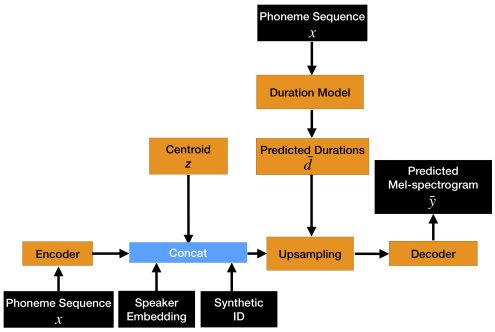


Figure 3: *Schematic diagram of the acoustic model during inference.*

Our full proposed low-resource TTS methodology is defined as follows:

1. Train a VC model to augment data for the target speaker.
2. Train a multi-speaker TTS model using recordings and synthetic data created in Step 1.
3. Fine-tune the TTS model with the recordings from the target speaker.
4. Fine-tune the TTS model with the cGAN approach.

The key contribution of this work is the change in TTS architecture from a Tacotron2-style attention-based model to a non-autoregressive acoustic model supported by external durations. The resulting TTS model is comprised of two main components: 1) an acoustic model that predicts mel-spectrogram $\tilde{y}$ from a phoneme sequence x, 2) a duration model which assists the acoustic model during inference by providing the duration $\tilde{d}$ of each phoneme. During training, ground truth durations d are used by the acoustic model. As in Huybrechts et al. [1], a Parallel WaveNet universal neural vocoder [33] is used to obtain the final speech signal from the generated mel-spectrogram.

2.1. Voice Conversion Model

A voice conversion model is used to perform data augmentation in Step 1 of the method. This model converts the speaker identity of a source audio to sound as though it was spoken by the target speaker.

As in Huybrechts et al. [1], we use the CopyCat [17] architecture for this model which is presented in Figure 1. The model consists of: 1) a phoneme encoder that learns latent representations from phonemes, 2) a prosody bottleneck encoder which disentangles prosody from the reference mel-spectrogram and 3) a parallel decoder which generates the mel-spectrogram given the phoneme and prosody bottleneck encoder's outputs, in addition to the target speaker embedding.

We follow the approach in Huybrechts et al. [1] to modify the original CopyCat model by concatenating speaker embeddings to the upsampled phonemes before feeding this to the phoneme encoder. This was found to help reduce occurrences of speaker leakage in [1].

The VC model was trained with 18 supporting speakers who were recorded in a conversational speaking style, in addition to the target speaker. For the highly expressive target speaker investigated in this paper, fine-tuning of the Copycat model was required to prevent issues with speaker leakage, unlike in Huybrechts et al. [1]. The model was trained on the data from all speakers for 50 k steps and then only on the target speaker's data for an additional 320 epochs. We hypothesise that this fine-tuning is required because the target speaker's data is much more expressive than that of the supporting speakers.

2.2. Acoustic Model

We use an acoustic model in Step 2-4 of the method, as illustrated in Figure 2. The topology of this model during inference is presented in Figure 3.

2.2.1. Encoder

Our encoder architecture is the same as that presented in Tacotron2 [2]. It is comprised of an embedding lookup followed by 3 convolution blocks each with a kernel size of 3. On top of that we apply a single bi-directional LSTM layer with a hidden dimension of 512 and a dropout of 0.1. We pass the phoneme sequence x as input to this encoder, to obtain phoneme embeddings $\tilde{x}$.

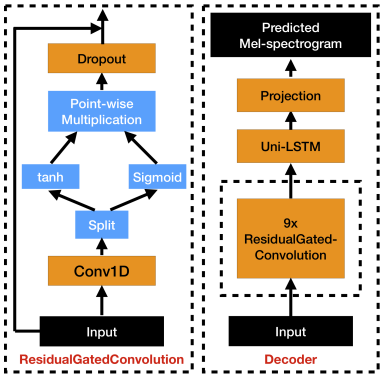


Figure 4: *Illustration of a residual gated convolution block and the decoder architecture.*

2.2.2. *Variational Autoencoder (VAE)*

TTS is a one-to-many problem as the same text can be spoken in many different, yet acceptable, ways. In autoregressive NTTS models, this effect is mitigated both by teacher-forcing as well as by conditioning on the latent acoustic representation obtained from a VAE [34]. In the proposed non-autoregressive architecture, we use only the VAE to pass information which cannot be inferred solely from the input phoneme sequence.

This encoder takes mel-spectrogram frames as input. It comprises 6 convolution blocks each with a kernel size of 5, followed by one GRU layer with a hidden dimension of 128. We take the last output from the GRU and perform a projection to 128 dimensions, in order to parametrise the posterior distribution. The first half of this output represents μ while the second half represents σ. Finally, we sample from the posterior distribution to obtain a final latent representation z. At inference time, we use a pre-calculated centroid of zs obtained from the available ground truth data for the target speaker.

2.2.3. *Upsampling and Additional Embeddings*

To each phoneme embedding we concatenate: 1) the latent z vector, 2) a speaker embedding obtained from a pre-trained GE2E-based [35] speaker verification model and 3) a one-hot 'synthetic ID' flag, indicating whether the data is ground truth or obtained from voice conversion. Then we upsample each phoneme embedding according to ground truth durations d (training-time) or predicted durations $\hat{d}$ (inference-time).

Similar to Parallel Tacotron [25], before passing these upsampled embeddings to the decoder, we provide positional information to indicate the relative position of a frame inside a phoneme. To each embedding we concatenate: 1) a transformer-style positional embedding [36] indicating the phoneme duration, 2) a transformer-style positional embedding indicating the frame's position inside a phoneme and 3) the fractional progress of the frame in a phoneme.

2.2.4. *Decoder*

The embedding sequence output from the upsampling component is passed as input to the decoder. The modelling task of the decoder was found to require local context in [25], therefore our decoder is comprised of 9 residual gated convolution layers. Each residual gated convolution block is composed of a 1D-convolution with kernel size 15 and a hidden dimension of 512, followed by a $\tanh$ filter and sigmoid activation gate which are element-wise multiplied and then added to a residual connection after a dropout of 0.1.

The convolution stack is followed by 2 uni-directional LSTM layers with a hidden dimension of 512 and a dropout of 0.1. Preliminary evaluations showed that this final LSTM stack improves audio quality. A schema of the decoder as well as the residual gated convolution architecture is presented in Figure 4.

2.2.5. *Conditional GAN Fine-Tuning*

GANs are a well established solution to the problem of 'over-smoothing' encountered during the optimisation of L1/L2 loss functions. With mel-spectrogram prediction, this effect manifests as lower brightness and poorer audio quality in the subjective perception of the speech signal.

Adversarial training of the acoustic model can be utilised as a fine-tuning step to mitigate such degradations [31, 32]. Typically, such an adversarial training involves only the mel-spectrogram being passed as input to the discriminator network. We explore an extension to this setup (cGAN), wherein we condition the discriminator on both acoustic and linguistic information. Additional conditioning allows for more meaningful gradient flow from discriminator to generator, which has been shown to improve adversarial training [29].

The entire acoustic model acts as the generator network. For the discriminator network we used the architecture presented in SAGAN [37]. As input to the discriminator we feed randomly cropped 64 frame chunks of the generated mel-spectrogram, along with the embeddings $\tilde{x}$ of the corresponding phoneme sequence and the latent acoustic information z from the VAE. Cropping was found to be more effective than feeding the whole mel-spectrogram. We hypothesise that this is because the goal of the fine-tuning step is to improve the segmental quality of the final mel-spectrogram, which is a more local, time-invariant task.

2.2.6. *Training Setup*

To train the acoustic model we use the Adam optimiser [38] with $\beta_1 = 0.9$ and $\beta_2 = 0.98$. We use a linear warm-up of the learning rate from 0.1 to 1 for the first 10 k steps, followed by an exponential decay from 10 k steps to 100 k steps with a minimum value of 10^{-5}.

In Step 2 of the method, the acoustic model is trained for 500 k steps with a mini-batch size of 32. The model is trained on both ground truth and synthetic data for our target speaker as well as data from supporting speakers, using the following loss function:

$$L_{Train} = L_1 + \gamma * D_{KL} \quad (1)$$

where L_1 is the L_1-distance between predicted and ground truth mel-spectrogram and D_{KL} is the Kullback–Leibler divergence between the VAE posterior distribution and $\mathcal{N}(0, 1)$. To avoid the collapse of D_{KL}, we used the same KL annealing scheme as presented in [39].

In Step 3 of the method, we fine-tune the model for an additional 30 k training steps, using only ground truth data from the target speaker, still optimising L_{Train}.

Finally, in Step 4 of the method, we freeze all VAE weights and fine-tune the acoustic model with the cGAN setup for an additional 30 k steps, also using only ground truth target speaker

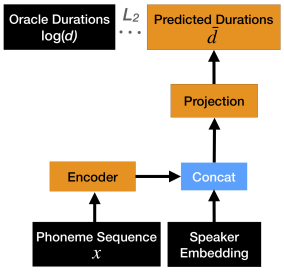


Figure 5: *Schematic diagram of the duration model used in Step 2 of the proposed method.*

data. During this step, the following generator loss (L_G) and Hinge discriminator loss (L_D) functions are used:

$$L_G = \mathbb{E}_{x,y\sim p_{data}}[D(y,\tilde{x},V(y)) - D(G(x,y),\tilde{x},V(y))] \quad (2)$$

$$L_D = \mathbb{E}_{x,y\sim p_{data}}[\text{ReLU}(1 + D(G(x,y),\tilde{x},V(y))) + \text{ReLU}(1 - D(y,\tilde{x},V(y)))] \quad (3)$$

where D is the discriminator network, G is the generator network (acoustic model) and V denotes the VAE. The discriminator is trained by optimising L_D, while the acoustic model is fine-tuned by optimising the total loss:

$$L_{GANFineTune} = L_1 + \alpha * L_G \quad (4)$$

2.3. Duration Model

We train a duration model in Step 2 of the method, whose architecture is presented in Figure 5. We model phoneme durations as the integer number of mel-spectrogram frames corresponding to each phoneme. We assume that ground truth phoneme durations are provided by an external aligner, such as the Gaussian Mixture Model (GMM) based Kaldi Speech Recognition Toolkit [40] used in our experiments.

To model the duration sequence, we first pass the phoneme sequence through an encoder and then apply a dense projection to 1 dimension followed by a ReLU activation function. During training, teacher-forcing is used i.e. only ground truth durations are input to the acoustic model, while predicted durations are used only at inference-time.

For the proposed multi-speaker acoustic model using reduced target speaker data, we train the duration model separate from the acoustic model. This model uses a phoneme encoder identical to the one described in Section 2.2.1, with the hidden dimension reduced to 256, whose output is concatenated with pre-trained speaker embeddings after they are passed through an affine layer. The training objective for this model is a L2 loss on durations in the log domain and it is trained for 150 k steps (with a mini-batch size of 32). We use an identical Adam optimiser configuration as that used for the acoustic model training.

For the full-data anchor models trained on a single-speaker, the embedding sequence used for duration prediction is the concatenation of the phoneme embeddings $\tilde{x}$ and the latent vector z from the acoustic model. In this setup, in line with the state-of-the-art [24, 25, 26], the two models are trained jointly, by adding an auxiliary L1 loss between ground truth and predicted durations to the total objective, with a weighting of 0.025.

Preliminary evaluations showed that separately training the duration model in the low-resource multi-speaker scenario performed better than the joint training of acoustic and duration models used for full-data single-speaker models.

3. Experiments

3.1. Data

For the 'highly expressive' target speaker, we selected the voice objectively identified as the most expressive speaker in our internal American English voice catalog. Expressivity was measured as variation along the three axes of frequency, power and durations by analysing respectively the mean and variance of static $\log f0$, $mgc0$ and phoneme duration features and their deltas, for each speaker.

In the 'full-data' (FD) setup we used $\approx$ 10 hours of recorded speech from the target speaker. In the low-resource aka 'data reduction' (DR) scenario, we investigated four different reduced data amounts: 3 hours, 1 hour, 30 minutes and 15 minutes of target speech. To perform data augmentation as detailed in Section 2.1, we supplemented the target speaker data in each DR scenario with 4.5 hours of synthetic data converted from the full-data of a single supporting speaker, by a VC model trained using the respective reduced target data amount for that scenario.

For supporting speakers in our multi-speaker models, we used an internal American English dataset comprising 18 speakers recorded in a conversational style. This dataset contained $\approx$ 65 hours of speech.

3.2. Evaluation

In each DR scenario, we evaluated our models by conducting MUSHRA tests [41] on the following two metrics:

- Naturalness – "Please rate the audio samples in terms of their naturalness".
- Speaker Similarity – "Please listen to the speaker in the reference sample first. Then rate how similar the speakers in each system sound compared to the reference speaker."

Each test was conducted independently, by 20 listeners, each evaluating 61 MUSHRA screens synthesised from a fixed test set of 61 held-out samples. To check for statistical significance, we performed paired t-tests using the Holm-Bonferroni correction method. All statistical differences presented are for $p \leq 0.05$.

3.3. GAN Fine-Tuning Study

We conducted a supporting study to investigate improvements from GAN fine-tuning on the naturalness of synthesised speech, demonstrating the impact of Step 4 of the proposed method. In this study, we evaluated the following four systems in a 3 hours DR scenario:

- *(Recordings)* Ground truth recordings.
- *(DR No-att)* Baseline without any GAN fine-tuning (i.e. Steps 1-3 of the proposed method).
- *(DR No-att + GAN)* Candidate system with additional fine-tuning using vanilla GAN (i.e. only mel-spectrogram input to discriminator).
- *(DR No-att + cGAN)* Candidate system with additional fine-tuning using Conditional GAN (i.e. conditioned on

the phoneme sequence, acoustic and prosody information).

Target Data	3 h
Naturalness	
Recordings	88.94
DR No-att	64.45
DR No-att + GAN	64.08
DR No-att +cGAN	66.57

Table 1: *Average MUSHRA scores for naturalness, showing the impact of Conditional GAN fine-tuning.*

As shown in Table 1, cGAN fine-tuning provides a statistically significant improvement to naturalness when compared to both vanilla GAN fine-tuning and the baseline without GANs. This demonstrates that the addition of conditioning information does indeed appear to help the discriminator make better distinctions of whether a sample is real or fake, which in turn leads to improvements in the samples produced by the generator.

3.4. Data Reduction Study

Our primary study investigates the impact of the proposed changes to the model architecture and methodology presented in Huybrechts et al. [1], on different amounts of reduced data for the highly expressive target speaker.

In this study, we evaluated the following candidate DR systems: *'DR No-att + cGAN'* and *'DR No-att'*, i.e. the proposed non-autoregressive, external duration TTS model with and without cGAN fine-tuning respectively. We compared them against a baseline DR system to investigate the ablation of our proposed architectural changes and against full-data anchor systems to investigate the ablation of data amount.

'DR baseline' denotes the system presented in Huybrechts et al. [1] which has been shown to synthesise high quality, expressive voices from as little as 15 minutes of data. The same synthetic data is used in the training of both candidate and baseline DR systems.

As full-data anchor systems we used: 1) *'FD Tacotron2'* – a Tacotron2-based TTS model and 2) *'FD No-att'* – the proposed non-autoregressive TTS model. Both full-data systems used an utterance-level VAE and were single-speaker, i.e. trained on all 10 hours of data from the target speaker.

The results of this study are presented in Table 2. They show that in terms of naturalness and speaker similarity, the proposed method, *'DR No-att + cGAN'* significantly outperforms the state-of-the-art approach from Huybrechts et al. [1] (i.e. *'DR Baseline'*) for every data amount, demonstrating a clear improvement to low-resource TTS. Improvements from the proposed changes to the model architecture are further highlighted by the result that *'DR No-att + cGAN'* significantly outperforms *'FD Tacotron2'* when there is 30 minutes or more of target speaker data (up to 95% data reduction) and matches it in the 15 minutes scenario.

Compared to *'FD No-att'*, a full-data model with similar architecture, *'DR No-att + cGAN'* is on par for naturalness while bringing a significant improvement to speaker similarity in the 3 hour scenario and is on par for both metrics in the 1 hour scenario. These results highlight the strength of the proposed 4 step methodology in compensating for the reduction in training data.

The gap between *'DR No-att'* and *'DR No-att + cGAN'* diminishes as we reduce the data further, suggesting that the audio quality improvements brought about from Step 4 (cGAN fine-tuning) are statistically significant only when a relatively large amount of data (3 hours) is available for the target speaker.

Target Data	3 h	1 h	30 min	15 min
Naturalness				
Recordings	85.70	83.61	86.39	82.30
FD No-att	64.17	65.01	61.41	69.27
FD Tacotron2	58.35	58.88	55.07	63.37
DR No-att	62.80	64.96	59.16	64.31
DR No-att + cGAN	64.94	65.29	59.33	64.22
DR baseline	54.40	59.86	51.21	58.73
Speaker similarity				
Recordings	91.94	92.47	94.04	95.95
FD No-att	69.90	65.21	64.85	70.37
FD Tacotron2	64.11	59.90	58.66	64.05
DR No-att	69.14	66.75	62.94	65.97
DR No-att + cGAN	71.54	66.72	62.95	66.21
DR baseline	63.71	60.74	53.58	60.43

Table 2: *Average MUSHRA scores for naturalness and speaker similarity, showing the performance of proposed method in the context of different amount of data. Note that each column is made up of MUSHRA evaluations for one particular data amount, thus scores are not comparable across different columns. Underlined values signify the best performing system amongst DR systems, up to statistically significant differences.*

4. Conclusions

We proposed improvements to the state-of-the-art low-resource TTS technology presented in Huybrechts et al. [1], addressing its limitations when applied to highly expressive voices. The improvements were to: 1) model architecture, i.e. the switch to a non-autoregressive acoustic model supported by external durations instead of an attention-based, autoregressive Tacotron2 architecture, and 2) methodology, i.e. an additional cGAN fine-tuning step.

The proposed system significantly outperforms the state-of-the-art in both naturalness and speaker similarity, closing the gap to recordings by 23.3% and 16.3% respectively, using as little of 15 minutes of speech from the target speaker. Further, compared to a Tacotron2-based model trained on full-data ($\approx$ 10 hours of speech), the proposed model is on par with just 15 minutes of target speaker data and significantly improves naturalness and speaker similarity with 30 minutes or more data. Finally, with 3 hours of target speaker data, our proposed architecture with additional cGAN fine-tuning outperforms even a full-data model of a similar architecture.

These contributions demonstrate a robust NTTS method that can build high quality, natural speech from as little as 15 minutes of target speaker data and can scale even to highly expressive voices. Such a method can save substantial cost and time invested in data collection for TTS.

Future work includes applying the proposed method to more voices that are challenging to model, such as expressive multi-lingual or character voices. Further, we intend to explore fine-grained prosody embeddings to better model and control expressive speech. We also intend to investigate the joint training of acoustic and duration models for the multi-speaker DR scenario, which was found to underperform compared to the separate training approach presented in this paper.

5. References

[1] G. Huybrechts, T. Merritt, G. Comini, B. Perz, R. Shah, and J. Lorenzo-Trueba, "Low-resource expressive text-to-speech using data augmentation," in *ICASSP*. IEEE, 2021, pp. 6593–6597.

[2] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan *et al.*, "Natural tts synthesis by conditioning wavenet on mel spectrogram predictions," in *ICASSP*. IEEE, 2018, pp. 4779–4783.

[3] R. Skerry-Ryan, E. Battenberg, Y. Xiao, Y. Wang, D. Stanton, J. Shor, R. Weiss, R. Clark, and R. A. Saurous, "Towards end-to-end prosody transfer for expressive speech synthesis with tacotron," in *ICML*. PMLR, 2018, pp. 4693–4702.

[4] N. Kalchbrenner, E. Elsen, K. Simonyan, S. Noury, N. Casagrande, E. Lockhart, F. Stimberg, A. Oord, S. Dieleman, and K. Kavukcuoglu, "Efficient neural audio synthesis," in *ICML*. PMLR, 2018, pp. 2410–2419.

[5] A. Oord *et al.*, "Parallel wavenet: Fast high-fidelity speech synthesis," in *ICML*. PMLR, 2018, pp. 3918–3926.

[6] Y.-A. Chung *et al.*, "Semi-supervised training for improving data efficiency in end-to-end speech synthesis," in *ICASSP*. IEEE, 2019, pp. 6940–6944.

[7] A. Gibiansky, S. Ö. Arik, G. F. Diamos, J. Miller, K. Peng, W. Ping, J. Raiman, and Y. Zhou, "Deep voice 2: Multi-speaker neural text-to-speech." in *NIPS*, 2017.

[8] Y. Jia *et al.*, "Transfer learning from speaker verification to multispeaker text-to-speech synthesis," in *NIPS*, 2018, pp. 4480–4490.

[9] N. Tits *et al.*, "Exploring transfer learning for low resource emotional tts," in *Proceedings of SAI Intelligent Systems Conference*. Springer, 2019, pp. 52–60.

[10] J. Latorre *et al.*, "Effect of data reduction on sequence-to-sequence neural tts," in *ICASSP*. IEEE, 2019, pp. 7075–7079.

[11] Y.-J. Chen *et al.*, "End-to-end text-to-speech for low-resource languages by cross-lingual transfer learning." in *Interspeech*, 2019, pp. 2075–2079.

[12] H. Zhang and Y. Lin, "Unsupervised Learning for Sequence-to-Sequence Text-to-Speech for Low-Resource Languages," in *Interspeech*, 2020, pp. 3161–3165.

[13] S. H. Mohammadi *et al.*, "An overview of voice conversion systems," *Speech Communication*, vol. 88, pp. 65–82, 2017.

[14] C.-C. Hsu, H.-T. Hwang, Y.-C. Wu, Y. Tsao, and H.-M. Wang, "Voice conversion from unaligned corpora using variational autoencoding wasserstein generative adversarial networks," in *Interspeech*, 2017, pp. 3364–3368.

[15] J. Lorenzo-Trueba, J. Yamagishi, T. Toda, D. Saito, F. Villavicencio, T. Kinnunen, and Z. Ling, "The voice conversion challenge 2018: Promoting development of parallel and nonparallel methods," *arXiv preprint arXiv:1804.04262*, 2018.

[16] T. Kaneko *et al.*, "Cyclegan-vc2: Improved cyclegan-based nonparallel voice conversion," in *ICASSP*. IEEE, 2019, pp. 6820–6824.

[17] S. Karlapati, A. Moinet, A. Joly, V. Klimkov, D. S'aez-Trigueros, and T. Drugman, "Copycat: Many-to-many fine-grained prosody transfer for neural text-to-speech," in *Interspeech*, 2020.

[18] J. Xu *et al.*, "Lrspeech: Extremely low-resource speech synthesis and recognition," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 2802–2812.

[19] Z. Zhang, Q. Tian, H. Lu, L.-H. Chen, and S. Liu, "Adadurian: Few-shot adaptation for neural text-to-speech with durian," *arXiv preprint arXiv:2005.05642*, 2020.

[20] M. He, Y. Deng, and L. He, "Robust Sequence-to-Sequence Acoustic Modeling with Stepwise Monotonic Attention for Neural TTS," in *Interspeech*, 2019, pp. 1293–1297.

[21] Y. Zheng, J. Tao, Z. Wen, and J. Yi, "Forward–backward decoding sequence for regularizing end-to-end tts," *IEEE/ACM Trans. Audio Speech & Lang. Process.*, vol. 27, no. 12, pp. 2067–2079, 2019.

[22] H. Guo, F. K. Soong, L. He, and L. Xie, "A New GAN-Based End-to-End TTS Training Algorithm," in *Interspeech*, 2019, pp. 1288–1292.

[23] E. Battenberg, R. Skerry-Ryan, S. Mariooryad, D. Stanton, D. Kao, M. Shannon, and T. Bagby, "Location-relative attention mechanisms for robust long-form speech synthesis," in *ICASSP*, 2020, pp. 6194–6198.

[24] Y. Ren, C. Hu, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, "Fastspeech 2: Fast and high-quality end-to-end text to speech," in *ICLR*, 2021.

[25] I. Elias, H. Zen, J. Shen, Y. Zhang, Y. Jia, R. J. Weiss, and Y. Wu, "Parallel tacotron: Non-autoregressive and controllable tts," in *ICASSP*. IEEE, 2021, pp. 5709–5713.

[26] J. Shen, Y. Jia, M. Chrzanowski, Y. Zhang, I. Elias, H. Zen, and Y. Wu, "Non-Attentive Tacotron: Robust and Controllable Neural TTS Synthesis Including Unsupervised Duration Modeling," *arXiv:2010.04301*, 2020.

[27] H. Zen, K. Tokuda, and A. Black, "Statistical Parametric Speech Synthesis," *Speech Communication*, vol. 51, no. 11, pp. 1039–1064, 2009.

[28] H. Ze, A. Senior, and M. Schuster, "Statistical parametric speech synthesis using deep neural networks," in *ICASSP*. IEEE, 2013, pp. 7962–7966.

[29] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.

[30] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *arXiv preprint arXiv:1406.2661*, 2014.

[31] T. Kaneko, H. Kameoka, N. Hojo, Y. Ijima, K. Hiramatsu, and K. Kashino, "Generative adversarial network-based postfilter for statistical parametric speech synthesis," in *ICASSP*, March 2017, pp. 4910–4914.

[32] T. Kaneko, S. Takaki, H. Kameoka, and J. Yamagishi, "Generative adversarial network-based postfilter for stft spectrograms," in *Interspeech*, 2017.

[33] Y. Jiao, A. Gabryś, G. Tinchev, B. Putrycz, D. Korzekwa, and V. Klimkov, "Universal neural vocoding with parallel wavenet," in *ICASSP*. IEEE, 2021, pp. 6044–6048.

[34] D. P. Kingma *et al.*, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[35] L. Wan, Q. Wang, A. Papir, and I. L. Moreno, "Generalized end-to-end loss for speaker verification," in *ICASSP*. IEEE, 2018, pp. 4879–4883.

[36] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *arXiv preprint arXiv:1706.03762*, 2017.

[37] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention generative adversarial networks," in *International conference on machine learning*. PMLR, 2019, pp. 7354–7363.

[38] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[39] S. R. Bowman, L. Vilnis, O. Vinyals, A. Dai, R. Jozefowicz, and S. Bengio, "Generating sentences from a continuous space," in *CoNLL*. ACL, 2016, pp. 10–21.

[40] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The kaldi speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding*, no. CONF. IEEE Signal Processing Society, 2011.

[41] R. ITU-R, "1534-1, method for the subjective assessment of intermediate quality levels of coding systems (mushra),"," *International Telecommunication Union*, 2003.

11th ISCA Speech Synthesis Workshop (SSW 11)
26-28 August 2021, Budapest, Hungary

Intelligibility and naturalness of articulatory synthesis with VocalTractLab compared to established speech synthesis technologies

Paul Konstantin Krug, Simon Stone, Peter Birkholz

Technische Universität Dresden
paul_konstantin.krug@tu-dresden.de

Abstract

In this work, the current state-of-the-art of articulatory speech synthesis (VocalTractLab) is compared to a wide range of different text-to-speech systems that once represented or still represent the continuously evolving state-of-the-art of speech synthesis technology. The comparison systems include neural and concatenative synthesis by Google and Microsoft, as well as Hidden Markov Model-based, unit-selection and diphone synthesis developed at universities (using MaryTTS, MBROLA and DRESS). A small corpus of 15 German sentences was synthesized using the text-to-speech (and, if available, re-synthesis) functionalities of each system. The intelligibility of the synthesized utterances was evaluated in an ASR experiment. The naturalness of the utterances was evaluated in a multi-stimulus Likert test by 50 German native speakers. As an additional reference, recordings of natural speech were used in the experiments. It was found that the articulatory synthesis can achieve a performance on par with the non-commercial synthesis systems in terms of intelligibility and naturalness, while being significantly outperformed by the commercial synthesis systems.

Index Terms: text-to-speech, articulatory speech synthesis.

1. Introduction

From mechanical speech apparatuses [1], to electrical vocal tract analogues [2–6], up to sophisticated computer simulations [7–13]: articulatory speech synthesis has been a topic of research for centuries. Despite the fact that this kind of synthesis can be considered the most natural approach to speech synthesis, as it aims to directly model the speech production process that happens in a human vocal tract, it never played a significant role outside the academic world [14]. This is mainly due to (i): the difficulties that arise from modelling the time-dependent vocal tract geometries, which need to be controlled up to a very precise level. This requires a deep understanding of human speech production and knowledge of articulatory movements, which are not easily accessible experimentally. (ii) For a long time, no complete aerodynamic-acoustic simulation of the vocal tract existed. And (iii): At any given time, better sounding alternative methods were available (e.g. formant synthesis, parametric synthesis, concatenation synthesis or recently neural synthesis) that did require less or no explicit knowledge of articulatory movements. Furthermore, the generation of synthesized utterances with articulatory synthesizers generally involves a lot of manual tuning, which is usually a very time consuming process that requires expert knowledge.

Apart from very few (and outdated) exceptions such as GnuSpeech [15], no modern articulatory text-to-speech (TTS) systems were available until now. This situation has changed with the recent development of the state-of-the-art articulatory synthesizer VocalTractLab[1] [12] (VTL) version 2.3 that introduced a fully automatic phoneme-to-speech conversion for German. Using this functionality, it is possible to generate high quality re-syntheses of any given German utterance. By extending the VTL with an additional grapheme-to-phoneme conversion (G2P) and an intonation model, it is possible to setup a complete TTS pipeline [16]. Although the produced speech by VTL sounds intelligible, it is not yet known how VTL speech compares against state-of-the-art systems of well established speech synthesis technologies in terms of intelligibility and naturalness.

The current study aims to rank the VTL synthesis among widely used speech synthesis technologies such as diphone, Hidden Markov Model (HMM), unit-selection and neural synthesis. It extends the state of research on articulatory synthesis by the following contributions:

1. A full articulatory TTS system based on the open source software VocalTractLab is presented (VTL-TTS).
2. A fair comparison of articulatory synthesis (both fully-automatic TTS and manual re-synthesis, which in this case means to derive phone durations and pitch information from natural speech recordings) with eight different types of syntheses, as well as natural speech, is presented in terms of intelligibility and naturalness. Although this involves systems under active development and thus can only serve as a snapshot, it gives valuable insight into the speech synthesis landscape on the whole at this point in time.

2. Methods

A small corpus of 15 German sentences, presented in Table 1, was synthesized in a neutral speaking style using different TTS systems, namely Google Cloud TTS [17–21][2], Microsoft Azure TTS [22][2], MaryTTS [23], DRESS [24], as well as VTL-TTS. Additionally, natural speech recordings of the 15 sentences were manually re-synthesized using VTL and MBROLA [25]. The intelligibility of the syntheses was evaluated using automatic speech recognition (ASR). The naturalness of the syntheses was evaluated by 50 German native speakers in a listening experiment. Finally, a deep learning-based system for speech naturalness evaluation (NISQA) [26] was evaluated against the results from the listening experiment. All audio sample files and the data files necessary to reproduce the synthesized files are available in the supplementary materials[3].

[1] https://www.vocaltractlab.de/ (Last visited 22.04.2021).

[2] Since the companies' systems are proprietary and continuously developed, no exact descriptions of the systems are available. Hence, the references should be understood as an (incomplete) overview of important contributions to the used technologies.

[3] https://github.com/TUD-STKS/TTS_Comparison_SSW21 (Last visited 22.04.2021).

10.21437/SSW.2021-18

2.1. Articulatory synthesis and TTS pipeline

2.1.1. VocalTractLab

The articulatory synthesizer VTL provides a one-dimensional aero-acoustic simulation [27] within a model of the vocal tract that is based on magnetic resonance imaging (MRI) scans of a real human vocal tract [12]. The current version VTL 2.3 provides three different types of vocal fold models [28–30]. In this study, the geometric glottis model [28] was used, which is the VTL default.
During the time domain simulation, the articulatory dimensions of VTL are controlled by a set of time-dependent functions, a so called *gestural score* [31, 32]. A gestural score consists of several tiers, which describe the shape of the articulators, the glottis shape, the intonation and the lung pressure, respectively. While VTL allows for the direct construction and manipulation of the gestural score and thus precise control, VTL 2.3 also offers a more convenient higher level user interface (for German speech). By providing a sequence of phone labels and their respective acoustic durations, a gestural score of articulatory movements can be automatically generated, excluding the pitch contour. Therefore, only the missing intonation needs to be generated either manually or by some external means (see Section 2.2.2). The generated score can be freely edited after the automatic generation, which allows a semi-automatic workflow where an utterance is initialized automatically and then tuned manually (e.g., to match a reference utterance).

2.1.2. VTL-TTS

The used VTL-TTS pipeline consists of several stages. First, a given plain input text is converted into its SAMPA transcription, using a proprietary Web service by Aristech Gmbh [16]. The transcription also provides further annotations, such as the utterance's syllables and information on the linguistic stress of the syllables. Subsequently, a set of 70 phonetic and linguistic features is calculated. An intonation contour for the utterance is then predicted using these features fed to a deep neural network. Finally, the phone durations are predicted using empirically determined, context-dependent reference values taken from [33]. The phone sequence is then turned into a gestural score using the segment sequence interface of VTL 2.3 described above and then converted into audio.

2.2. Stimuli preparation and preprocessing

2.2.1. TTS synthesis

Six of the TTS voices were accessed via their Web clients, namely Microsoft Azure TTS[4], Google Cloud TTS[5] and MARYTTS[6]. In case of the former two services, both a neural synthesis (in the following referred to as *Azure-Neural* and *Google-Neural*), and a parametric/unit-selection[7] synthesis (in the following referred to as *Azure-Standard* and *Google-Standard*) were used to produce the desired samples. In case of the MARYTTS system, samples were synthesized via HMM-based synthesis (using the German voice *dfki-pavoque-neutral-hsmm de male hmm*, in the following referred to as *dfki-HMM*) and via unit-selection synthesis (using the German voice *dfki-pavoque-neutral de male unitselection general*, in the following referred to as *dfki-unit*) [34, 35]. The other MARYTTS parameters were set the following way (for both voices): "Input Type": *TEXT*, "Output Type": *AUDIO*, "Audio-Out": *WAVE_FILE* and "Audio-Effects": *Default (all turned off)*. For the Azure-Neural and Azure-Standard syntheses, the parameter "Voice" was set to *Conrad (Neural)* and *Stefan*, respectively. The other parameters were set the following way (for both the neural and parametric/unit-selection syntheses): "Language": *German (Germany)*, "Voice Style": *General*, "Speaking Speed": *1.00* and "Pitch": *0.00*. For the Google-Neural and Google-Standard syntheses, the parameter "Voice type" was set to *WaveNet* and *Basic*, respectively. The parameter "Voice name" was set to *de-DE-Wavenet-B* and *de-DE-Standard-B*, respectively. The other parameters were set the following way (for both voices): "Language": *Deutsch (Deutschland)*, "Speed": *1.00*, "Pitch": *0.00* and "Audio device profile": *Default*. Furthermore, samples were created using DRESS, which is a pure diphone TTS synthesis using the TD-PSOLA [36] algorithm. The male voice *Jörg* was used during the synthesis. The "Rhythmn" parameter was set to *Klatt* and the "Intonation" parameter was set to *Fujisaki (dt)*. Finally, samples were created using VTL-TTS using the previously described processing pipeline.

2.2.2. Re-synthesis

The term re-synthesis describes a synthetic reproduction of a natural speech recording that matches the original recording as precisely as possible. In case of VTL, a manual re-synthesis performed by an expert represents the highest quality that is currently achievable with the software. Hence, manual re-syntheses can give an idea of the maximum possible VTL-TTS performance, if the pre-processing (i.e. G2P, phone duration prediction and intonation prediction etc.) was ideal. For this reason, the manual VTL re-synthesis was also evaluated against the TTS systems in the experiments. In order to generate the natural utterances, necessary for the re-syntheses, a 24-year-old German native speaker was recorded at a sample rate of 44.1 kHz. Subsequently, the recordings were loaded into VTL, where the respective phoneme sequence was aligned with the natural speech so that the reproduced speech matched the original utterances as closely as possible in terms of timing. In order to match the intonation as well, the natural f_0 contour of each sentence was extracted using the software PRAAT [37]. The software TARGETOPTIMIZER [38, 39] (TO) was used in order to fit the natural contours using the TARGET-APPROXIMATION-MODEL [40, 41] (TAM). This step was necessary since the pitch and articulatory gestures of VTL are based on the TAM. The obtained pitch gestures were loaded into VTL and manually fine-tuned when necessary. The audio samples were synthesized using the speaker file *JD2*, which is the default VTL speaker. There is no relation between the recorded speaker and the speaker on whose data the JD2 model is based on (apart from both persons being male). The audio samples were exported as WAV files with a sample rate of 44.1 kHz.
In order to have a second re-synthesis system to compare with VTL, an additional diphone re-synthesis was made using the open source software MBROLA[8]. The same phone durations and pitch contours as for the VTL re-syntheses were used. However, the used database for the male German speaker *de2* does

[4]https://azure.microsoft.com/en-us/services/cognitive-services/text-to-speech (last visited 09.02.2021).
[5]https://cloud.google.com/text-to-speech (last visited 22.01.2021).
[6]http://mary.dfki.de:59125/ (last visited 22.01.2021).
[7] The companies are not specific about the exact technology that is used for the standard (non-neural) voices. They state such voices are created using either parametric or unit-selection synthesis or a mixture of both.
[8]https://github.com/numediart/MBROLA (Last visited 14.02.2021).

	Utterance	IPA	Translation
1	Aber sehen will sie ihn doch.	ˈʔaːbɐ ˈzeːn vɪl ziː ʔiːn dɔx	*But she wants to see him.*
2	Er sah viele bunte Regenbogen.	eːɐ̯ zaː ˈfiːlə ˈbʊntə ˈʁeːgn̩ˌboːgn̩	*He saw many colourful rainbows.*
3	Chabos wissen wer der Babo ist.	ˈʧaːboːs ˈvɪsn̩ veːɐ̯ deːɐ̯ ˈbaːboː ʔɪst	*The boys know who the boss is.*
4	Das Telefon ist seit sieben Tagen kaputt.	das ˈteːləfoːn ʔɪst zaɪ̯t ˈziːbn̩ ˈtaːgn̩ kaˈpʊt	*The phone has been broken for seven days.*
5	Die Artikel waren wieder vorrätig.	diː ˌʔaʁˈtiːkl̩ ˈvaːʁən ˈviːdɐ ˈfoːɐ̯ˌʁɛtɪç	*The products were in stock again.*
6	Die Soße ist viermal übergekocht.	diː ˈzoːsə ʔɪst ˈfiːɐ̯ maːl ˈʔyːbɐgəˌkɔxt	*The sauce boiled over four times.*
7	Die Straßenbahn fuhr weiter geradeaus.	diː ˈʃtʁaːsn̩ˌbaːn fuːɐ̯ ˈvaɪ̯tɐ gəʁaːdəˈʔaʊ̯s	*The tram continued straight ahead.*
8	Diese Zeitung ist bereits veraltet.	ˈdiːzə ˈʦaɪ̯tʊŋ ʔɪst bəˈʁaɪ̯ʦ fɛɐ̯ˈʔaltət	*This newspaper is already outdated.*
9	Sie fährt keinen Ferrari, sondern einen Maserati.	ziː fɛːɐ̯t ˈkaɪ̯nən feˈʁaːʁiː ˈzɔndɐn ˈʔaɪ̯nən mazəˈʁaːti	*She does not drive a Ferrari, but a Maserati.*
10	Benno gefällt die orange Vase.	ˈbɛno gəˈfɛlt diː ʔoˈʁaŋʒə ˈvaːzə	*Benno likes the orange vase.*
11	Es kann hilfreich sein, wenn man weiß, wie ein Unterstand gebaut wird.	ʔɛs kan ˈhɪlfʁaɪ̯ç zaɪ̯n vɜn man vaɪ̯s viː ʔaɪ̯n ˈʔʊntɐʃtant gəˈbaʊ̯t vɪʁt	*It can be helpful to know how to build a shelter.*
12	Er schützt vor Kälte, Wind und Niederschlägen.	ʔeːɐ̯ ʃʏʦt foːɐ̯ ˈkɛltə vɪnt ʔʊnt ˈniːdɐʃlɛːgən	*It protects against cold, wind and precipitation.*
13	Conny glaubt eigentlich nicht mehr an den Osterhasen.	kɔni glaʊ̯pt ˈaɪ̯gn̩tlɪç nɪçt meːɐ̯ ʔan deːn ˈoːsteˌhaːzən	*Conny doesn't really believe in the Easter Bunny any more.*
14	Sie läuft schnell hin.	ziː lɔɪ̯ft ʃnɛl hɪn	*She runs there quickly.*
15	Der Petersdom ist das Wahrzeichen des Vatikans.	deːɐ̯ ˈpeːtɐsˌdoːm ɪst das ˈvaːɐ̯ʦaɪ̯çn̩ dɛs vatiˈkaːns	*St Peter's Basilica is the landmark of the Vatican*

Table 1: *The used utterances in German, their canonical IPA transcription, and English translation.*

not contain a glottal stop. The durations of existing glottal stops in the segment sequence files used in the VTL re-synthesis were therefore split half and half between the left and right neighbouring phones. Secondary diphthongs such as /oːɐ/ were broken down into the two individual vowels, each with half the total duration. For MBROLA re-syntheses, the f_0 contours were constructed as linear interpolations between f_0 support points. On average, as many f_0 support points were used as there were phones in the utterance.

2.2.3. Natural speech

In addition to the synthetic speech samples, natural speech recordings of the 15 German sentences were also evaluated in all experiments to serve as anchor points. The speaker for the natural stimuli was different from the speaker for the re-synthesis reference recordings in order to avoid possible biases, e.g. regarding the f_0 contour. For the natural samples, a male 27-year-old non-professional German native speaker was recorded at a sample rate of 44.1 kHz. As in the previous case, there is no relation between this speaker and the VTL JD2 model. For the recordings a large diaphragm condenser microphone was used (*Microtech Gefell M930*). It was connected to a low-noise pre-amplifier (*Behringer Eurorack MX 1602*). The pre-amp was then connected to an audio interface (*MOTU 896 HD*) which was connected to a PC via FireWire. The natural speech audio samples were recorded in a sound-proofed audio studio. The speaking style was neutral.

2.2.4. Re-sampling and loudness normalization

The various synthetic and natural speech samples have different sample rates. Hence, the amount of high frequency content differs among the samples, since no frequencies can be present beyond the respective Nyquist frequencies. However, the presence or absence of high frequencies are part of the technologies that should be evaluated in this study. Hence, the samples were intentionally not downsampled to the smallest sample rate present in the data (which would implicate a high frequency cut-off for some of the samples). Instead, they were upsampled to the largest present sample rate that is 44.1 kHz to facilitate further processing without distorting the frequency contents. Afterwards, all samples were loudness normalized. This is very important since the various speech samples produced with the different technologies (even though peak normalized) differed widely in their loudness. However, the loudness of a sample might significantly impact the rating on a psychometric scale [42]. Hence, the audio amplitudes of all samples were first peak normalized to -1 dB FS. Subsequently, the integrated loudness according to the ITU-R BS.1770-4 recommendation (measured in dB LUFS) was calculated for each sample using the PYTHON library PYLOUDNORM. Using the same tool, all samples were then loudness normalized to the minimal loudness obtained in the previous step, which was -25.7 dB LUFS. This way all stimuli had the same loudness and the maximum peak amplitude among all samples was -1 dB FS.

2.3. Evaluation of intelligibility

Evaluating the intelligibility of the audio samples in a perception experiment with human listeners would be challenging, due to the high number of participants that would be required to obtain an adequate statistical power. Hence, automatic speech recognition was chosen as a tool to measure the intelligibility of the synthetic and natural speech samples. Four state-of-the-art commercial ASR systems, namely Google Web API, Microsoft Azure speech-to-text, IBM Watson speech-to-text and Wit.ai (owned by Facebook), were accessed via their respective API using the PYTHON libraries SPEECHRECOGNITION and IBM-WATSON. Four different systems were used in order to reduce a possible impact from the biases of the ASR systems towards certain speech styles, f_0, voice etc. The audio files were sent to each service and the speech-to-text conversion was returned as a string. The word error rate (WER) between the true text and the ASR answer was calculated using the python

library JIWER. Thereby, both the true and recognized strings were pre-processed in the following way: The punctuation was removed from the strings, all characters were converted to lower case, double or multiple white spaces were converted to a single white space, leading and trailing whitespaces were removed.

2.4. Evaluation of naturalness

2.4.1. Listening experiment

In order to evaluate the naturalness of all samples, an online perception experiment was carried out using the tool webMUSHRA[9] [43]. The experiment was designed as a multi-stimulus Likert test. Thereby, participants would see a single page per sentence that contained all eleven versions of that sentence. Each version had to be played and rated in order to proceed to the next page. Participants could play an audio sample as often as desired. Each page displayed the utterance text at the top of the page. Below that each page featured the following instructions (translated to English): "*On a scale of 1 to 5 stars, how natural (i.e., how human) does each utterance sound? (1: Very unnatural, 2: Rather unnatural, 3: Neither, 4: Rather natural 5: Very natural). You have to play all versions to the end and rate all versions.*"
At the beginning of the test, participants were asked to play an example audio sample in order to adjust their listening volume to a pleasant level. Thereby, the example file was the sentence (translated to English): "*Please listen to the following sample sentence and adjust the volume so that you find it comfortable.*". It was synthesized using the IBM TTS[10] online client that was not used for other samples in the experiment. Just as all other samples, the example file was loudness normalized to -25.7 dB LUFS.
In total, 50 subjects (18 male, 32, female) aged between 18 and 50 years (median: 24.0 years, mean: 26.6 ± 6.7 years) participated in the experiment. Participants were required to be German native speakers, but due to the online nature, no additional screenings were conducted. To avoid a bias of the results, experts in (articulatory) speech synthesis technology were not encouraged to participate.

2.4.2. NISQA

As an automatic kind of speech quality assessment, the pre-trained CNN-BLSTM NISQA-TTS[11] [26] model was used in order to evaluate the naturalness of the synthesized speech samples. The predicted NISQA scores were then compared to the ratings of the human listeners to evaluate the predictive power of such an automated assessment system. This is of particular interest for articulatory synthesis, since the produced speech is not directly derived from original, human recordings, which might break the assumptions of a pre-trained assessment model.

3. Results

3.1. Evaluation of intelligibility

The word error rates across all samples are shown in Figure 1 for all four ASR systems separately. While the median of each distribution is zero, one can see that the means (Google: 0.08 ± 0.15, Azure: 0.13 ± 0.22, Wit.ai: 0.16 ± 0.24, IBM: 0.19 ± 0.27) differ due to the different amount of outliers. Based on two-sided Mann-Whitney U tests (MWU tests), the Google WER distribution of the Google ASR system is significantly different from those of Wit.ai and IBM ($p < 0.01$), but not significantly different from the distribution of the Azure system ($p > 0.01$). No significance was observed between permutations of Azure, Wit.ai and IBM ($p > 0.01$).

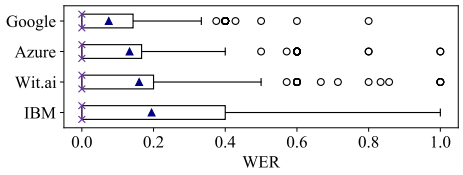


Figure 1: *Word error rates across all speech samples, separated into single distributions for the four ASR systems, shown as box plots. The position of the median of each distribution is indicated by two x-shaped markers. The position of the respective mean is indicated by a triangle.*

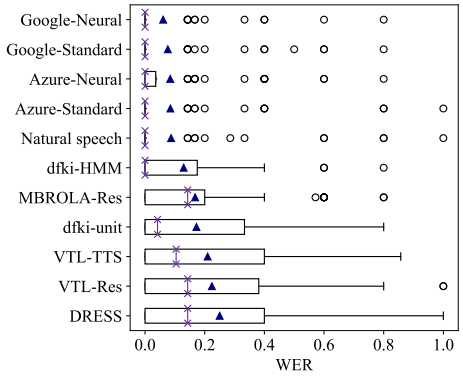


Figure 2: *Word error rates across all ASR systems, separated into single distributions for each type of synthesis, shown as box plots. Medians are indicated by two x-shaped markers. Means are indicated by a triangle.*

Figure 2 shows the WER distributions for all tested synthesis types across all ASR systems. The synthesis types are sorted by their respective mean (top: best performance, bottom: worst performance). It is observed that the first five synthesis types (Google-Neural and Standard, Azure-Neural and Standard, as well as the natural speech) achieve a median WER of 0.0 across all ASR systems, which means they are mostly identified correctly. The distributions differ slightly in their mean values, but this is mainly due to the outliers. While the median WER of the dfki-HMM syntheses is also 0.0, the distribution is still significantly broader than the distribution of the natural speech samples and those of the Google syntheses ($p < 0.01$, based on two-sided MWU tests), resulting in a higher mean. No significant difference was found among permutations of WER dis-

[9] Despite its name, the tool is not limited to MUSHRA tests, but can be used for several kinds of listening experiments. In this analysis, a multi-stimulus Likert test was performed.
[10] https://www.ibm.com/demos/live/tts-demo/self-service/home (Last visited 22.01.2021).
[11] https://github.com/gabrielmittag/NISQA (Last visited 14.02.2021).

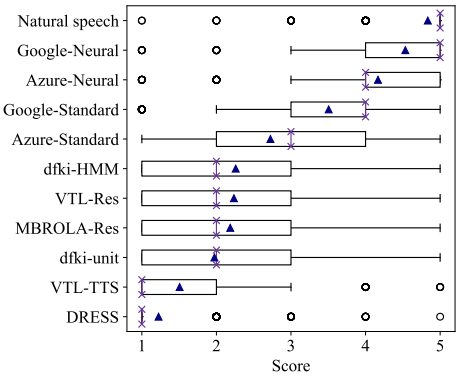


Figure 3: *Likert scores across all listeners, separated into single distributions for each type of synthesis, shown as box plots. Medians are indicated by two x-shaped markers. Means are indicated by a triangle.*

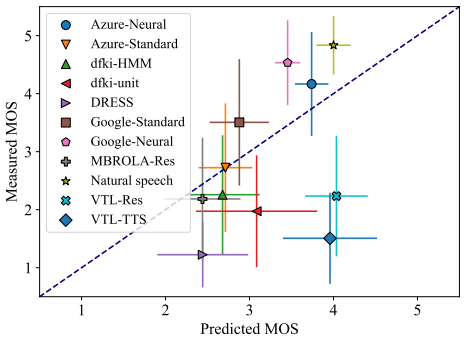


Figure 4: *Subjective MOS measured in the listening experiment plotted against predicted MOS values determined with the NISQA network. The errorbars indicate the $\pm 1\,\sigma$ interval around the mean.*

tributions of the non-commercial synthesis systems. The WER medians of the five worst performing technologies deviate from zero and range from 0.04 (dfki-unit) to 0.14 (DRESS). The mean values range from 0.17 ± 0.22 (MBROLA) to 0.25 ± 0.3 (DRESS).

3.2. Evaluation of naturalness

The results from the listening test are shown in Figure 3. The order of synthesis types decreases in performance from top to bottom (top: rated as most natural, bottom: rated as most unnatural). The constituents of all possible distribution pairs, except for permutations of dfki-HMM, VTL-Res and MBROLA-Res, are significantly different ($p < 0.01$) from each other, based on two-sided MWU tests. The natural speech performed best, with a mean rating of 4.84 ± 0.50. It is followed by the two neural syntheses (Google-Neural: 4.53 ± 0.73, Azure-Neural: 4.17 ± 0.89). The commercial parametric/unit-selection syntheses perform worse than the neural syntheses, with mean values of 3.51 ± 1.09 and 2.72 ± 1.11, respectively. The re-syntheses perform worse and similar to the dfki syntheses. VTL-TTS is rated significantly less natural (1.51 ± 0.78) and DRESS samples were rated to be the least natural sounding samples (1.22 ± 0.56).

Figure 4 shows the measured subjective scores plotted against the predicted scores from the NISQA-TTS model. It is observed that the predicted scores do not agree well with the measured data. While the performance of VTL-Res, VTL-TTS, dfki-unit and DRESS is greatly overestimated, the performance of the neural syntheses and the natural speech is underestimated. The linear correlation coefficient between the predicted and measured values is $\rho = 0.28$.

4. Discussion

A small corpus of 15 German sentences was synthesized using a wide range of different TTS systems that once represented or still represent the continuously evolving state-of-the-art both in the commercial and the academic domain of speech synthesis technology. The intelligibility and the naturalness of the syntheses was evaluated and compared against natural speech in an ASR experiment and in a listening experiment, respectively.

From the ASR experiment, it was observed that the WER's of the commercial TTS systems did not differ significantly from the WER of natural speech. It can be concluded that the obtained WER is rather limited by the recognition performances of the ASR systems and less by the quality of the artificial speech samples. The main reasons for the significantly worse performance of the non-commercial systems are probably the synthesis artifacts that are quite audible in case of dfki-HMM, dfki-unit and DRESS. Further, the intonation and phone durations have an impact on the performance. This is well exemplified in case of VTL-TTS and VTL-Res. Despite pitch contours and phone durations copied from natural utterances, VTL-Res performs worse than VTL-TTS with regard to WER. It seems likely that the longer and more uniform distibuted phone durations of the VTL-TTS system increase the intelligibility in this case.

As expected, the participants in the listening experiment considered the natural speech samples as the most natural sounding samples. Despite not being directly comparable due to the experimental setup of the Likert test, the obtained scores for the Google-Neural and Standard syntheses are in agreement with the MOS scores reported in [21]. In terms of naturalness, VTL-TTS performs significantly worse than VTL-Res. Hence, a more realistic modeling of intonation and phone duration could improve the articulatory TTS pipeline a lot.

To conclude, none of the TTS systems is both, as natural and as intelligible as natural speech yet, even though the commercial neural voices come very close. However, the non-commercial syntheses perform significantly worse. Within the subgroup of academic systems, semi-automatic articulatory re-synthesis proved to be very competitive in terms of naturalness and was not significantly worse than the best non-commercial system dfki-HMM. However, in order for articulatory synthesis to keep up with the modern commercial systems, the overall quality would have to improve greatly. Starting points for improving intelligibility and naturalness of VTL syntheses include an improved modeling of the noise sources inside the vocal tract, modeling tongue-loops [44], and microprosodic effects.

5. Acknowledgements

This work has been partially funded by the Leverhulme Trust Research Project Grant RPG-2019-241: "High quality simulation of early vocal learning".

6. References

[1] W. v. Kempelen, "Mechanismus der menschlichen Sprache nebst Beschreibung einer sprechenden Maschine." Degen, 1791.

[2] H. Dudley *et al.*, "A synthetic speaker," *J. Franklin Inst.*, vol. 227, no. 6, pp. 739–764, 1939.

[3] H. Dudley, "Remaking speech," *J. Acoust. Soc. Am.*, vol. 11, no. 2, pp. 169–177, 1939.

[4] H. K. Dunn, "The calculation of vowel resonances, and an electrical vocal tract," *J. Acoust. Soc. Am.*, vol. 22, no. 6, pp. 740–753, 1950.

[5] K. N. Stevens *et al.*, "An electrical analog of the vocal tract," *J. Acoust. Soc. Am.*, vol. 25, no. 4, pp. 734–742, 1953.

[6] G. Rosen, "Dynamic analog speech synthesizer," *J. Acoust. Soc. Am.*, vol. 30, no. 3, pp. 201–209, 1958.

[7] P. Mermelstein, "Articulatory model for the study of speech production," *J. Acoust. Soc. Am.*, vol. 53, no. 4, pp. 1070–1082, 1973.

[8] P. Rubin *et al.*, "An articulatory synthesizer for perceptual research," *J. Acoust. Soc. Am.*, vol. 70, no. 2, pp. 321–328, 1981.

[9] S. Maeda, "A digital simulation method of the vocal-tract system," *Speech Commun.*, vol. 1, no. 3-4, pp. 199–229, 1982.

[10] P. Rubin *et al.*, "CASY and extensions to the task-dynamic model," in *1st ETRW on Speech Production Modeling: From Control Strategies to Acoustics; 4th Speech Production Seminar: Models and Data*, 1996, pp. 125–128.

[11] O. Engwall, "Combining MRI, EMA and EPG measurements in a three-dimensional tongue model," *Speech Commun.*, vol. 41, no. 2-3, pp. 303–329, 2003.

[12] P. Birkholz, "Modeling consonant-vowel coarticulation for articulatory speech synthesis," *PloS ONE*, vol. 8, no. 4, p. e60603, 2013.

[13] A. Pont *et al.*, "Finite element generation of sibilants /s/ and /z/ using random distributions of Kirchhoff vortices," *Int. J. Numer. Methods Biomed. Eng.*, vol. 36, no. 2, p. e3302, 2020.

[14] C. H. Shadle and R. I. Damper, "Prospects for articulatory synthesis: A position paper," in *SSW4*, 2001.

[15] D. Hill *et al.*, "Real-time articulatory speech-synthesis-by-rules," in *Proc. AVIOS*, vol. 95, 1995, pp. 11–14.

[16] S. Stone *et al.*, "Prospects of articulatory text-to-speech synthesis," in *Proc. ISSP*, 2020 (Accepted).

[17] X. Gonzalvo *et al.*, "Recent advances in Google real-time HMM-driven unit selection synthesizer," in *Proc. Interspeech*, 2016, pp. 2238–2242.

[18] H. Zen *et al.*, "Fast, compact, and high quality LSTM-RNN based statistical parametric speech synthesizers for mobile devices," in *Proc. Interspeech*, 2016, pp. 2273–2277.

[19] A. van den Oord *et al.*, "WaveNet: A generative model for raw audio," *CoRR*, vol. abs/1609.03499, 2016.

[20] Y. Wang *et al.*, "Tacotron: A fully end-to-end text-to-speech synthesis model," *CoRR*, vol. abs/1703.10135, 2017.

[21] J. Shen *et al.*, "Natural TTS synthesis by conditioning wavenet on mel spectrogram predictions," *CoRR*, vol. abs/1712.05884, 2017.

[22] Y. Ren *et al.*, "FastSpeech: Fast, robust and controllable text to speech," in *Proc. NeurIPS*, vol. 32, 2019, pp. 3171–3180.

[23] M. Schröder and J. Trouvain, "The German text-to-speech synthesis system MARY: A tool for research, development and teaching," *Int. J. Speech Technol.*, vol. 6, pp. 365–377, 2003.

[24] R. Hoffmann *et al.*, "Evaluation of a multilingual TTS system with respect to the prosodic quality," in *Proc. ICPhS*, vol. 3, 1999, pp. 2307–2310.

[25] T. Dutoit *et al.*, "The MBROLA project: Towards a set of high quality speech synthesizers free of use for non commercial purposes," in *Proc. ICSLP*, vol. 3, 1996, pp. 1393–1396.

[26] G. Mittag and S. Möller, "Deep learning based assessment of synthetic speech naturalness," in *Proc. Interspeech*, 2020, pp. 1748–1752.

[27] P. Birkholz, "Enhanced area functions for noise source modeling in the vocal tract," in *Proc. ISSP*, 2014, pp. 32–40.

[28] P. Birkholz *et al.*, "Perceptual optimization of an enhanced geometric vocal fold model for articulatory speech synthesis." in *Proc. Interspeech*, 2019, pp. 3765–3769.

[29] ——, "Synthesis of breathy, normal, and pressed phonation using a two-mass model with a triangular glottis," in *Proc. Interspeech*, 2011, pp. 2681–2684.

[30] K. Ishizaka and J. L. Flanagan, "Synthesis of voiced sounds from a two-mass model of the vocal cords," *Bell Syst. Tech. J.*, vol. 51, no. 6, pp. 1233–1268, 1972.

[31] P. Birkholz, "Control of an articulatory speech synthesizer based on dynamic approximation of spatial articulatory targets," in *Proc. Interspeech*, 2007, pp. 2865–2868.

[32] P. Birkholz *et al.*, "Model-based reproduction of articulatory trajectories for consonant-vowel sequences," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 19, no. 5, pp. 1422–1433, 2010.

[33] B. Möbius and J. Von Santen, "Modeling segmental duration in German text-to-speech synthesis," in *Proc. ICSLP*, vol. 4, 1996, pp. 2395–2398.

[34] I. Steiner *et al.*, "Symbolic vs. acoustics-based style control for expressive unit selection," in *SSW7*, 2010, pp. 114–119.

[35] M. Schröder *et al.*, "Open source voice creation toolkit for the MARY TTS Platform," in *Proc. Interspeech*, 2011, pp. 3253–3256.

[36] C. Hamon *et al.*, "A diphone synthesis system based on time-domain prosodic modifications of speech," in *Proc. ICASSP*, 1989, pp. 238–241.

[37] P. Boersma and D. Weenick, "Praat: Doing phonetics by computer (version 6.0.43) [computer program]," 2005, http://www.praat.org (Last visited 15.02.2021).

[38] P. Birkholz *et al.*, "Estimation of pitch targets from speech signals by joint regularized optimization," in *Proc. EUSIPCO*, 2018, pp. 2075–2079.

[39] P. K. Krug *et al.*, "Targetoptimizer 2.0: Enhanced estimation of articulatory targets," in *Studientexte zur Sprachkommunikation: Elektronische Sprachsignalverarbeitung 2021*, 2021, pp. 145–152.

[40] Y. Xu and Q. E. Wang, "Pitch targets and their realization: Evidence from Mandarin Chinese," *Speech Commun.*, vol. 33, no. 4, pp. 319–337, 2001.

[41] S. Prom-On *et al.*, "Modeling tone and intonation in Mandarin and English as a process of target approximation," *J. Acoust. Soc. Am.*, vol. 125, no. 1, pp. 405–424, 2009.

[42] E. Vickers, "The loudness war: Background, speculation, and recommendations," in *Audio Engineering Society Convention 129*. Audio Engineering Society, 2010.

[43] M. Schoeffler *et al.*, "webMUSHRA — A comprehensive framework for web-based listening tests," *J. Open Res. Software*, vol. 6, no. 1, p. 8, 2018.

[44] H. Nam *et al.*, "Hearing tongue loops: Perceptual sensitivity to acoustic signatures of articulatory dynamics," *J. Acoust. Soc. Am.*, vol. 134, no. 5, pp. 3808–3817, 2013.

11th ISCA Speech Synthesis Workshop (SSW 11)
26-28 August 2021, Budapest, Hungary

Perception of smiling voice in spontaneous speech synthesis

Ambika Kirkland[1], *Marcin Włodarczak*[2], *Joakim Gustafson*[1], *Éva Székely*[1]

[1]Division of Speech, Music and Hearing, KTH Royal Institute of Technology, Stockholm, Sweden
[2]Department of Linguistics, Stockholm University, Sweden

kirkland@kth.se, wlodarczak@ling.su.se, jkgu@kth.se, szekely@kth.se

Abstract

Smiling during speech production has been shown to result in perceptible acoustic differences compared to non-smiling speech. However, there is a scarcity of research on the perception of "smiling voice" in synthesized spontaneous speech. In this study, we used a sequence-to-sequence neural text-to-speech system built on conversational data to produce utterances with the characteristics of spontaneous speech. Segments of speech following laughter, and the same utterances not preceded by laughter, were compared in a perceptual experiment after removing laughter and/or breaths from the beginning of the utterance to determine whether participants perceive the utterances preceded by laughter as sounding as if they were produced while smiling. The results showed that participants identified the post-laughter speech as smiling at a rate significantly greater than chance. Furthermore, the effect of content (positive/neutral/negative) was investigated. These results show that laughter, a spontaneous, non-elicited phenomenon in our model's training data, can be used to synthesize expressive speech with the perceptual characteristics of smiling.
Index Terms: speech synthesis, text-to-speech, smiling voice, smiled speech

1. Introduction

There are many well-documented functions of smiling in interpersonal communication. A smile can influence a speaker's perceived desire for cooperation[1] as well as their perceived trustworthiness [2], competence [3], extroversion, sympathy, kindness, and attractiveness [4]. And smiling is not merely a visual phenomenon–it creates changes in speech that can be perceived by listeners. The features associated with smiling voice include greater pitch height and pitch range [5, 6], and higher formant frequencies for some vowels [7, 8]. These audible characteristics allow smiling voice to mirror at least some of the social functions of smiling even in the absence of visual cues (e.g., conveying trustworthiness in virtual agents [9]).

With the advancements of conversational AI allowing for more nuanced interactions than ever before [10], synthetic voices of conversational agents need to become more realistic and versatile, displaying character, and complex conversational capabilities. In the area of expressive speech synthesis, there has been a relatively recent shift of research interest from synthesizing speech reflecting specific emotion categories or dimensions, towards unsupervised approaches of synthesizing "speaking styles" [11, 12]. These data-driven approaches have benefited from the availability of audiobooks, which contain a higher variability of speaking styles than traditional TTS corpora, but are lacking explicit annotation found in corpora of specifically recorded emotional speech. Less attention has been given to the interpretability and perceptual effect of synthesized styles using unsupervised methods. On the other end of the spectrum, various systems have been proposed for interpretable and intuitive control of prosodic features such as melody, rhythm, [13], pitch range, phone duration and spectral tilt [14]. However, for more stylistic characteristics, the gap between controllability and interpretability still remains to be closed. Thanks to recent advances in deep learning which have resulted in more robust systems both in text-to-speech and in speech processing tools for annotation and segmentation, spontaneous speech synthesis has made a leap forward in terms of naturalness and appropriateness for certain contexts [15]. As corpora of spontaneous speech have become available targets for text-to-speech, we are no longer restricted to modeling speaking styles in audiobooks, which are mostly a result of colorful reading, such as the speaker imitating characters. Real-world spontaneous speech data contains a myriad of speech phenomena that reveal the speaker's cognitive state, attitude stance, etc., which are represented in a variety of acoustic-prosodic and segmental features. Much of the research in spontaneous speech synthesis to date has been focused on modeling and understanding the use of hesitations such as *uh* and *um* [16, 17] and breathing [18], with many styles and phenomena left to be explored, both in terms of synthesis and perception.

This paper focuses on synthesizing a specific voice style, namely amused speech following laughter in a spontaneous monologue, which we refer to here as "smiling voice". We propose a context-driven method for synthesizing speech following laughter, using state-of-the art neural TTS built entirely from spontaneous conversational speech. In the training data, laughter (short affect burst) is not explicitly elicited, emerging as part of the spontaneous delivery contributing to the narrative. The perceptual effect of smiling voice is explored in different contexts, using sentences with positive, negative and neutral sentiment.

2. Related work

While much of the research on smiling voice has involved naturally produced speech, there have been a few investigations of smiling voice in synthesized speech. Lasarcyk and Trouvain [19], for example, synthesized four different German vowels using articulatory synthesis, and applied combinations of three different parameters to these vowels which correspond to the effects of smiling on articulation: raised f0, spread lips, and raised larynx. They found that higher f0 resulted in a greater degree of perceived smiling for all vowels. Both spread lips and a raised larynx influenced vowel formant frequencies as well as the perception of smiling, but this effect was different for different vowels. The vowels /a:/ and /y:/ were perceived as more smiley when synthesized with spread lips, while /i:/ showed no difference and /u:/ was considered less smiley with spread lips. The vowels /a:/ and /i:/ were perceived as more smiley with a raised larynx, but this parameter had no effect on the other vow-

10.21437/SSW.2021-19

els.

Another approach [20] used HMM-based synthesis to allow for a controllable degree of smiling in synthesized speech. Two models were created using recordings of neutral and smiling speech from one actor. For the recordings of smiling speech, the actor was instructed to smile and "sound happy" but not to laugh. A new model with controllable degrees of smiling was created by using a weighted-sum interpolation between the neutral and smiling models, with a degree of smile that varied according to the weights used. The evaluation showed that higher weights resulted in synthesized speech that was perceived as smiling to a greater degree, but also less natural.

In terms of synthesizing amused or happy-sounding speech, generating laughter is another important issue. Some previous approaches attempting to combine laughter and smiling voice have synthesized these two components independently from one another and then combined them. An HMM-based approach in [21], for example, inserted vowels produced while laughing into smiling speech generated with a different method, and the approach of [22] inserted phrase-sized "affect bursts" using concatenative speech synthesis. A more recent effort to synthesize laughter [23] employed a sequence-to-sequence neural text-to-speech system, with the goal was to create natural-sounding laughter which could then be integrated with a model for smiling speech.

In contrast to the approaches described above, our method of producing smiling voice neither explicitly manipulates acoustic parameters, nor does it use data that was explicitly elicited while smiling. Rather, we employ a context-driven approach on spontaneous data, generating smiling voice by synthesizing speech following laughter in one integrated model.

3. Database and synthesis

3.1. Spontaneous speech corpus

The TTS corpus was created from the audio recordings of the Trinity Speech-Gesture Dataset (TSGD) [24], which is comprised of 25 impromptu monologues by a male actor, on average 10.6 minutes long. The recordings were performed over multiple recording sessions by a male speaker of Irish English. The actor is speaking in a colloquial style, spontaneously and without interruption on topics such as hobbies, daily activities, and interests. During the monologues, he addresses a person seated behind the cameras who is giving visual, but no verbal feedback. Because a large part of the monologues involve story-telling, the actor often engages in retelling entertaining anecdotes, which naturally elicit laughter followed by the impression of amused, smiling voice, the synthesis of which is the focus of the current paper.

3.2. Annotation

To create a TTS corpus, the recording was transcribed using ASR and subsequently manually corrected to contain as few errors as possible, and to ensure that all filler words are accurately transcribed. In order to maximize the utterance length in the corpora and to enable insertion of inhalation breaths in the TTS, we used a data augmentation method called *breathgroup bigrams*, which essentially consists of segmenting a speech corpus into stretches of speech delineated by breath events, and then combining these breath groups in an overlapping fashion to form utterances no longer than 11 seconds [18] (see Figure 1). This method also makes it possible to learn contextual information beyond respiratory cycles during TTS training. Aside

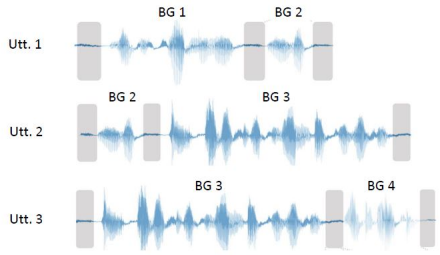


Figure 1: *Illustration of the breathgroup-bigram utterance structure [18] applied to create the TTS corpus from continuous recordings of spontaneous speech. Breath events are highlighted in grey.*

from filled pauses such as *uh* and *um*, the ASR transcription was enhanced with manual annotation of laughter, style breaks and silent pauses, the latter indicated with a comma. Both the filled pauses and laughter were transcribed using ARPABET phones. No new characters were introduced outside the standard. If a laughter involved ingressive airstream and was directly followed by more speech, the last voiced inhalation was annotated as breath event.

3.3. Systems

Two systems were trained using the sequence-to-sequence neural TTS engine Tacotron 2 [25]. The first system uses the standard Tacotron 2 architecture. The second system implements an utterance-level prosody control method, similar to [14], to be able to direct f0 and speech rate at inference. Speech rate (syllables/second) over the utterance and mean f0 are normalised, aligning the 1st and the 99th percentile points of the data to -1 and 1 respectively, and allowing outliers to go outside of that range. Normalized values for both features are appended to each utterance's encoded text and passed to the attention and decoder blocks from the pre-trained model. In order to fit the additional features, the input dimension to the attention, LSTM, projection and gate layers in the decoder are expanded. The additional weights added to the model are initialized with zero values. As such, at the start of the training the model evaluates as the pre-trained model. This method allows for directing mean f0 and speech rate on utterance level based on the natural distribution of these features in the corpus, as opposed to direct manipulation.

We used a PyTorch implementation of Tacotron 2[1], training each voice using transfer learning for 200k iterations on top of a pre-trained model trained on the LJ speech corpus [26]. Transfer learning based on a model trained on a large read-speech corpus has been shown to improve the quality of spontaneous speech synthesis [15]. For vocoding, the pre-trained universal model of WaveGlow [27] was fine-tuned for 290k iterations.

3.4. Synthesis of smiling voice

Our hypothesis is that due to the natural occurrence of laughter in the spontaneous speech corpus, synthesizing a laughter token followed by a breath event will result in an amused speaking style, characteristic of smiling voice in the subsequent speech.

[1] https://github.com/NVIDIA/tacotron2

Our reasoning is that the presence of smiling in speech that follows laughter introduces acoustic differences from comparable speech sounds not preceded by laughter, and accurately reproducing these differences will reduce the loss function (MSE) used to train the synthesizer. The synthesizer's ability to achieve these loss-function reductions and "remember" when to produce smiling speech also across a breath likely relies primarily on the encoder (rather than the acoustic memory offered by the autoregression and the LSTM in the decoder), since the encoder contains several CNN layers ideal for learning short-range dependencies and operates on the phone level, where the smiling token and the next speech sound are adjacent. To gain an insight into the perceptual effect of this method, the two systems were used to create two different conditions of synthesizing smiling voice. The baseline original Tacotron 2 architecture was used in the first condition, which we call *unconstrained*, because it allows the system to use the proximity of laughter to influence the rendering without any further constraints. The second condition employs our prosody-controllable architecture. During inference, we set both the normalized mean f0 and speech rate values to 0, in order to assess whether smiling voice can still be elicited while directing the system to render a realization close to the median of the distribution in the corpus for these two prosodic features. Hence, we call this condition *constrained*. We propose this method to help isolate other acoustic-prosodic features characteristic to smiling speech, to be able to assess their perceptual impact.

4. Evaluation

4.1. Stimuli

The samples for this experiment were synthesized from 36 utterances that stated an opinion. Twelve of each type of statement was used: positive (e.g., "I agree with that"), negative (e.g., "I don't really agree with that idea") and neutral (e.g., "It's fine with me either way"). These utterances were then synthesized with the constrained system and the unconstrained system, both preceded by laughter and without laughter. This resulted in a total of 144 stimuli with combinations of 3 different parameters: model (constrained/unconstrained), context (laughter/no laughter) and content (positive/negative/neutral). Laughter and inhalation breaths were removed from the beginning of each utterance, as we were interested in whether the utterances themselves would carry the perceptual characteristics of smiling and did not want the participants to base their judgments on whether or not they heard laughter.

4.2. Acoustic-prosodic analysis

The evaluation samples were analyzed for a number of acoustic and prosodic features to determine whether they differed between model (constrained vs. unconstrained), context (post-laughter vs. no laughter), and/or content (positive/negative/neutral). Speech rate (syl/sec), mean f0, and f0 variation were measured and compared for the four different combinations of model and context. In addition, to compare the conditions in terms of breathiness, we calculated median smoothed cepstral peak prominence (CPPS, [28]) of all voiced frames in an utterance. CPPS quantifies strength of the first harmonic relative to the regression line over the power cepstrum, with high values corresponding to more modal voice and low values indicating breathiness. No significant difference was found in mean f0. However, analysis of variance showed a significant main effect of context on f0 variation. f0 variation (as measured by the standard deviation of f0 per utterance) was higher for speech following laughter (M=14.49, SD=5.56) than for speech synthesized without laughter (M=12.27, SD=5.61), $F(1,33) = 4.61$, $p < 0.05$. There was also a main effect of model on speech rate. The samples synthesized with the unconstrained model had a higher speech rate (M=5.15, SD=0.96) than samples synthesized with the constrained model (M=5.08, SD=1.01), $F(1,33) = 7.67$, $p < 0.05$. Finally, analysis of variance showed a significant effect of model on CPPS in voiced segments. Samples synthesized with the constrained model had a higher CPPS (M=12.35, SD=1.70) than samples synthesized with the unconstrained model (M=11.94, SD=1.57), $F(1,33) = 6.86$, $p < 0.05$.

4.3. Naturalness test

The systems were assessed for naturalness based on a web-based MUSHRA-like listening test. The test involved four versions of each utterance side by side (post-laughter/constrained, post-laughter/unconstrained, no laughter/constrained and no laughter/constrained) in randomized order with a scale for each item that ranged from 1 (very unnatural) to 5 (very natural).

4.4. Pairwise listening test

The extent to which post-laughter speech sounded like smiling was evaluated with a web-based forced-choice audio discrimination task. In one version of the test, stimuli synthesized with the constrained model were used. The other version used stimuli synthesized with the unconstrained model. Otherwise the setup was identical: smiling and non-smiling versions of each of the 36 utterances were presented side by side and the task was to choose which of the two versions sounded the most as if the speaker was smiling. The samples could be played as many times as needed. The order in which the two versions were displayed was randomized, as was the order of the utterances. The TTS samples used in the experiments are available here: https://www.speech.kth.se/tts-demos/ssw2021smiling

5. Results

5.1. Naturalness test

Twenty-one participants recruited online via Prolific completed the test. 54.38% were female and 47.62% were male. A within-subjects factorial analysis of variance showed that there was no main effect of content (positive/negative/neutral), context (post-laughter/no laughter) or model (constrained/unconstrained) on how natural-sounding participants rated the stimuli. The interaction between model and content was significant, $F(2,19) = 5.62$, $p < 0.05$, however, simple main effects of content were not significant for either the constrained or the unconstrained model. Results are summarized in Figure 2.

5.2. Pairwise listening test

A total of 60 participants were recruited via Prolific, of which 55.9% were female and 44.1% were male. All participants were native speakers of English. Half (30) received the unconstrained version of the task while the other half received the constrained version. One participant from the unconstrained group was excluded from the final analysis because their completion time was over 4 standard deviations above the mean.

Participants who heard stimuli synthesized with the unconstrained model identified the post-laughter synthesized speech as smiling 67.62% of the time, while participants who heard

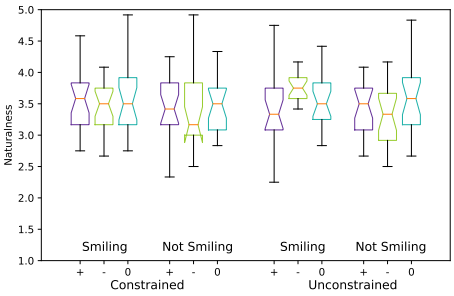


Figure 2: *Results of MUSHRA-like naturalness test on conversational utterances with positive (+), negative (-) and neutral (0) linguistic content.*

stimuli created with the constrained model identified post-laughter speech as smiling at a rate of 62.55%. Single-sample t-tests showed that this rate was significantly higher than chance for both participants who heard stimuli synthesized with the prosody-constrained model, $t(29) = 5.01$, $p < 0.001$, and those who rated stimuli synthesized with the unconstrained model, $t(28) = 10.26$, $p < 0.001$.

A mixed factorial analysis of variance was carried out to investigate the effect of content (positive/negative/neutral) and model (constrained/unconstrained) on the rate of identifying post-laughter speech as smiling. There were significant main effects of both content ($F(2,56) = 44.25$, $p < 0.001$) and model ($F(1,57) = 6.97$, $p < 0.05$). Participants who heard the prosodically unconstrained samples rated the post-laughter speech as smiling more often (M=67.62, SD=14.29) than those who heard the prosodically constrained samples (M=60.46, SD=17.33). In addition, participants were more likely to rate utterances that stated positive opinions as smiling (M=74.44,SD=12.17) compared to negative (M=65.96,SD=16.76) and neutral statements (M=51.55,SD=17.40). Post-hoc tests with the Bonferroni correction showed that the differences between these means were all significant ($p < 0.01$).

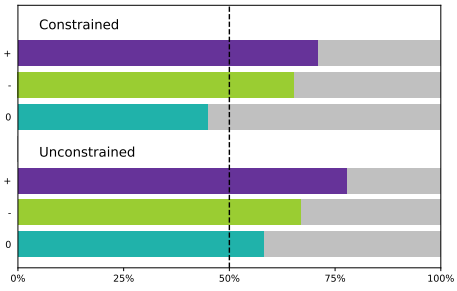


Figure 3: *Results of pairwise listening test, with coloured bars representing correct identification of smiling voice for utterances of positive (+), negative (-) or neutral (0) linguistic content produced by each model.*

6. Discussion

As hypothesized, it appears that speech following synthesized laughter is perceived as smiling, showing that the voice style we referred to as "smiling voice" conveys some of the perceptual aspects of smiling. The content of the utterances (whether they stated a positive, negative or neutral opinion) seemed to play a role in how participants performed at classifying post-laughter speech as smiling: listeners found it easier to discriminate between smiling voice and non-smiling voice when the content of the utterance was positive. Note that due to the use of a forced choice test in our evaluation, this does not mean that positive linguistic content increases the likelihood of perceived amusement in speech, but rather that it improves discrimination between two utterances on the basis of perceived amusement. This may indicate that, as a consequence of the context-driven approach, the TTS system was better at generating utterances that sounded like smiling when the content was positive. An alternative explanation would be that there is an effect of congruence between content and perceived emotional valence, whereby participants had an easier time distinguishing between smiling and non-smiling speech when the content and expressive characteristics of the synthesized smiling speech matched. However, participants had the most difficult time distinguishing between smiling and non-smiling speech when the linguistic content was neutral, which makes the first possibility more plausible.

Unlike in some previous studies, smiling speech in this case was not perceived as less natural than non-smiling speech. There appears to have been some joint effect of model and content on perceived naturalness, but since the differences in naturalness between positive, negative and neutral content were not significant with either model, this is difficult to interpret. The takeaway is that smiling voice synthesized with our method did not sound less natural.

In terms of acoustic/prosodic features, there were some differences between the constrained and unconstrained model. The unconstrained model produced breathier speech with a higher speech rate. However, these differences did not, in turn, seem to affect discrimination between smiling and non-smiling voice. Although participants who listened to samples from the unconstrained model did have an easier time with the discrimination task, there was no association between their performance and the parameters on which the models differed. Rather, f0 variation seems to have had the largest impact on performance at the discrimination task independent of model, consistent with previous findings that f0 variation is higher in naturally produced smiling speech [5, 6].

7. Conclusions

By synthesizing speech following laughter, we were able to exploit a spontaneous phenomenon in our models' training data to create the impression of smiling, without affecting the naturalness of the speech signal. This was the case even in a prosody-constrained model that restricted f0 and speech rate variation towards the median in the corpus, although listeners found the discrimination task more challenging with this model. Due to the context-driven nature of our method it seems that the linguistic content of the utterances affected the ease of discriminating between smiling and non-smiling speech. It is not entirely clear, whether this is due to the smiling speech sounding more like smiling when synthesized from utterances that suggest agreement, the non-smiling speech sounding less like smil-

ing in this context, both, or some other difference between the stimuli that made the discrimination task easier by making the stimuli sound more dissimilar.

The fact that the mere proximity of synthesized speech to synthesized laughter can create an impression of smiling means that it may not be necessary to synthesize laughter and smiling speech independently, as previous approaches suggest. Integrated into a conversational system equipped with voice style management modules, this approach could both create smiling voice that emerges in the context of laughter, and standalone amused speech (where the synthesized laughter is masked in the output), thereby improving the dialogue systems' capability to engage in informal social interactions.

8. Acknowledgements

This research is supported by the Swedish Research Council projects: Perception of speaker stance – using spontaneous speech synthesis to explore the contribution of prosody, context and speaker (VR-2020-02396), Connected: context-aware speech synthesis for conversational AI (VR-2019-05003), Prosodic functions of voice quality dynamics (VR-2019-02932), and the Riksbankens Jubileumsfond project CAPTivating – Comparative Analysis of Public speaking with Text-to-speech (P20-0298).

9. References

[1] L. Johnston, L. Miles, and C. N. Macrae, "Why are you smiling at me? social functions of enjoyment and non-enjoyment smiles," *British Journal of Social Psychology*, vol. 49, no. 1, pp. 107–127, 2010.

[2] K. Schmidt, R. Levenstein, and Z. Ambadar, "Intensity of smiling and attractiveness as facial signals of trustworthiness in women," *Perceptual and motor skills*, vol. 114, no. 3, pp. 964–978, 2012.

[3] K. Krys, C.-M. Vauclair, C. A. Capaldi, V. M.-C. Lun, M. H. Bond, A. Domínguez-Espinosa, C. Torres, O. V. Lipp, L. S. S. Manickam, C. Xing *et al.*, "Be careful where you smile: Culture shapes judgments of intelligence and honesty of smiling individuals," *Journal of nonverbal behavior*, vol. 40, no. 2, pp. 101–116, 2016.

[4] E. Otta, F. F. E. Abrosio, and R. L. Hoshino, "Reading a smiling face: Messages conveyed by various forms of smiling," *Perceptual and motor skills*, vol. 82, no. 3_suppl, pp. 1111–1121, 1996.

[5] C. Émond, L. Ménard, M. Laforest, F. Bimbot, C. Cerisara, C. Fougeron, G. Gravier, and L. Lamel, "Perceived prosodic correlates of smiled speech in spontaneous data." in *INTERSPEECH*, 2013, pp. 1380–1383.

[6] V. C. Tartter, "Happy talk: Perceptual and acoustic effects of smiling on speech," *Perception & psychophysics*, vol. 27, no. 1, pp. 24–27, 1980.

[7] I. Torre, "Production and perception of smiling voice," in *Proceedings of the First Postgraduate and Academic Researchers in Linguistics at York (PARLAY 2013) Conference. York, UK.*, 2014.

[8] M. Keough, A. Ozburn, E. K. McClay, M. D. Schwan, M. Schellenberg, S. Akinbo, and B. Gick, "Acoustic and articulatory qualities of smiled speech," *Canadian Acoustics*, vol. 43, no. 3, 2015.

[9] I. Torre, J. Goslin, and L. White, "If your device could smile: People trust happy-sounding artificial agents more," *Computers in Human Behavior*, vol. 105, p. 106215, 2020.

[10] Y. Zhang, S. Sun, M. Galley, Y.-C. Chen, C. Brockett, X. Gao, J. Gao, J. Liu, and B. Dolan, "Dialogpt: Large-scale generative pre-training for conversational response generation," *arXiv preprint arXiv:1911.00536*, 2019.

[11] Y. Wang, D. Stanton, Y. Zhang, R.-S. Ryan, E. Battenberg, J. Shor, Y. Xiao, Y. Jia, F. Ren, and R. A. Saurous, "Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis," in *International Conference on Machine Learning*. PMLR, 2018, pp. 5180–5189.

[12] Y.-J. Zhang, S. Pan, L. He, and Z.-H. Ling, "Learning latent representations for style control and transfer in end-to-end speech synthesis," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6945–6949.

[13] R. Valle, J. Li, R. Prenger, and B. Catanzaro, "Mellotron: Multispeaker expressive voice synthesis by conditioning on rhythm, pitch and global style tokens," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6189–6193.

[14] T. Raitio, R. Rasipuram, and D. Castellani, "Controllable neural text-to-speech synthesis using intuitive prosodic features," in *Proc. Interspeech*, 2020, pp. 4432–4436.

[15] É. Székely, G. E. Henter, J. Beskow, and J. Gustafson, "Spontaneous conversational speech synthesis from found data," in *Interspeech*, 2019, pp. 4435–4439.

[16] R. Dall, "Statistical parametric speech synthesis using conversational data and phenomena," Ph.D. dissertation, School of Informatics, The University of Edinburgh, Edinburgh, UK, 2017.

[17] É. Székely, G. E. Henter, J. Beskow, and J. Gustafson, "How to train your fillers: uh and um in spontaneous speech synthesis," in *Proc. SSW*, vol. 10, 2019, pp. 245–250.

[18] ——, "Breathing and speech planning in spontaneous speech synthesis," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7649–7653.

[19] E. Lasarcyk and J. Trouvain, "Spread lips+ raised larynx+ higher f0= smiled speech?-an articulatory synthesis approach," *Proceedings of ISSP*, pp. 43–48, 2008.

[20] K. El Haddad, H. Cakmak, A. Moinet, S. Dupont, and T. Dutoit, "An HMM approach for synthesizing amused speech with a controllable intensity of smile," in *2015 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*. IEEE, 2015, pp. 7–11.

[21] K. El Haddad, S. Dupont, J. Urbain, and T. Dutoit, "Speech-laughs: an hmm-based approach for amused speech synthesis," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 4939–4943.

[22] N. Campbell, "Conversational speech synthesis and the need for some laughter," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 4, pp. 1171–1178, 2006.

[23] N. Tits, K. E. Haddad, and T. Dutoit, "Laughter synthesis: Combining seq2seq modeling with transfer learning," *arXiv preprint arXiv:2008.09483*, 2020.

[24] Y. Ferstl and R. McDonnell, "Investigating the use of recurrent motion modelling for speech gesture generation," in *Proc. IVA*, 2018, pp. 93–98. [Online]. Available: https://trinityspeechgesture.scss.tcd.ie

[25] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan *et al.*, "Natural tts synthesis by conditioning wavenet on mel spectrogram predictions," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4779–4783.

[26] K. Ito, "The LJ Speech dataset," https://keithito.com/LJ-Speech-Dataset/, 2017.

[27] R. Prenger, R. Valle, and B. Catanzaro, "Waveglow: A flow-based generative network for speech synthesis," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 3617–3621.

[28] J. Hillenbrand and R. A. Houde, "Acoustic correlates of breathy vocal quality: Dysphonic voices and continuous speech," *Journal of Speech Language and Hearing Research*, vol. 39, no. 2, 1996.

11th ISCA Speech Synthesis Workshop (SSW 11)
26-28 August 2021, Budapest, Hungary

Voicy: Zero-Shot Non-Parallel Voice Conversion in Noisy Reverberant Environments

Alejandro Mottini, Jaime Lorenzo-Trueba*, Sri Vishnu Kumar Karlapati, Thomas Drugman*

Amazon.com

(amottini,truebaj,srikarla,drugman)@amazon.com

Abstract

Voice Conversion (VC) is a technique that aims to transform the non-linguistic information of a source utterance to change the perceived identity of the speaker. While there is a rich literature on VC, most proposed methods are trained and evaluated on clean speech recordings. However, many acoustic environments are noisy and reverberant, severely restricting the applicability of popular VC methods to such scenarios. To address this limitation, we propose Voicy, a new VC framework particularly tailored for noisy speech. Our method, which is inspired by the de-noising auto-encoders framework, is comprised of four encoders (speaker, content, phonetic and acoustic-ASR) and one decoder. Importantly, Voicy is capable of performing non-parallel zero-shot VC, an important requirement for any VC system that needs to work on speakers not seen during training. We have validated our approach using a noisy reverberant version of the LibriSpeech dataset. Experimental results show that Voicy outperforms other tested VC techniques in terms of naturalness and target speaker similarity in noisy reverberant environments.

Index Terms: voice-conversion, zero-shot, noisy reverberant environments

1. Introduction

Voice Conversion (VC) is the task of modifying an utterance from a source speaker to make it sound like it was uttered by a target speaker, while preserving the original linguistic content [1]. VC is a key component of many modern applications, including text-to-speech (TTS) [2], speech enhancement [3], and speaking assistance [4] systems. Due to its success in these fields, VC has been studied extensively in recent years [5].

However, despite their success in generating realistic samples, most current VC approaches have two important limitations. First, they are trained and evaluated on clean speech recordings, such as LibriSpeech [6] or VCTK [7]. This is a shortcoming, since most real acoustic environments are noisy and reverberant. Second, not all methods can perform non-parallel zero-shot conversion [8, 9]. These shortcomings limit their applicability to certain industrial use-cases, such as applying VC to noisy utterances captured by voice-controlled virtual assistants. In such scenarios, a production VC system should work well in more realistic acoustic conditions (noisy and reverberant) and be robust to microphones with different characteristics. In addition, VC methods should be capable of transforming utterances from speakers not seen during training, and be able to change the speaker's identity while preserving the quality and naturalness of the speech. Finally, a production VC system should be scalable and robust.

In this paper we propose Voicy, a new VC method that fulfils the desired characteristics outlined before. Our approach, based on de-noising auto-encoders [10] and the AutoVC model [8], is especially tailored for noisy reverberant speech, and capable of performing non-parallel zero-shot VC. Importantly, the proposed phonetic and acoustic-ASR encoders, an improvement over the original AutoVC formulation, significantly improves the intelligibility of the converted speech in noisy conditions. Moreover, since Voicy is based on auto-encoders, it is more robust and easier to train than other GAN- [9] and Flow-based [11] approaches.

To validate our approach, we have created a noisy reverberant version of the LibriSpeech dataset [6], and used it to train and test both our method and other selected baselines. Results show that Voicy outperforms other VC techniques in terms of naturalness and target speaker similarity in noisy reverberant environments. Converted speech samples are provided here [1].

2. Related Work

Based on the required training data, VC methods can first be characterized as parallel or non-parallel. Models in the first category [12, 13] depend on a training set of aligned speech pairs of source and target speakers uttering the same phrase. Conversely, non-parallel models only require source and target speaker's utterances, but they do not need to be aligned or match in terms of content. Non-parallel VC techniques can further be characterized as either zero-shot or not, depending on their ability to transform utterances of speakers unseen during training. Naturally, non-parallel zero-shot VC is the most challenging but valuable framework, and is therefore the focus of this work.

Regardless of the type of required data, the actual conversion technique behind each VC method varies. Some methods rely on traditional statistical approaches such as Gaussian Mixture models (GMM) [14], while newer techniques use Deep-Learning-based approaches. Within this family, different approaches exist, most notably, Generative-Adversarial-Network-based (GAN) [9, 15], Variational-Auto-Encoder-based (VAE) [16], Auto-Encoder-based [8], and Flows-based [11] models. Each family has its own trade-off between conversion quality, complexity and ease of training. In general, Auto-Encoder-based models appear to have the best trade-off [8]. In particular, [17] proposes a variational-autoencoder method conditioned on the phonetic contents of utterances, but it is not zero-shot, and was only evaluated on a small dataset of clean utterances.

Finally, a small body of work is dedicated to VC in noisy environments. In [18], a parallel exemplar-based VC model is proposed. Another approach is [19], where a speech-enhancement-based technique that applies two different filtering methods to suppress noise is proposed. Then, a tradi-

* Equal contribution

[1] https://github.com/alexa/amazon-voice-conversion-voicy

10.21437/SSW.2021-20

tional BLSTM-VC model [20] is used to convert the filtered utterances. Although successful, these approaches have several shortcomings, including their inability to perform zero-shot conversion, and the fact that the presented results are compared against relatively weak baselines (GMM-based VC).

3. Voicy: Our Proposed Method

3.1. General System Description

Voicy is comprised of 5 modules (see Figure 1), and uses two representations of an utterance: Mel-spectrogram and transcription, represented using phonemes.

Our speaker encoder follows [21]. We use a model pre-trained on VCTK [7], which remains fixed during training. In addition, our content encoder is inspired by AutoVC [8], but with minor modifications to the hyper-parameters (size of filters, etc.).

The phonetic encoder, with its architecture detailed in Figure 1, is responsible for encoding the sequence of phonemes into a sentence-level representation of the text. As such, the content encoder is not forced to be in charge of both maintaining the linguistic information and the prosody of the speech. The main effect of adding this component is having a specific module in charge of intelligibility, which improves the naturalness of the converted speech (see Section 4.2).

Finally, the ASR module, comprised of CNN and bi-LSTM layers (Figure 1), learns to predict the phonetic embeddings produced by the phonetic encoder, but working in the audio instead of the text domain. As such, once our model is trained, the phonetic encoder can be substituted by the ASR module, removing the need for the textual representation of the input utterance during inference. This makes the applicability of this approach in production more viable. As an alternative, one could also consider a system were a standard ASR model that automatically generates a textual representation of the utterances, which could then be encoded thanks to the phonetic encoder. However, to maximize performance, we have opted for our design that uses the best of both worlds: use transcriptions if available, or an ASR module when not. This motivated our choice of having 2 modules instead of just one.

The decoder's architecture follows [8], and is comprised of GRU and CNN layers. It receives the output of the speaker, content, and phonetic or ASR encoders (depending on the stage), and outputs the converted spectrogram. We can interpret its 3 inputs as: (1) what is being uttered (linguistic information captured by the phonetic/ASR embedding), (2) who uttered it (speaker identity captured by the speaker embedding), (3) how it is uttered (prosody information captured by the content encoder). Finally, the Universal WaveRNN-like Vocoder [22] is used to convert the produced Mel-spectrogram into a waveform.

3.2. Architecture Description

Let us first define the tuple (S, Z, A) representing speaker S, content (phonetic and prosodic) Z, and audio segment A. Let us now take two such tuples, (S_1, Z_1^i, A_1^i) and (S_2, Z_2^k, A_2^k), where the first corresponds to the i-th tuple of the source speaker 1, and the second to the k-th tuple of the target speaker 2. The goal of any VC system is to produce the output utterance $\hat{A}_{1\to2}^i$ that keeps the content of A_1^i, while changing the perceived speaker to S_2. Since we tackle the zero-shot VC problem, S_1 or S_2 do not need to be part of the training set.

To achieve this, Voicy uses five modules: speaker encoder $E_s(.)$, content encoder $E_c(.)$, phonetic encoder $E_{ph}(.)$, acoustic-ASR encoder $E_{ASR}(.)$, and decoder $D(.,.)$. Both the phonetic and acoustic-ASR encoders are improvements over AutoVC [8], which, along with the use of the de-noising auto-encoder technique, allow our model to perform VC in a noise-robust manner.

More concretely, given an utterance A, we represent it using two modalities: its Mel-spectrogram ML, and its transcription, represented using phonemes PH. For simplicity, we denote $ML = f_{ML}(A)$ and $PH = f_{ph}(A)$. Then, let us represent the input/output of each module as:

$$\begin{aligned} C = E_c(ML), U = E_s(ML) \\ R = E_{ASR}(ML), P = E_{ph}(PH) \\ \hat{A}_{.\to.} = D(C, U, R, P) \end{aligned} \tag{1}$$

where D receives either R or P, but not both. During training (Figure 1), we use inputs (S_1, Z_1^i, A_1^i) and (S_1, Z_1^j, A_1^j), representing two different utterances from the same speaker S_1. Following the de-noising auto-encoder methodology, and leveraging the parallel corpus of clean and noisy data we constructed (see Section 4), we also consider tuple $(S_1, Z_1^i, \tilde{A_1^i})$, where $\tilde{A_1^i}$ is a clean version of A_1^i, always available during training. Then, for each training utterance A_1^i containing either clean, noisy, or noise reverberant speech, the model produces $C_1^i = E_c(f_{ML}(A_1^i))$, $U_1^j = E_s(f_{ML}(A_1^j))$, $R_1^i = E_{ASR}(f_{ML}(A_1^i))$, $P_1^i = E_{ph}(f_{PH}(A_1^i))$ and $\hat{A}_{1\to1}^i = D(C_1^i, U_1^j, P_1^i)$. We then compute the total loss:

$$\begin{aligned} L = L_{recon} + \beta L_{phonetic} + \lambda L_{content} \\ L_{recon} = \|\hat{A}_{1\to1}^i - \tilde{A_1}^i\|_2 \\ L_{phonetic} = \|R_1^i - P_1^i\|_1 \\ L_{content} = \|E_c(\hat{A}_{1\to1}^i) - C_1^i\|_1 \end{aligned} \tag{2}$$

with λ and β hyper-parameters of the model. L_{recon}, $L_{phonetic}$ and $L_{content}$ are loss functions computed using the inputs and outputs of the difference modules, and are presented in Figure 1. Speaker encoder $E_s(.)$ is assumed to be pre-trained and remains fixed during training. The model only sees utterances from a single speaker (when working in batches, many speakers are used) during training, and the reconstruction loss is always computed between the output of the decoder and the clean version of the input. This teaches the encoders to be robust to noise.

Once trained, the model can convert utterances from source to target speakers. For this step (see Figure 1), let us again consider tuples of two speakers (S_1, Z_1^i, A_1^i) and (S_2, Z_2^k, A_2^k). We then use the trained modules to compute: $C_1^i = E_c(f_{ML}(A_1^i))$, $U_2^k = E_s(f_{ML}(A_2^k))$, $R_1^i = E_{ASR}(f_{ML}(A_1^i))$ and $\hat{X}_{1\to2}^i = D(C_1^i, U_2^k, R_1^i)$. As we can see, the phonetic encoder E_{ph} is no longer needed, and E_{ASR} takes its place since it learned how to approximate its behavior. As such, we can convert utterances for which we do not have the transcription. Moreover, the speaker encoder E_s now receives an utterance from the target speaker S_2, and its output is fed to the decoder to reconstruct the speech as if it was uttered by S_2.

4. Experimental Validation

4.1. Experimental Protocol

Using LibriSpeech as the basis, we have first created reverberant utterances using the Aachen Impulse Response (AIR) Database

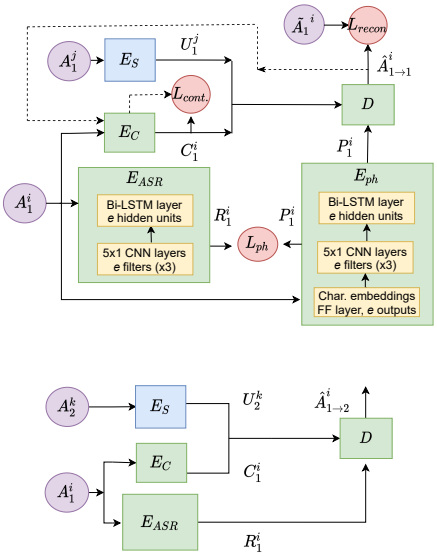


Figure 1: *Voicy during training (top) and test (bottom) phase. For simplicity, $f_{ML}(A)$ and $f_{ph}(A)$ are not shown. The speaker encoder E_S (in blue) is pre-trained and fixed during training.*

[23], which contains over 200 room impulse responses for diverse settings. Due to the sizes and properties of the rooms, the reverb can be significant, with reverberation time (T60) ranging from 0.12 to 1.25 seconds. For this set of transformations, no noise is added. In addition, we have created a second set of utterances containing both noise and reverberation using Pyroomacoustics [24], a room acoustics simulation package. Using this simulator, we created 3D shoebox rooms of different sizes, trying to approximate the dimensions of realistic home rooms. Then, for each room configuration, we have added 3 audio sources: (1) clean LibriSpeech utterance, (2) white noise of different levels, and (3) external noises selected at random from the DEMAND dataset [25]. This dataset is comprised of real-world noise from a variety of settings. The position of these 3 sources, along with the position of the (virtual) microphone, are varied randomly for each room/utterance. The resulting utterances have on average less reverberation than the previous set (due to the room size), but have two noise sources.

Both approaches are applied to LibriSpeech train-clean-100 (250 speakers, 100 hours), train-clean-360 (920 speakers, 360 hours) and dev-clean (40 speakers, 5.3 hours). Both train-clean-100 and train-clean-360, along with their reverberant and noisy reverberant versions, are combined into a single training set. To create the evaluation set (see Section 4.2), we first combine the dev-clean and its reverberant and noisy-reverberant versions. We then randomly select a subset of 400 utterances, covering all speakers and noise levels (clean, reverb, and noisy-reverb). The SNR distributions in both training and test sets match, with a maximum SRN of 35 dB (for the clean Librispeach audios), a minimum of -2.2 dB, and an average of 16 dB. All audio has a sampling frequency of 24kHz. Finally, we chose 2 random target speakers from the training set, one female and one male, and used each VC model to transform the 400 utterances into them.

Mel-spectrograms are extracted using the LibRosa library [26], with 80 coefficients and frequencies ranging from 50 Hz to 12 kHz. To obtain the phonetic representation, we have used the transcription provided with LibriSpeech, and the Montreal Forced Aligner [27].

As baselines, we have considered four methods besides Voicy (referred to as Proposed in Section 4.2). First, AutoVC [8], which inspired our work. In addition, to compare our model against a speech-enhancement-based approach, we trained AutoVC on the original clean LibriSpeech, and used it to transform de-noised de-reverbed test utterances obtained by applying the LogMMSE [28] and the Weighted Prediction Error [29] speech enhancement methods to our noisy evaluation set. This methodology is referred to as Preproc. Moreover, we considered StarGAN-VC [9], an established GAN-based VC model. StarGAN-VC uses a one-hot representation of the speakers, and needs to see utterances from both training and test set speakers during training. To address this problem, we have taken 10% of the test set and added it to StarGAN-VC training set. This 10% is not included in the evaluation set of the methods. Finally, we perform a simple ablation study by removing the acoustic-ASR encoder from our architecture and using this model as a baseline. We refer to this approach as Phonetic. Since this variant uses the phonetic encoding during inference, and is fed the ground-truth transcription, its performance should provide an upper bound of the metrics. We have used Wilcoxon signed-rank test to do pairwise comparisons between the different approaches. Hyper-parameters were optimized using random search to maximize the perceived quality of samples under informal listening.

To quantify the performance of the methods, we run two perceptual evaluations inspired by past voice conversion challenges [1], looking at Naturalness and speaker Similarity. The evaluations were crowd-sourced on Amazon Mechanical Turk, and designed according to MUltiple Stimuli with Hidden Reference and Anchor (MUSHRA) [30], but without forcing any system to be rated as 100. Each evaluator rated 20 screens, and selected the Naturalness and Similarity to the target speaker using a 0 to 100 scale. In each evaluation screen, listeners were presented with samples from the 5 systems, and with recordings of the target speaker as hidden reference. A different random recording of the target speaker was provided as explicit reference. We collected 3 scores per utterance, for both target speakers.

4.2. Results

We first evaluated the methods in clean acoustic conditions (original LibriSpeech data, no noise nor reverberation). Results (not presented for brevity) showed that the performance difference between the methods is minor for both metrics, with Phonetic outperforming the others due to its ability to leverage the phonetic information available during inference. In this scenario, Voicy does not outperform the other methods.

However, when analyzing the results for the reverberant (Figure 2) and noisy reverberant utterances (Figure 3), we observe a clear difference between our systems and the baselines. Once again, Phonetic outperforms all others for both metrics, while the Proposed approach significatively outperforms the remaining baselines in terms of Naturalness on both reverberant and noisy reverberant conditions, likely due to the added information provided by the acoustic encoder during inference. Re-

sults are more mitigated for Similarity due to the high variance of the results for all methods. The relatively large variances in performance is source/target dependent, but also on the level and type of noise. Models that handle noise better have significant less variance. Moreover, the use of Mechanical Turkers can add significant variability to the scores, even if precise instructions are given to them. Nevertheless, statistical analyses of the pair-wise comparisons between the systems show that the only non-significant improvement between Voicy and the baselines is for Similarity for reverberant conditions. For example, when comparing Proposed and Star-GAN, we observe $p = 1.35E-5$ and $p = 8.46E-6$ for Naturalness in reverberant and noisy-reverberant conditions respectively, and $p = 0.21$ and $p = 0.038$ for Similarity. This could be due to the fact that the added phonetic context does not add any speaker disentangling information. In addition, we observe how AutoVC's performance can improve by applying speech-enhancement techniques as pre-processing (Preproc), particularly for the noisy reverberant utterances. The reader is encouraged to listen to converted samples in different noise conditions (see Section 1).

Moreover, we have estimated the SNR of the utterances using the Pysepm toolbox [2], and analyzed the performance of the models for different noise levels. Results are presented in Figure 4) for a SNR range of interest (noisy conditions). Results show that our method outperforms the others (except Phonetic) in terms of Naturalness until 8 dB, and until 5 dB for Similarity. For higher levels, Star-GAN and AutoVC appear to match or slightly outperform Voicy, which could be due to the fact the added phonetic encoder does not contribute to the speech reconstruction in clean acoustic conditions. We also observe a degradation in performance for all of the systems at 5 dB SNR. We believe that during the random process of adding noise, more samples fell into the 5db noise bucket. The larger number of speakers and samples might have affected the performance of all models equally.

Overall, results concur in showing that Voicy outperforms the baselines in terms of Naturalness and Similarity in noisy reverberant environments, except for Phonetic, which has an "unfair" advantage during inference. The Proposed approach, with its acoustic encoder, tries to match its performance without the need for the phonetic representation during inference, but is unable to reach this upper bound. Nonetheless, it outperforms the other baselines. We believe these results could be improved using a better pre-trained ASR module.

5. Conclusions

In this work we presented Voicy, a new VC model designed for use-cases that require noise/reverb robustness as well as zero-shot transformation, which is not possible with current VC approaches. Our architecture is comprised of five modules, including a phonetic and an acoustic-ASR encoder, which help improve the intelligibility of the converted speech in noisy environments. We have created a noisy reverberant version of the LibriSpeech dataset, and used to train and test both our method and four other baselines. Results show that Voicy outperforms the baselines in terms of naturalness and speaker similarity in noisy reverberant environments. In the future, we will improve our acoustic-ASR encoder, and study if there is prosodic leakage in the phonetic embedding.

[2] https://github.com/schmiph2/pysepm

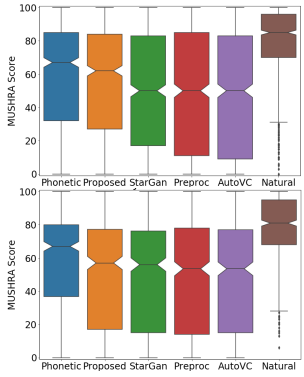


Figure 2: *Naturalness (top) and Similarity (bottom) of the systems for reverberant utterances.*

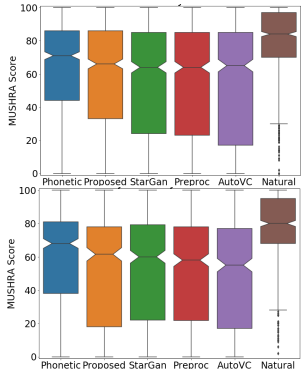


Figure 3: *Naturalness (top) and Similarity (bottom) of the systems for noisy reverberant utterances.*

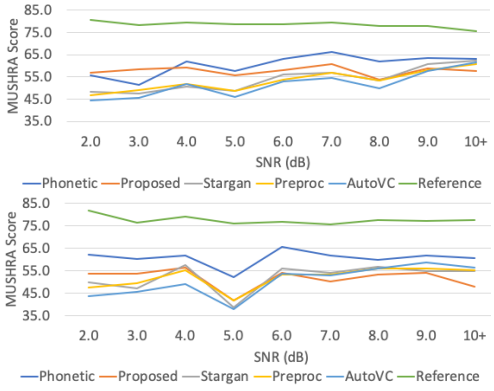


Figure 4: *Average score for Naturalness (top) and Similarity (bottom) by SNR.*

6. References

[1] J. Lorenzo-Trueba, J. Yamagishi, T. Toda, D. Saito, F. Villavicencio, T. Kinnunen, and Z. Ling, "The voice conversion challenge 2018: Promoting development of parallel and nonparallel methods," *arXiv preprint arXiv:1804.04262*, 2018.

[2] M. Cotescu, T. Drugman, G. Huybrechts, J. Lorenzo-Trueba, and A. Moinet, "Voice conversion for whispered speech synthesis," *IEEE Signal Processing Letters*, 2019.

[3] T. Toda, M. Nakagiri, and K. Shikano, "Statistical voice conversion techniques for body-conducted unvoiced speech enhancement," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 9, pp. 2505–2517, 2012.

[4] K. Nakamura, T. Toda, H. Saruwatari, and K. Shikano, "Speaking-aid systems using gmm-based voice conversion for electrolaryngeal speech," *Speech Communication*, vol. 54, no. 1, pp. 134–146, 2012.

[5] S. H. Mohammadi and A. Kain, "An overview of voice conversion systems," *Speech Communication*, vol. 88, pp. 65–82, 2017.

[6] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 5206–5210.

[7] J. M. K. Veaux, Christophe; Yamagishi, "Cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit," 2017.

[8] K. Qian, Y. Zhang, S. Chang, X. Yang, and M. Hasegawa-Johnson, "Autovc: Zero-shot voice style transfer with only autoencoder loss," in *Proc. of the 36th International Conference on Machine Learning*, vol. 97, 2019, pp. 5210–5219.

[9] H. Kameoka, T. Kaneko, K. Tanaka, and N. Hojo, "Stargan-vc: Non-parallel many-to-many voice conversion using star generative adversarial networks," in *2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2018, pp. 266–273.

[10] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proc. of the 25th international conference on Machine learning*, 2008, pp. 1096–1103.

[11] J. Serrà, S. Pascual, and C. S. Perales, "Blow: a single-scale hyperconditioned flow for non-parallel raw-audio voice conversion," in *Advances in Neural Information Processing Systems*, 2019.

[12] M. Abe, S. Nakamura, K. Shikano, and H. Kuwabara, "Voice conversion through vector quantization," *Journal of the Acoustical Society of Japan (E)*, vol. 11, no. 2, pp. 71–76, 1990.

[13] R. Takashima, T. Takiguchi, and Y. Ariki, "Exemplar-based voice conversion in noisy environment," in *Spoken Language Technology Workshop (SLT)*. IEEE, 2012, pp. 313–317.

[14] Y. Stylianou, O. Cappé, and E. Moulines, "Continuous probabilistic transform for voice conversion," *IEEE Transactions on speech and audio processing*, vol. 6, no. 2, pp. 131–142, 1998.

[15] F. Fang, J. Yamagishi, I. Echizen, and J. Lorenzo-Trueba, "High-quality nonparallel voice conversion based on cycle-consistent adversarial network," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5279–5283.

[16] H. Kameoka, T. Kaneko, K. Tanaka, and N. Hojo, "Acvae-vc: Non-parallel many-to-many voice conversion with auxiliary classifier variational autoencoder," *IEEE TASLP*, 2019.

[17] Y. Saito, Y. Ijima, K. Nishida, and S. Takamichi, "Non-parallel voice conversion using variational autoencoders conditioned by phonetic posteriorgrams and d-vectors," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 5274–5278.

[18] R. Takashima, R. Aihara, T. Takiguchi, and Y. Ariki, "Noise-robust voice conversion based on spectral mapping on sparse space," in *Eighth ISCA Workshop on Speech Synthesis*, 2013.

[19] X. Miao, M. Sun, X. Zhang, and Y. Wang, "Noise-robust voice conversion using high-quefrency boosting via sub-band cepstrum conversion and fusion," *Applied Sciences*, vol. 10, no. 1, 2020.

[20] L. Sun, S. Kang, K. Li, and H. Meng, "Voice conversion using deep bidirectional long short-term memory based recurrent neural networks," in *International conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2015, pp. 4869–4873.

[21] L. Wan, Q. Wang, A. Papir, and I. L. Moreno, "Generalized end-to-end loss for speaker verification," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018.

[22] J. Lorenzo-Trueba, T. Drugman, J. Latorre, T. Merritt, B. Putrycz, R. Barra-Chicote, A. Moinet, and V. Aggarwal, "Towards achieving robust universal neural vocoding," *Interspeech*, 2019.

[23] M. Jeub, M. Schafer, and P. Vary, "A binaural room impulse response database for the evaluation of dereverberation algorithms," in *16th International Conference on Digital Signal Processing*. IEEE, 2009.

[24] R. Scheibler, E. Bezzam, and I. Dokmanić, "Pyroomacoustics: A python package for audio room simulation and array processing algorithms," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 351–355.

[25] J. Thiemann, N. Ito, and E. Vincent, "The diverse environments multi-channel acoustic noise database (demand): A database of multichannel environmental noise recordings," in *Proc. of Meetings on Acoustics*, vol. 19, no. 1, 2013.

[26] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "librosa: Audio and music signal analysis in python," in *Proc. of the 14th python in science conference*, 2015.

[27] M. McAuliffe, M. Socolof, S. Mihuc, M. Wagner, and M. Sonderegger, "Montreal forced aligner: Trainable text-speech alignment using kaldi." in *Interspeech*, 2017, pp. 498–502.

[28] Y. Ephraim and D. Malah, "Speech enhancement using a minimum mean-square error log-spectral amplitude estimator," *IEEE transactions on acoustics, speech, and signal processing*, vol. 33, no. 2, pp. 443–445, 1985.

[29] T. Nakatani, T. Yoshioka, K. Kinoshita, M. Miyoshi, and B.-H. Juang, "Speech dereverberation based on variance-normalized delayed linear prediction," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 7, pp. 1717–1731, 2010.

[30] I. Recommendation, "Method for the subjective assessment of intermediate sound quality (mushra)," *ITU, BS*, pp. 1543–1, 2001.

11th ISCA Speech Synthesis Workshop (SSW 11)
26-28 August 2021, Budapest, Hungary

Rapping-Singing Voice Synthesis based on Phoneme-level Prosody Control

Konstantinos Markopoulos[⋆], *Nikolaos Ellinas*[⋆], *Alexandra Vioni*[⋆], *Myrsini Christidou*[⋆], *Panos Kakoulidis*[⋆], *Georgios Vamvoukakis*[⋆], *Georgia Maniati*[⋆], *June Sig Sung*[†], *Hyoungmin Park*[†], *Pirros Tsiakoulis*[⋆], *Aimilios Chalamandaris*[⋆]

[⋆] Innoetics, Samsung Electronics, Greece
[†] Mobile Communications Business, Samsung Electronics, Republic of Korea

{k.markop, n.ellinas}@samsung.com,
{a.vioni, m.christidou}@partner.samsung.com,
{p.kakoulidis, g.vamvouk, g.maniati, js6.sung, hm94.park, p.tsiakoulis, aimilios.ch}@samsung.com

Abstract

In this paper, a text-to-rapping/singing system is introduced, which can be adapted to any speaker's voice. It utilizes a Tacotron-based multi-speaker acoustic model trained on read-only speech data and which provides prosody control at the phoneme level. Dataset augmentation and additional prosody manipulation based on traditional DSP algorithms are also investigated. The neural TTS model is fine-tuned to an unseen speaker's limited recordings, allowing rapping/singing synthesis with the target's speaker voice. The detailed pipeline of the system is described, which includes the extraction of the target pitch and duration values from an a capella song and their conversion into target speaker's valid range of notes before synthesis. An additional stage of prosodic manipulation of the output via WSOLA is also investigated for better matching the target duration values. The synthesized utterances can be mixed with an instrumental accompaniment track to produce a complete song. The proposed system is evaluated via subjective listening tests as well as in comparison to an available alternate system which also aims to produce synthetic singing voice from read-only training data. Results show that the proposed approach can produce high quality rapping/singing voice with increased naturalness.

Index Terms: text-to-speech, rapping voice synthesis, singing voice synthesis, text-to-rapping, text-to-singing, prosody control, prosody manipulation, neural models

1. Introduction

With the recent development of neural text-to-speech (TTS), the task of singing voice synthesis (SVS) is gaining popularity, since it has become feasible to produce natural and expressive speech more effectively. Before the development of deep neural network based synthesis, SVS systems were mainly based on unit selection technology [1, 2, 3, 4] or parametric TTS [5, 6]. During the last few years, with the establishment of neural TTS systems such as Tacotron [7], it has become possible to investigate approaches like neural rapping and singing.

SVS is a complicated task. As in regular TTS, a large and powerful neural model that accurately predicts acoustic features must be designed and tuned. Additionally, singing information such as musical notes and rhythm must be accurately followed, in order to produce high quality samples. For rapping, the procedure is the same, though the focus is less on musical notes, and more on pitch variation and rhythm, the latter translating into accurate phoneme durations.

1.1. Related work

Several approaches of rapping and singing voice synthesis have been proposed over the years. Early methods were based on unit concatenation [1, 2, 3, 4] and statistical parametric synthesis [5, 6]. There was also an attempt focused on speech-to-rap voice conversion, based on a phase vocoder and beat tracking [8]. Nevertheless, such approaches had significant limitations and did not achieve high quality synthesized speech. In recent years, many steps have been made towards high quality SVS, with the use of neural and deep learning methods. Hybrid, mixed and conditioned models have been introduced that advance further the SVS systems. Some notable approaches are the WaveNet variant architecture [9] used for parametric singing synthesis, WGANSing which is a pitch conditioned Generative Adversarial Network (GAN) [10] and an adversarially trained, pitch conditioned sequence-to-sequence Korean singing model [11]. Another GAN-based approach is unsupervised cross-domain singing voice conversion [12], which uses additional perceptual losses on its generator output. Mellotron [13], a multi-speaker expressive voice synthesis model based on GST-Tacotron 2 [14], also has SVS capabilities. Moreover, Jukebox [15] generates singing voice with accompaniments, and UTACO [16] consists of an attention-based sequence-to-sequence mechanism and a vocoder with dilated causal convolutions. Another approach was Durian-SC, a duration-informed, phoneme-to-acoustic features' alignment model with composite conditioning that allows SVS [17]. The transformer architecture has also been employed to address the problem, as in DeepSinger [18], which employs separate encoders for its features, and HiFiSinger [19], which includes a FastSpeech-based [20] model and multi-scale adversarial training approaches. Recently, ByteSing [21] was introduced, a Tacotron 2 model combined with a duration prediction model and uses linguistic along with musical embeddings.

1.2. Proposed method

In this paper, we propose a complete text-to-rapping and singing approach based on a system that, unlike other methods, relies solely on spoken data and can be adapted to an unseen target voice with very limited data. Our method is based on a fine-grained prosody manipulation multi-speaker TTS model presented in [22]. Combined with augmentation of training data, our method can achieve phoneme-level prosody manipulation (F0 and duration), which allows us to generate rapping and singing synthesized speech.

10.21437/SSW.2021-21

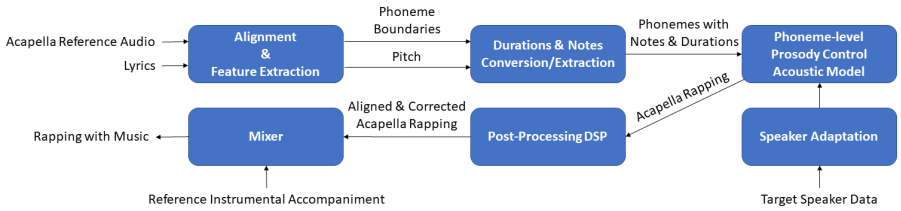


Figure 1: *Proposed text-to-rapping/singing system*

We train a multi-speaker multilingual TTS model on internal spoken data, in US English and Korean languages, and show that the prosody control capabilities at the phoneme-level musical note and duration are effective in all training speakers. For the case of speaker adaptation with very limited data, we resume the model training on 2 speakers of both languages with only about 11 minutes of spoken data, and show that the resulting models can also maintain the same prosody control capabilities. Extracting the desired prosody information from a given a capella song and presenting it as input to our model enables us to synthesize a capella utterances that closely follow the reference, allowing rapping/singing synthesis with the voice of every speaker included in the training set.

We present the proposed approach, from the stage of phoneme alignment and F0 extraction and discretization, up to the final post-processing steps required to produce the actual song with the voice of the selected speaker. This includes several processing steps and modules as well as a fine-tuning digital signal processing (DSP) stage based on synchronous overlap-add algorithms [23, 24], in order to achieve exact alignment of the synthetic speech to the target song music track. Finally, our system is evaluated via crowd-sourced listening tests against the ground truth samples, as well as against samples from Mellotron, a widely accepted state-of-the-art SVS system. Objective evaluation results are also presented, showing that our model follows the desired prosody patterns appropriately.

2. Method

2.1. Overview

Our proposed text-to-rapping/singing system aims at producing a song with the voice of a target speaker, based only on the lyrics and an a capella version of the song by its original singer. A block diagram of the system is shown in Figure 1 and consists of four main parts.

The first part involves the preprocessing of the lyrics by a front-end module in order to obtain the phonemes, as well as the extraction of the required prosodic features from the reference audio, i.e. phoneme durations and F0 (Section 2.2). Second, to avoid extreme F0 values which can hinder the output quality, F0 contours are converted so as to lie within the range of the target speaker (Section 2.3). Third, the converted F0 values along with the phoneme durations are discretized and used by the TTS model, which produces the synthesized song by controlling the prosody at the phoneme-level (Section 2.4). The fourth step is post-processing, including the introduction of a DSP module, which is necessary for accurate time alignment of the produced song with the original time specification, if the instrumental accompaniment track needs to be combined. This task can be performed by a mixer which combines the synthesized a capella utterances with the music track (Section 2.5). A detailed analysis of each step is given in the following sections.

2.2. Alignment & feature extraction

A capella songs are used as reference since the acoustic model is trained only on neutral style spoken data which do not contain music tracks. Initially, phoneme alignments are automatically extracted from the song using an HMM monophone acoustic model trained using flat start initialization [25]. As the forced-alignment model is trained only on spoken utterances and may not produce precise phoneme boundaries, a manual correction of the alignments was performed. Most corrections were attributed to accommodate for long vowel durations in the singing data, which are not usually encountered in spoken speech.

F0 contours are calculated using the algorithm included in the Praat toolkit [26]. Interpolation is applied on the unvoiced regions in order to avoid zero values and the final contour is smoothed.

2.3. Note conversion/extraction

The acoustic model is trained to control the prosody within the range of each speaker. When a reference song is used as input to the proposed system, the F0 contour of the singing speaker may have different range than that of the target speaker. This is due to the fact that the song originates from a different speaker with diverse source characteristics or gender. Also, the singing speech itself may vary in extreme F0 values both in the lower and higher end between two speakers.

We use Eq (1) in order to transpose the F0 contour of the reference speaker to match the range of the target speaker:

$$f_{target} = \frac{median(f_{speaker})}{median(f_{ref})} \cdot f_{ref} \quad (1)$$

where $f_{speaker}$ and f_{ref} represents all the F0 values of the target and reference speaker's utterances respectively.

Two approaches were examined on F0 transposition. The first approach takes one single (global) pitch value for the reference speaker, which is the median F0 value, while the second recalculates the median F0 value for every verse. Early results demonstrated that the former approach works better, producing more stable samples, and avoids the discontinuities in octaves that may occur due to the recalculation of the F0.

After the conversion, the average phoneme-level F0 is calculated by using the previously extracted alignments. The corresponding musical note and octave can be extracted by using formulas (2) and (3):

$$h = \left\lfloor 12 \cdot \log_2 \frac{f_{target}}{440} \right\rceil + 57 \quad (2)$$

$$octave = \left\lfloor \frac{h}{12} \right\rfloor \quad \text{and} \quad note = (h \bmod 12) \tag{3}$$

where h represents the distance in semitones from the note C_0.

2.4. Acoustic model

For controlling prosody, we use an acoustic model based on previous work [27]. This method uses unsupervised clustering on phoneme-level F0 and duration values in order to extract a sequence of discrete learnable prosodic labels for each utterance, which is then used to condition the decoder of a Tacotron-based acoustic model [7, 14] in parallel to the phoneme sequence. The model is also augmented with a secondary Mixture-of-Logistics (MoL) attention module [28] which operates on the prosodic sequence only, aiming to disentangle the phonetic and prosodic content. The final model is able to control the prosody at the phoneme level both for F0 and duration while maintaining high speech quality.

In [27], it is shown that this model is also effective at representing musical notes instead of F0 values that are derived from K-means unsupervised clustering. In the current work, we follow the procedure described in Section 2.3 to assign the notes and octaves to separate learnable embeddings, so that any possible combinations are modeled appropriately, even non-existing ones in the training set. This global representation of musical notes is also speaker-independent, being suitable for our multi-speaker setup described below.

For the duration labels, we follow an improved and more stable method than K-means, as in [22]. The values are sorted in ascending order and grouped into a desired number of intervals, so that an equal number of samples is contained in each interval. This alleviates the problem of voice quality deterioration in extreme values which are not common in the training set, while slightly decreasing the duration control range. In this work, in order to investigate the effect of the number of labels involved, we have experimented with two different setups, one with 15 and another with 30 duration labels.

The training setup follows a multi-speaker/multi-lingual scenario, similar to the parallel work done in [22]. A multitude of speakers of both genders and 2 languages is used instead of a single one, in order to capture as many prosody patterns as possible, which will help increase the range of the model in both F0 and duration. That way each speaker has the capability to rap/sing a wider variety of songs. Augmentation is also employed both in F0 and duration by applying pitch shifting and tempo alteration respectively, further increasing the quantity and range of the training data. This setup also allows us to perform speaker adaptation using limited data from unseen speakers and enable rapping/singing for the adaptation speaker in both languages. This process becomes easier as the model does not have unseen values due to the global musical note representation for the F0 and the duration intervals which are derived from all speakers and are common throughout the training procedure.

The different speakers are assigned a learnable speaker embedding, which also conditions the decoder at each step. We also use a linear adversarial speaker classifier on the phoneme encoder outputs, in order to make them speaker independent, as well as a residual variational encoder which captures other latent factors of the recordings [29]. This method is shown to improve naturalness and stability by simply using a zero vector during inference, which is essentially the prior mean. The multilingual setup does not include language embeddings, but simply considering every phone for each language with a different label, increasing the total number of phones to the sum of each language phoneset. For the speaker adaptation, we found that freezing the weights of the phoneme encoder, prosody encoder and attention modules yields better results. We can account this to the fact that the model has already learned rich representations which must not be forgotten in the adaptation stage by the target speaker's limited data.

Table 1: *Multilingual multi-speaker dataset for text-to-singing/rapping model training. 'tr' and 'ad' refer to training and adaptation speakers respectively, while 'f' and 'm' refer to gender.*

Speaker	Language	Rec. Hours	Utterances
us_tr_f1	en-us	41.21	36185
us_tr_f2	en-us	38.88	45841
us_tr_m1	en-us	36.82	40442
ko_tr_f1	ko	51.37	40503
ko_tr_f2	ko	54.29	29289
us_ad_m1	en-us	0.19	165
ko_ad_m1	ko	0.18	149

2.5. Post-processing

The multi-speaker/multi-lingual prosody control model produces a capella utterances which are derived between silence tokens from the original lyrics. These utterances must be concatenated in the time domain in order to obtain the full verse of the song. Additionally, the duration values are discrete in our model, resulting in some form of quantization, hence the final durations may not exactly match the original song. The accurate matching of the synthesized song with the original, requires lengthening or shortening of speech, which in our case is performed utilizing time-domain DSP methods.

This stage is based on two main algorithms: WSOLA [24] and PSOLA [30]. These algorithms are both able to modify the duration of each phoneme without affecting the pitch and resynthesize the original audio using the overlap-add technique. In early listening experiments, we found that WSOLA produced slightly better results in almost every comparison, so we included this method in the evaluations that follow. A Mixer element can also be included for producing the final song, mixing the time-aligned processed synthetic a capella utterance with the respective music track. Mixing more than one voices is also possible, providing a multi-speaker song.

3. Experiments & results

In this section, we describe our experimental setup, along with the method followed to objectively and subjectively evaluate the proposed system. The evaluation results are then discussed.

3.1. Experimental setup

The acoustic model is trained with an internal multilingual multi-speaker dataset containing 222 hours of speech in both US English (en-us) and Korean (ko) from 1 male and 4 female speakers. The recorded dataset is a general TTS corpus of neutral speaking style. For speaker adaptation, we use about 11 minutes of recordings from an unseen male speaker from each language. The adaptation utterances are selected with a corpus selection process described in [31] which ensures maximum phonetic coverage for the required amount of recordings. Details regarding the data used for training and adaptation are

Table 2: *Average semitone pitch & duration (ms) score for objective evaluation*

		Training Speakers	Adaptation Speakers
Pitch Error (semitones)	15-class	0.85 ± 0.22	0.86 ± 0.23
	30-class	0.93 ± 0.28	0.93 ± 0.21
	Mellotron	0.51 ± 0.15	
Duration Error (ms)	15-class	27 ± 9	29 ± 8
	30-class	29 ± 10	28 ± 7
	Mellotron	25 ± 9	

presented in Table 1. The augmentations applied to the original training data, similar to [22], include increasing and decreasing the F0 by 2, 4 or 6 semitones and the speaking rate from 70% - 130%. The final training set size after the augmentations is 415 hours.

All audio data was resampled to 24 kHz. The acoustic features were extracted in order to match the modified LPCNet Vocoder [32] and consist of 20 Bark-scale cepstral coefficients, the pitch period and pitch correlation. The phoneme encoder maps the input phoneme sequence into 256 dimensional embeddings and further applies a CBHG module. In the prosody encoder, prosodic labels are mapped into 64 dimensional embeddings. These are processed by a single 128-dimensional feed-forward Pre-Net with ReLU activation and a bidirectional Gated Recurrent Unit (GRU) layer with 128 dimensions in each direction. The decoder contains 3 recurrent layers, a 256-dimensional attention GRU and two 512-dimensional residual LSTMs. The attention modules used have a mixture of 5 logistic distributions and 256-dimensional feed-forward layers. Dropout regularization [33] of rate 0.5 is applied on all Pre-Net and Post-Net layers and Zoneout [34] of rate 0.1 is applied on LSTM layers. We use the Adam optimizer [35] for training the network parameters with batch size 32. The learning rate is initially 10^{-3} and decays linearly to $3 \cdot 10^{-5}$ after $100,000$ iterations. We also apply L2 regularization with factor 10^{-6}. Speaker adaptation involves further training of the prosody model for 5K iterations, with frozen weights in the phoneme encoder, prosody encoder and attention modules.

3.2. Objective evaluation

Objective evaluation of the proposed systems in terms of a capella song synthesis was attempted. Clips from 4 rap songs and 4 songs, all sung a capella in English, were used as ground truth for this evaluation[1]. All the proposed models along with Mellotron [13] were evaluated against the ground truth a capella songs. The samples for Mellotron were produced by Mellotron's latest GitHub repository along with a pretrained Mellotron model and respective WaveGlow model based on LibriTTS dataset. A random female voice from the model was selected for inference of the audio stimuli.

In Figure 2, the pitch contours of 2 a capella reference songs are presented, along with the pitch contours of the audio stimuli inferred by the systems under comparison. One can notice that the synthesized pitch contours closely follow the reference ones, while they are shifted appropriately in order to match the F0 mean value of the inference speaker of each model. In Figure 3, where the F0 contours of a reference and synthesized clip are illustrated in a MIDI-like graph, a MIDI value is produced

[1]The reader is encouraged to listen to the audio samples at: https://innoetics.github.io/publications/rappotron/index.html

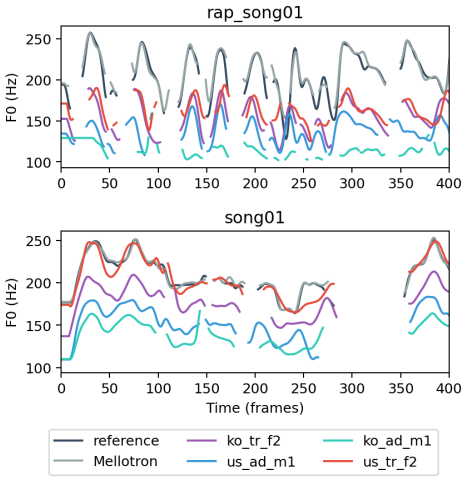


Figure 2: *Pitch contours of the original song, Mellotron and proposed models.*

for each phoneme, and the trend of the synthesized melody following in parallel the ground truth one is obvious.

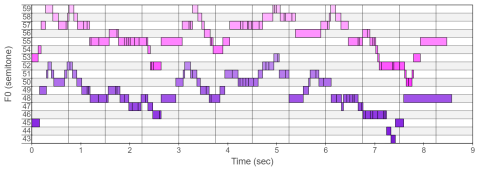


Figure 3: *Pitch contours of the original and synthesized song in a MIDI representation (pink and purple values respectively). The MIDI values are calculated on a per phoneme basis.*

As an objective metric for measuring the accuracy of our approach, we calculated the average distance of F0 and duration between the reference and the synthesized audio on a per phoneme basis. F0 values of the reference audio were transposed again, so as to target the speaker's mean F0 value before calculation of the distance. As depicted in Table 2, there is no significant difference in these metrics between speakers seen in the training set and those used only for adaptation, neither on F0 nor on duration values. This observation is valid for both 15 and 30 duration label models. Mellotron seems to achieve an F0 contour closer to the reference, as was also illustrated in Figure 2.

Overall, our method manages to reproduce a capella singing with accurate pitch and duration phoneme values especially in cases where the respective a capella reference audio is straightforward, without notably long durations or extreme low/high notes. In the latter cases, we noticed that our approach did not accurately produce phonemes with the target F0 or duration values, leading either to attention failures or duration mismatches with the reference audio.

Table 3: *Mean Opinion Score (MOS) evaluation results with 95% confidence interval*

	30 duration labels				**15 duration labels**			
	rap_songs		**songs**		**rap_songs**		**songs**	
setup	**plain**	**w/ post-proc**	**plain**	**w/ post-proc**	**plain**	**w/ post-proc**	**plain**	**w/ post-proc**
us_tr_f1	3.60±0.36	3.65 ±0.30	3.36±0.33	3.71 ±0.33	3.65±0.32	3.72 ±0.35	3.69±0.33	3.69 ±0.29
us_tr_f2	3.60±0.30	3.86 ±0.29	3.38±0.38	3.50 ±0.38	3.53±0.31	3.74 ±0.35	3.81±0.29	3.55 ±0.37
us_tr_m1	3.60±0.33	4.00 ±0.29	3.64±0.32	3.60 ±0.36	3.51±0.27	3.56 ±0.32	3.64±0.34	3.62 ±0.29
ko_tr_f1	3.35±0.33	3.70 ±0.36	3.17±0.34	3.33 ±0.41	3.44±0.37	3.35 ±0.34	3.81±0.32	3.57 ±0.31
ko_tr_f2	3.37±0.34	3.42 ±0.35	3.45±0.39	3.33 ±0.37	3.40±0.32	3.40 ±0.34	3.69±0.35	3.81 ±0.32
us_ad_m1	3.67±0.35	3.74 ±0.29	3.74±0.35	3.33 ±0.37	3.79±0.31	3.51 ±0.35	3.57±0.34	3.33 ±0.39
ko_ad_m1	3.42±0.39	3.47 ±0.39	3.26±0.42	3.26 ±0.40	3.63±0.36	3.51 ±0.32	3.26±0.38	3.43 ±0.38
Total	**3.52 ±0.34**	**3.69 ±0.33**	**3.43 ±0.36**	**3.44 ±0.37**	**3.56 ±0.32**	**3.54 ±0.33**	**3.64 ±0.33**	**3.57 ±0.33**
Mellotron	3.33±0.38		3.17±0.49					

3.3. Subjective evaluation and discussion

A subjective evaluation was carried out via a formal listening test. Two clips from songs and two clips from rap songs, all in English a capella, were used as ground truth, and the respective audio stimuli produced by our system and Mellotron were rated. In the framework of this subjective evaluation, aside from the overall quality, we also opted to assess: a) the effect of the number of duration labels per phoneme, and b) the effect of the post-processing stage for duration matching to the reference audio via WSOLA (as described in Section 2.5). By combining the aforementioned parameters we produced all 4 possible models for each speaker described in Table 1. In total, 60 listeners (Amazon Mechanical Turkers) participated in the listening test, rating each synthetic stimulus on a 5-point Likert scale on both melody and intelligibility, with 1 indicating "Totally off-tune or wrong lyrics" and 5 indicating "Exactly same melody and lyrics".

The average MOS and 95% confidence interval for each voice model versus song type (rap song and song) and post-processing are presented in Table 3. The results show that our approach provides satisfactory output, equally for both rapping and singing, even if the latter is considered as a more complex task. Overall, post-processing for matching the word boundaries between ground-truth and audio stimuli does not seem to provide any consistent and robust improvement. Although there is no statistical significance in the pairwise differences observed between models, an improvement tendency is observed when post-processing is used for rap songs and the 30-duration-labeled model. This post-processing stage remains necessary in order to align the generated songs with the music track at the final mixing stage, in case of music accompaniment of the synthesized a cappella voice. The 15-class duration labeling yields an improvement tendency for singing, which is most probably attributed to the fact that fewer but more populated classes in training entail better learning for our model.

A closer inspection of the results leads us to the conclusion that adapted speakers achieve similar quality results with the speakers who are seen during training. This is prominent especially for English, where the adapted voice performs equally well to the rest of the English training set voices, while, at the same time, the Korean adapted voice follows closely the performance of the Korean training voices. Such similar MOS ratings confirm our hypothesis that our approach is robust for limited speaker data scenarios. As far as the per speaker and language performance is concerned, Korean voice models have received lower MOS scores than the English ones. This is most probably due to the fact that the reference songs we evaluated are exclusively in English. Our informal evaluation showed that these non-native voice models mainly suffered from lower intelligibility or lack of naturalness, as they did not bear native English accent and thus sounded more artificial when synthesizing English a capella songs.

Our informal listening evaluation of the samples showed that Mellotron output is melodic but often bears intelligibility issues or audio artifacts. This observation may justify why Mellotron scored lower compared to our system in the formal subjective evaluation (Table 3). Nevertheless, Mellotron outperformed our approach in a test song where vibrato singing was prominent, a voice characteristic that Mellotron can capture and transfer well, in contrast to our approach where only F0 and duration values per phoneme are provided.

4. Conclusions

In this paper, we presented an approach for producing high-quality singing and rapping synthesis from a Tacotron-based fine-grained prosody-control voice model trained solely on read data. Even though its results do not match the output of an SVS system trained on singing data, our approach achieves satisfactory results in both singing and rapping. Experiments showed that equally good singing synthesis can be achieved for limited-data target voices via adaptation. It is worth-noting that our system, similarly to other systems based on spoken-only data, suffers limitations in its ability to produce too long and extremely low or high-pitched sounds. In other words, although it can provide satisfactory results for rapping and simple songs, it may fail to produce adequate singing for more challenging songs with wide variations in both note values and duration. Another native limitation to our current approach is the lack of ability to transfer micro-prosodic characteristics into the synthetic output, such as vibrato or tremolo. Mellotron was shown to better imitate micro-prosodic singing voice qualities with the help of a more complex model and attention mechanism. Further research on controlling singing voice characteristics, such as loudness and vibrato, as well as on including singing data in the training process is required. Moreover, we plan to investigate ways for automating parts of the proposed process, such as the alignment optimization of the target singing data, so as to eliminate manual effort in our method.

5. References

[1] M. Macon, L. Jensen-Link, E. B. George, J. Oliverio, and M. Clements, "Concatenation-based midi-to-singing voice synthesis," in *Audio Engineering Society Convention 103*. Audio Engineering Society, 1997.

[2] A. J. Hunt and A. W. Black, "Unit selection in a concatenative speech synthesis system using a large speech database," in *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, vol. 1. IEEE, 1996, pp. 373–376.

[3] H.-Y. Gu and J.-K. He, "Singing-voice synthesis using demi-syllable unit selection," in *2016 International Conference on Machine Learning and Cybernetics (ICMLC)*, vol. 2. IEEE, 2016, pp. 654–659.

[4] J. Bonada, M. Umbert Morist, and M. Blaauw, "Expressive singing synthesis based on unit selection for the singing synthesis challenge 2016," *Morgan N, editor. Interspeech 2016; 2016 Sep 8-12; San Francisco, CA. ISCA; 2016. p. 1230-4.*, 2016.

[5] K. Nakamura, K. Oura, Y. Nankaku, and K. Tokuda, "Hmm-based singing voice synthesis and its application to japanese and english," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 265–269.

[6] K. Saino, H. Zen, Y. Nankaku, A. Lee, and K. Tokuda, "An hmm-based singing voice synthesis system," in *Ninth International Conference on Spoken Language Processing*, 2006.

[7] W. et al, "Tacotron: Towards end-to-end speech synthesis," in *Proc. Interspeech*, 2017.

[8] M. Wu, C. Lu, and J. R. Jang, "Automatic conversion from speech to rap music," in *2014 International Conference on Electrical Engineering and Computer Science (ICEECS)*, 2014, pp. 245–250.

[9] M. Blaauw and J. Bonada, "A neural parametric singing synthesizer modeling timbre and expression from natural songs," *Applied Sciences*, vol. 7, no. 12, p. 1313, 2017.

[10] P. Chandna, M. Blaauw, J. Bonada, and E. Gomez, "Wgansing: A multi-voice singing voice synthesizer based on the wasserstein-gan." IEEE, 2019, pp. 1–5.

[11] J. Lee, H.-S. Choi, C.-B. Jeon, J. Koo, and K. Lee, "Adversarially trained end-to-end korean singing voice synthesis system," 2019.

[12] A. Polyak, L. Wolf, Y. Adi, and Y. Taigman, "Unsupervised Cross-Domain Singing Voice Conversion," in *Proc. Interspeech 2020*, 2020, pp. 801–805.

[13] R. Valle, J. Li, R. Prenger, and B. Catanzaro, "Mellotron: Multispeaker expressive voice synthesis by conditioning on rhythm, pitch and global style tokens," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6189–6193.

[14] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan *et al.*, "Natural TTS synthesis by conditioning wavenet on MEL spectrogram predictions," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4779–4783.

[15] P. Dhariwal, H. Jun, C. Payne, J. W. Kim, A. Radford, and I. Sutskever, "Jukebox: A generative model for music," *arXiv preprint arXiv:2005.00341*, 2020.

[16] O. Angelini, A. Moinet, K. Yanagisawa, and T. Drugman, "Singing synthesis: with a little help from my attention," 2020.

[17] L. Zhang, C. Yu, H. Lu, C. Weng, C. Zhang, Y. Wu, X. Xie, Z. Li, and D. Yu, "DurIAN-SC: Duration Informed Attention Network Based Singing Voice Conversion System," in *Proc. Interspeech 2020*, 2020, pp. 1231–1235.

[18] Y. Ren, X. Tan, T. Qin, J. Luan, Z. Zhao, and T.-Y. Liu, "Deepsinger: Singing voice synthesis with data mined from the web," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 1979–1989.

[19] J. Chen, X. Tan, J. Luan, T. Qin, and T.-Y. Liu, "Hifisinger: Towards high-fidelity neural singing voice synthesis," 2020.

[20] Y. Ren, Y. Ruan, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, "Fastspeech: Fast, robust and controllable text to speech," *arXiv preprint arXiv:1905.09263*, 2019.

[21] Y. Gu, X. Yin, Y. Rao, Y. Wan, B. Tang, Y. Zhang, J. Chen, Y. Wang, and Z. Ma, "Bytesing: A chinese singing voice synthesis system using duration allocated encoder-decoder acoustic models and wavernn vocoders," 2021.

[22] M. Christidou, A. Vioni, N. Ellinas, G. Vamvoukakis, K. Markopoulos, P. Kakoulidis, J. S. Sung, H. Park, A. Chalamandaris, and P. Tsiakoulis, "Improved prosodic clustering for multi-speaker and speaker-independentphoneme-level prosody control," in *SPECOM*, 2021 [SUBMITTED].

[23] F. Charpentier and M. Stella, "Diphone synthesis using an overlap-add technique for speech waveforms concatenation," in *ICASSP'86. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 11. IEEE, 1986, pp. 2015–2018.

[24] W. Verhelst and M. Roelands, "An overlap-add technique based on waveform similarity (wsola) for high quality time-scale modification of speech," in *1993 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2. IEEE, 1993, pp. 554–557.

[25] S. Raptis, P. Tsiakoulis, A. Chalamandaris, and S. Karabetsos, "Expressive speech synthesis for storytelling: the innoetics' entry to the blizzard challenge 2016," in *Proc. Blizzard Challenge*, 2016.

[26] P. Boersma, "Praat: doing phonetics by computer," *http://www.praat. org*, 2006.

[27] A. Vioni, M. Christidou, N. Ellinas, G. Vamvoukakis, P. Kakoulidis, T. Kim, J. S. Sung, H. Park, A. Chalamandaris, and P. Tsiakoulis, "Prosodic clustering for phoneme-level prosody control in end-to-end speech synthesis," in *Proc. ICASSP*, 2021.

[28] N. Ellinas, G. Vamvoukakis, K. Markopoulos, A. Chalamandaris, G. Maniati, P. Kakoulidis, S. Raptis, J. S. Sung, H. Park, and P. Tsiakoulis, "High quality streaming speech synthesis with low, sentence-length-independent latency," in *Proc. Interspeech*, 2020.

[29] Y. Zhang, R. J. Weiss, H. Zen, Y. Wu, Z. Chen, R. Skerry-Ryan, Y. Jia, A. Rosenberg, and B. Ramabhadran, "Learning to Speak Fluently in a Foreign Language: Multilingual Speech Synthesis and Cross-Language Voice Cloning," in *Proc. Interspeech 2019*, 2019, pp. 2080–2084. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2019-2668

[30] Y. Laprie and V. Colotte, "Automatic pitch marking for speech transformations via td-psola," in *9th European Signal Processing Conference (EUSIPCO 1998)*. IEEE, 1998, pp. 1–4.

[31] A. Chalamandaris, P. Tsiakoulis, S. Raptis, and S. Karabetsos, "Corpus design for a unit selection TTS system with application to Bulgarian," in *Proc. 4th Conference on Human language technology: challenges for computer science and linguistics*, 2009, pp. 35–46.

[32] R. Vipperla, S. Park, K. Choo, S. Ishtiaq, K. Min, S. Bhattacharya, A. Mehrotra, A. G. C. Ramos, and N. D. Lane, "Bunched lpcnet: Vocoder for low-cost neural text-to-speech systems," *arXiv preprint arXiv:2008.04574*, 2020.

[33] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[34] D. Krueger, T. Maharaj, J. Kramár, M. Pezeshki, N. Ballas, N. R. Ke, A. Goyal, Y. Bengio, A. Courville, and C. Pal, "Zoneout: Regularizing rnns by randomly preserving hidden activations," *arXiv preprint arXiv:1606.01305*, 2016.

[35] D. P. Kingma and J. Ba, "Adam: A methodfor stochastic optimization," in *International Conference onLearning Representations (ICLR)*, 2015.

11th ISCA Speech Synthesis Workshop (SSW 11)
26-28 August 2021, Budapest, Hungary

Exploring Disentanglement with Multilingual and Monolingual VQ-VAE

Jennifer Williams[1], *Jason Fong*[1], *Erica Cooper*[2], *Junichi Yamagishi*[2]

[1]Centre for Speech Technology Research, University of Edinburgh, UK
[2]National Institute for Informatics, Japan

{j.williams,jason.fong}@ed.ac.uk, {ecooper,jyamagis}@nii.ac.jp

Abstract

This work examines the content and usefulness of disentangled phone and speaker representations from two separately trained VQ-VAE systems: one trained on multilingual data and another trained on monolingual data. We explore the multi- and monolingual models using four small proof-of-concept tasks: copy-synthesis, voice transformation, linguistic code-switching, and content-based privacy masking. From these tasks, we reflect on how disentangled phone and speaker representations can be used to manipulate speech in a meaningful way. Our experiments demonstrate that the VQ representations are suitable for these tasks, including creating new voices by mixing speaker representations together. We also present our novel technique to conceal the content of targeted words within an utterance by manipulating phone VQ codes, while retaining speaker identity and intelligibility of surrounding words. Finally, we discuss recommendations for further increasing the viability of disentangled representations.

Index Terms: code-switching, voice conversion, content-based privacy

1. Introduction

One of the main benefits of using Vector Quantization Variational Autoencoders (VQ-VAE) for speech synthesis is that this architecture facilitates learning rich representations of speech [1, 2, 3, 4] in the form of discrete latent sequences. These learned representations come from vector-quantized *codebooks* that behave as a clustering space with prototype centroids. Each entry in a codebook is represented by a pair consisting of a *code* (also known as an index or token) and its corresponding vector. The code is a discrete integer value, and the vector is a learned n-dimensional array of continuous values. In this paper, we are interested in the content and usefulness of codebooks after a VQ-VAE model has been trained. Specifically, we are the first to compare multilingual and monolingual VQ-VAE codebook representations for phone and speaker, with the aim to observe how well they adapt to voice transformation, linguistic code-switching and content-masking.

The original VQ-VAE architecture design was based on a single VQ space: one encoder, one VQ codebook, and one decoder. That design proved to be useful across different objectives in image, video, and speech processing [3]. Since then, others have shown that the architecture could be expanded by stacking encoders which result in learning multiple different VQ spaces at the same time [2, 5] or even hierarchical representations [6]. These extended models provide more generalization capability, in part because they learn richer representations.

It is possible to model multiple types of information in the speech signal with little or no supervision. In the process of learning to represent different types of information, the stacked VQ-VAE architectures are also providing a means to separate informational factors. This act of separating information from representations is known by several names, including factorization and disentanglement. Traditionally, factorization has served the purpose of removing irrelevant information from a representation such as a speaker embedding – and then discarding what had been deemed irrelevant [7]. After information has been removed, it could be argued that a representation is in some way more "pure". On the other hand, disentanglement retains information. At the time of this writing we use the term *disentanglement* to describe the phenomenon of isolating multiple types of distributed information from one source, into separate external representations. Functionally, this is a form of distributed representation learning.

Currently there are no single-best techniques to measure the intrinsic goodness of disentangled representations apart from probing how well they perform in extrinsic tasks [8, 9, 10, 11]. Recent efforts for phone and speaker disentanglement have been limited to contrastive tasks such as phone recognition and speaker recognition [2, 12]. Or observing that one representation "gains" information while another "loses" information [10, 13] by measuring changes in classification accuracy.

Our work adds additional task-based evaluation by exploring disentanglement in both a multilingual and monolingual model. In order for the multilingual model to perform well at tasks such as voice transformation and linguistic code-switching, the learned representations must completely separate phonetic content and speaker information. We also introduce a novel technique that uses VQ phone codes to manipulate targeted content in the speech signal without altering the sound of a speaker's voice. Our exploration exposes some of the interesting capabilities of disentangled representations. We also offer ideas for improving the VQ-VAE architecture.

2. Related Work

Early versions of the VQ-VAE architecture with a single encoder and VQ phone codebook are known to be well-suited to voice conversion. Particularly [14] showed that grouping latent embeddings together during the training process helps with mispronunciations. Their system relied on one-hot speaker encodings, but they suggest that the model could be made to generalize to unseen speakers by using externally-learned speaker embeddings instead. Our VQ-VAE implementation uses a similar approach to group latent embeddings, but goes one step further to simultaneously learn VQ speaker and phone embeddings.

In [15], they propose a VQ technique that disentangles speaker and content information in a fully unsupervised manner for monolingual one-shot voice conversion. Phone embeddings originate from a VQ codebook whereas speaker embeddings are learned as a difference between discrete VQ codes and continuous VQ vectors. Finally, the speaker and content representations are re-combined additively (instead of by concatenation)

10.21437/SSW.2021-22

and passed to the decoder as local conditions. While the method works very well in one-shot voice conversion, it does require a target speaker sample. Since the speaker representations rely on differences between internal VQ embeddings, it is not clear how the content and speaker representations could be used externally to this system, or whether or not it works for multilingual data.

A dual-encoder VQ-VAE was proposed by [1] which modeled the phone content and F0. This approach of using two encoders and learning two VQ codebooks was also used in [2] who sought to learn speaker identity as well as speech content at the same time. In [2], they explored several variations of dual-encoder approach with different kinds of supervision. They found that the adversarial model performed disentanglement best between the speaker and content. In this paper, we utilize their pre-trained English VCTK model for multilingual adaptation as well as our experiments.

While VQ-VAE has received a lot of attention for its potential in voice conversion, other challenges remain for multilingual speech synthesis. In [16] and [17], they showed it is possible to use DNNs to synthesize voices across languages, but these methods perform speaker adaptation rather than learning embeddings that could be re-purposed. Therefore these methods require an exemplar sentence that contains specific words and phrases. Likewise [5, 18, 19] propose universal multi-language multi-speaker TTS systems, but it is not clear that the internal embeddings are re-useable for other speech tasks and the number of evaluated languages is small.

Speech is often a primary medium for communicating sensitive information such as financial details or medical information. To date, most speech privacy scenarios reflect the need to protect speaker voice characteristics [20, 21]. The work of [22] proposes shuffling audio in a speech file to transform it into a speech "bag of words" so that the content and meaning cannot be easily gleaned from ASR. Likewise [13] proposes using acoustic transformations to conceal the words of speech audio. Our approach to content privacy is inspired by [23] which created a *speech privacy sound*. However, instead of privacy for speaker identity, we mask targeted words in a phrase by manipulating the sequence of discrete VQ phone codes.

3. Data

The multilingual SIWIS dataset [24] contains four languages: English, German, French, and Italian. There are 36 unique speakers. Each speaker is bilingual or trilingual and has been recorded in two or three languages. The dataset languages were imbalanced, so our train/test splits also preserved this imbalance as shown in Table 1. The monolingual English VCTK dataset [25] contains 109 speakers with different accents. For VCTK, we used the same train/test splits as in [2]. All audio was downsampled to 16 kHz and normalized with sv56. The preprocessing steps were followed using scripts provided by [1].

Table 1: *SIWIS data splits across languages and speakers.*

Language	Training		Validation		Held-out	
	Spk	Utt	Spk	Utt	Spk	Utt
English (EN)	18	2387	18	603	4	16
French (FR)	26	3405	26	841	5	16
German (DE)	13	1719	13	376	4	18
Italian (IT)	13	1689	13	430	3	10

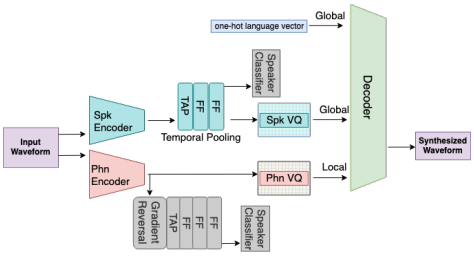


Figure 1: *VQ-VAE overview from [2], two encoders and VQ spaces which modeled speaker identity as a global condition, and speech phones as a local condition. We added a global one-hot language vector for our multilingual training.*

4. VQ-VAE Model Adaptation

We started with a dual-encoder VQ-VAE model that was pre-trained and provided by [2]. It learned two separate encoders and two separate VQ codebooks for speech content and speaker identity (Figure 1). They had trained the model to 500k steps using English VCTK data.

We used the pre-trained model from [2] and adapted it to multilingual SIWIS data. For the model adaptation, a projection layer from the pre-trained WaveRNN decoder was discarded but we kept all other parameters from the encoders and VQ codebooks. We also added a one-hot language vector as global conditions to the WaveRNN decoder. We trained the multilingual model on all four languages mixed together for 550k steps while monitoring the validation losses.

The goal is not to learn to disentangle languages, but to learn representations of content and speaker that are shared across multiple languages. For example, to learn phone VQ representations from multiple languages in a single VQ codebook. During the model adaptation, we did not experiment with changing the codebook sizes from the pre-trained model. Therefore we used a codebook size of 256 for the speaker codebook, and 512 for the phone codebook.

The input to the encoder was a waveform. After the waveform was downsampled by each encoder, it was transformed into a sequence of VQ codes and vectors for phones, and a single VQ code and vector for speaker identity. The VQ vectors were then provided to the WaveRNN decoder. Finally the output was a reconstructed waveform.

5. Task-Based Evaluation

The purpose of a task-based evaluation is to understand how learned phone or speaker representations perform in tasks that benefit from disentanglement. We describe four very small "proof-of-concept" tasks and corresponding results. The synthesized speech[1] was assessed using human listening judgements. For the listening tests, participants were recruited from the Prolific[2] platform and the listening test materials were hosted by Qualtrics[3]. We grouped our listening test tasks on the basis of language and dataset in order to utilize similar participants. This resulted in a total of seven separate listening tests and also allowed for consistency among our listener pool.

[1] Speech examples: https://rhoposit.github.io/ssw11
[2] https://www.prolific.co/
[3] https://www.qualtrics.com/uk/

Table 2: *MOS naturalness scores for copy-synthesis. Results are reported for the multilingual model (SIWIS data) as well as the monolingual English model (VCTK data).*

Data	Natural	Synthetic	Δ
SIWIS-EN	4.1	1.6	$\downarrow$ 2.5
SIWIS-FR	3.4	2.9	$\downarrow$ 0.5
SIWIS-DE	3.7	2.5	$\downarrow$ 1.2
VCTK-EN	4.0	3.3	$\downarrow$ 0.7

For example, the same set of French speakers evaluated French MOS copy-synthesis, French MOS voice transformation, and French voice transformation speaker similarity. All of our participants self-identified as "fluent" in their respective languages, including pairs for code-switching: English-French, or English-German. While the multilingual model training included Italian data, this language was omitted from the evaluation as there were few speakers in the held-out set to select representative samples for gender, as well as bilingual/trilingual overlap. For each of the seven listening tests, we recruited 20 people and they were compensated at the rate of £ 7.50 per hour.

5.1. Copy-Synthesis

One way to gauge the quality of a trained VQ-VAE is to perform copy-synthesis. If copy-synthesis quality is very good then the internal VQ representations are more likely to also be good, however this is not guaranteed. While this does not inform us about the quality of the internal representations, it provides a starting point. This section is included as a sanity check. However, since the listening test was very small the reported MOS values may not generalize.

Listeners rated the naturalness on a Likert scale of 1-5 (where 5 is natural). We evaluated 6 examples per language using data from the held-out set, for a total of 24 samples. We report the average MOS naturalness scores in Table 2. The synthetic speech results in lower MOS scores for the monolingual and multilingual models. In the multilingual model, English and German naturalness was lower. The MOS for French had the smallest change from natural to synthetic. Evaluating with higher quantities of speech samples would provide a better perspective of the average MOS scores per language.

5.2. Voice Transformation

We present results from a *voice transformation* task. We tried to change the speaker identity by replacing the speaker code to one of other codes obtained after the VQ-VAE optimization. Individual speaker codes do not always correspond to speakers included in the training dataset and hence this is not a conversion to specific identity of a target speaker. But, we would be able change the speaker identity by replacing the VQ speaker codes while keeping the VQ phone codes unchanged. For each model, we identified which VQ speaker codes had been learned during training. Neither of the two models utilized all of the possible speaker codebooks (the codebook size was 256 for both models), even though both models were trained with multi-speaker data. In the multilingual model (SIWIS), there were 11 VQ speaker codebooks utilized for 36 unique speakers. In the monolingual model (VCTK), there were 18 VQ speaker codebooks utilized for 110 unique speakers. Our VQ-VAE model under-estimated the number of speakers and seems to merge some speakers into one cluster.

5.2.1. Single-Representation

This version of voice transformation changes one single speaker VQ code at a time, without mixing or combining speaker codes. For the multilingual model, we selected one male and female speaker (**spk13**-male, **spk04**-female) from the SIWIS data and seen conditions. Then we extracted the VQ phone and speaker codes. We replaced their speaker codes with each of the 11 multilingual VQ speaker codes from the codebook. We used 2 utterances per speaker, per language for a total of 12 examples. For the one-hot language vector, we used the language from the source sentence. For the monolingual model and codebook, we followed the same approach selecting a male and female speaker from the VCTK data and seen conditions (**p229**-female-English, **p302**-male-Canadian). We selected 2 utterances for each speaker, for a total of 4 examples.

5.2.2. Mixed-Representations

This version of voice transformation mixes speaker VQ codes to create new voices, in a spirit similar to *zero-shot* voice conversion. Ideally, this could be done using various combinations of VQ speaker codes and weighting them. In this work, we mixed two representations by calculating an unweighted mean between two VQ codebook vectors. In a vector space, the resulting representation is a new centroid that is equidistant between the paired vectors. We randomly paired VQ speaker codes for each model, and then mixed them. We synthesized the same source utterances as before.

Table 3: *Multilingual (SIWIS) MOS naturalness scores for voice transformation and voice mixing.*

Speaker Code	English	French	German
Code 85	2.4	2.9	3.4
Code 192	2.6	3.0	3.1
Code 238	2.5	3.0	3.2
Code 131+248	2.4	3.1	3.3

Table 4: *Monolingual English (VCTK) MOS naturalness scores for voice transformation and voice mixing.*

Speaker Code	English
Code 67	2.3
Code 109	2.3
Code 242	2.5
Code 109+242	2.4

5.2.3. Results

For the listening tests, we randomly selected 4 speaker VQ codes (3 single-representations, 1 mixed) from each model. Participants listened to all 8 samples in their language and marked naturalness on a scale of 1 to 5. The results for MOS naturalness are provided in Table 3 and Table 4. MOS naturalness is changes depending on the speaker VQ code and language. The mixed VQ speaker vectors did not degrade the quality of the synthesized speech overall. In the multilingual model, French and German had better naturalness than English for all four of the reported VQ speaker codes. This is a similar pattern for naturalness in the earlier copy-synthesis task.

We also asked our listeners about speaker similarity. The purpose of this was to understand the consistency of the VQ

(a) *Multilingual model* (b) *Monolingual model*

Figure 2: *Voice transformation speaker VQ code similarity matrix. Annotations represent the percent of listeners who marked a pair of utterances as the same speaker. Note that the monolingual and multilingual models utilize different speaker VQ codebooks.*

speaker codes. Listeners were provided with matched and unmatched pairs in an A/B test, and were asked to decide if the A/B examples were from the same or different speaker. For example, a matched pair was 2 synthetic speech utterances using the target speaker VQ code **238**. An unmatched pair was 2 synthetic speech samples using two different speaker codes such as **238** and **85**. There were 16 total matched pairs and 24 unmatched pairs per language and dataset. This format allowed us to observe similarities and differences across a particular language and speaker VQ code. Recall that our voice transformation task did not utilize target speakers, only the learned VQ codes from the speaker codebooks. Speaker similarity results are reported in Figure 2. The annotations in the figure represent the percent of listeners who marked a pair of utterances as the same speaker. A clear diagonal would indicate that the speaker VQ codes are consistently unique. In the multilingual model codes **131+248** and **192** are less consistent. German appears to be more consistent than French or English. In the monolingual model, we observed a pair of VQ speaker codes that participants identified as being inconsistent: **67** and **242**.

Table 5: *Speaker similarity for linguistic code-switching. A/B measured how often listeners said the speaker was the same between synthetic and natural speech. Inter-Utt measured how often listeners reported consistent speaker within an utterance.*

	Speaker Similarity	
Data	A/B	Inter-Utt
English-French	57.9%	69.0%
French-English	30.8%	60.7%
English-German	67.5%	77.5%
German-English	75.0%	77.5%

5.3. Linguistic Code-Switching

The purpose the linguistic code-switching task was to find out if we could generate speech using analysis-synthesis, wherein the speech has multiple languages within the same utterance. We simulated code-switching by concatenating together VQ phone codes from utterances in different languages but from the same speaker. This was possible because the SIWIS data contained utterances from bilingual and trilingual speakers. We used the sequence of VQ phone codes from entire audio files instead of word or phrase level granularity, and we did not change or modify the VQ phone code contents. We selected 6 utterances for English and German, and 6 utterances for English and French using both male and female speakers from the held-out set. We also swapped the language order, essentially doubling the number of exemplars. This was to observe if the WaveRNN decoder is sensitive to language ordering, since the decoder could only accept a single one-hot language code. This resulted in 24 code-switched files (6 per language and order pair). For the one-hot language vector, we used the language of the first utterance. The speech was synthesized from VQ phone and speaker codes without performing any modifications to the codes apart from the concatenation.

Our main interest for this task was to find out if the multilingual model could preserve speaker similarity while also synthesizing the multilingual speech. Listeners were presented with (A) code-switched synthetic speech from concatenated VQ phone codes, and (B) code-switched speech from concatenated audio files. In this A/B test, participants were asked if the speaker was the same between the two A/B samples.

We also presented listeners with single code-switched examples from only (A) and asked the listeners to judge if the speaker voice was consistent throughout an utterance, or if it changed. This was measured because we had sometimes observed that the speaker voice was not consistent within an utterance. Results are reported in Table 5. We observed slightly more consistency for English-German pairs, compared to French. The A/B similarity for the French-English pair was particularly low, which means that the decoder had difficulty switching from French to English. This could be due to the language imbalances in the SIWIS dataset, or differences in the VQ phone code frequencies between these two languages. More investigation would uncover which part of the utterance was failing, and why the decoder was unable to recover. Better performance on German was also reflected in the other tasks.

This analysis-synthesis task does not reflect how code-switching works with speakers in real-life because it was done at the utterance level instead of the word or phrase-levels. As mentioned earlier, the purpose was to observe if the model, especially WaveRNN, is capable of it. More investigation is required to understand and quantify the limits and edge cases of VQ-VAE for code-switching. In addition, the quantity of evaluated samples was particularly small, which makes it difficult to generalize the results or draw strong conclusions. We attempted to also measure intelligibility, however the listeners did not follow instructions often enough to perform calculations of intelligibility scores. For example, some listeners identified the names of the languages rather than the words of the utterance.

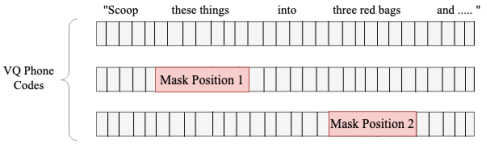


Figure 3: *Diagram showing two different content masking positions for VQ phone codes on a given phrase.*

5.4. Content-Based Privacy Masking

The purpose of exploring content-based privacy is to develop a capability that conceals certain sensitive words or phrases in a manner that does not disrupt the normal flow and feel of a speech utterance. For example, in some use-cases it might be preferable to transform a sensitive phrase into a speaker's mumbling voice instead of a cut, beep, silence or static. Different types of masks may affect speech recognition (ASR) or speaker verification (ASV) differently.

In this task, we used the monolingual model because we had reliable alignments for the VCTK data [26]. We hand-selected phrases that occurred mid-utterance and concealed them to try and render the target phrases unintelligible, while keeping the surrounding words intelligible. First, we used the forced-alignments to determine the timestamp end-points of the target phrase. Next, we used those endpoints to determine the location of the target phrase in the sequence of VQ phone codes. Finally, we modified only the VQ phone codes corresponding to the target phrase. We experimented with two different masking positions, as shown in Figure 3, as well as two different masking methods. We have taken advantage of forced-alignments in this toy problem as well as knowing the target phrases beforehand. In real-world applications it may require keyword spotting or another mechanism to decide which words and phrases get masked. Performing this in real-time versus from a speech database would introduce additional engineering challenges.

The first masking method was to replace true VQ phone codes of the target phrase with VQ phone codes from ICRA noise signals [27]. Since the noise has speech-like spectral and temporal properties, it is expected to generate speech-like, but, meaningless phone codes. The speech-shaped noise offers a non-recoverable masking, which is useful for applications where speech content redaction must be persistent. First, we analyzed this noise to obtain its VQ phone codes. Even though the noise does not truly contain phones, the resulting VQ phone code sequence represented the noise quite well. Next, we replaced the sequence of true VQ phone codes for our target phrase with a randomly selected sequence of the SSN VQ codes of the same length. Our second technique was to simply reverse the the order of the true VQ phone codes for the target phrase, while leaving the remaining VQ phone codes intact. The VQ code reversal method does render the target phrase unintelligible, however it could be recovered by playing the audio backwards. We did not attempt other masking methods, however it may be possible to use silence or randomly selected VQ phone codes. It is also unknown if VQ-VAE could be used for recoverable masking, wherein the masked could be undone. Whether or not this is desirable depends on the use-case.

5.4.1. Results

We selected two utterances that were shared between a female and male speaker. Next, we selected two target phrases to mask, at different positions in the sentence. For the first utterance, the two target phrases were "these things" (position1) and "three red bags" (position2). For the second utterance, the two target phrases were "sunlight strikes" (position1) and "raindrops in the air" (position2). In total, 16 examples were evaluated.

Participants were instructed that one or more words had been removed from the utterance, but were not told which ones. They were asked whether or not the speaker voice was consistent throughout the utterance and we measured the proportion of positive responses as shown in Table 6. Overall the SSN was better for maintaining speaker identity throughout the utterance. In general, masking the phrase at position2 resulted in more consistency, which could be due to the challenges of using an auto-regressive decoder like WaveRNN. Listeners also performed an A/B preference test which revealed a slight preference for SSN over reversal masking. Finally, we measured ASR-based intelligibility as word error rate (WER) using the IBM Watson Speech-to-Text API[4]. We first calculated the WER on natural, unmasked audio as a baseline and found it was 24%. This is higher than expected but likely due to pronunciations and the audio quality. The other WER is reported in Table 6. Overall, the WER increased compared to natural, unmasked speech. The position1 resulted in better intelligibility, and the two different techniques were comparable on average. It is unclear if the rise in WER is due to the masking or if intelligibility was lost for unmasked words. Future work must provide a procedure to better evaluate content-based masking.

Table 6: *Speaker similarity and ASR-based WER for content masking, comparing two methods and target phrase positions.*

Masks	Speaker Similarity	ASR-Based WER
Reversal Position1	63.7%	47%
Reversal Position2	77.5%	68%
SSN Position1	70.0%	53%
SSN Position2	76.2%	61%

6. Discussion

We have shown that it is possible to adapt an existing monolingual VQ-VAE model to a new multi-speaker multi-language dataset with reasonable performance on copy-synthesis, voice transformation, and linguistic code-switching[5]. This is an important finding for multi-lingual speech synthesis.

The manner in which the VQ speaker codebooks are underutilized for both models has some implications for the limitations of the VQ-VAE architecture. It is sometimes referred to as *codebook collapse* analogous to posterior collapse in VAE. We observed similar codebook collapse in our VQ phone codebooks as the VQ speaker codebooks. In both models, the phone codebook size was set to 256, however the multilingual model utilized 161 entries and the monolingual model utilized 170 entries. The quantity of utilized entries is far greater than the size of a requisite phone set – even in the multilingual model. We examined the distribution of VQ phone codes for each language in the multilingual model and found that all four languages utilized similar codebooks with similar frequencies.

[4] https://www.ibm.com/cloud/watson-speech-to-text

[5] Code/models: https://github.com/rhoposit/multilingual_VQVAE

The diversity of the learned codebooks should be improved. The size of codebooks must be pre-determined at the time of initializing the architecture. As we have shown, VQ-VAE models can be adapted to new datasets, but having hard-coded constraints (such as the codebook sizes) may be a limiting factor. Our recommendation is to develop a way to dynamically add or remove VQ codebooks during the training process. This would make it possible to learn only and all of the codebook vectors that matter. The true capabilities of VQ-VAE modeling are limited by its toolkit implementation: the nature of the tensor graph and how it is used in memory does not accommodate dynamic modeling to its fullest potential.

We have described a method to synthesize high-quality speech in multiple languages (including code-switching) from a single multilingual model, based on learned representations. This will be useful for speech-to-speech translation, controllable speech synthesis, and data augmentation. In future work, we are interested in adding additional internal representations to the dual-encoder VQ-VAE model in an effort to perform further disentanglement of speech signal characteristics.

7. Acknowledgements

We sincerely thank Evelyn Williams at the University of Edinburgh for helping implement the listening tests. This work was partially supported by the EPSRC Centre for Doctoral Training in Data Science, funded by the UK Engineering and Physical Sciences Research Council (grant EP/L016427/1) and University of Edinburgh; and by a JST CREST Grant (JPMJCR18A6, VoicePersonae project), Japan. Some of the numerical calculations were carried out on the TSUBAME 3.0 supercomputer at the Tokyo Institute of Technology.

8. References

[1] Y. Zhao, H. Li, C.-I. Lai, J. Williams, E. Cooper, and J. Yamagishi, "Improved Prosody from Learned F0 Codebook Representations for VQ-VAE Speech Waveform Reconstruction," *INTERSPEECH*, 2020.

[2] J. Williams, Y. Zhao, E. Cooper, and J. Yamagishi, "Learning Disentangled Phone and Speaker Representations in a Semi-Supervised VQ-VAE Paradigm," *ICASSP*, 2021.

[3] A. v. d. Oord, O. Vinyals, and K. Kavukcuoglu, "Neural Discrete Representation Learning," *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, 2017.

[4] Y. Yasuda, X. Wang, and J. Yamagishi, "End-to-End Text-to-Speech Using Latent Duration Based on VQ-VAE," *ICASSP*, 2021.

[5] Y. Zhang, R. J. Weiss, H. Zen, Y. Wu, Z. Chen, R. Skerry-Ryan, Y. Jia, A. Rosenberg, and B. Ramabhadran, "Learning to Speak Fluently in a Foreign Language: Multilingual Speech Synthesis and Cross-Language Voice Cloning," *INTERSPEECH*, 2019.

[6] P. Dhariwal, H. Jun, C. Payne, J. W. Kim, A. Radford, and I. Sutskever, "Jukebox: A Generative Model for Music," *arXiv preprint arXiv:2005.00341*, 2020.

[7] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-End Factor Analysis for Speaker Verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2010.

[8] D. Raj, D. Snyder, D. Povey, and S. Khudanpur, "Probing the Information Encoded in X-Vectors," *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, Dec 2019.

[9] R. Peri, H. Li, K. Somandepalli, A. Jati, and S. Narayanan, "An empirical analysis of information encoded in disentangled neural speaker representations," in *Proc. Odyssey 2020 The Speaker and Language Recognition Workshop*, 2020, pp. 194–201.

[10] J. Williams and S. King, "Disentangling Style Factors From Speaker Representations." in *INTERSPEECH*, 2019, pp. 3945–3949.

[11] Y.-A. Chung, H. Tang, and J. Glass, "Vector-quantized autoregressive predictive coding," *Proc. Interspeech 2020*, pp. 3760–3764, 2020.

[12] J. Ebbers, M. Kuhlmann, T. Cord-Landwehr, and R. Haeb-Umbach, "Contrastive Predictive Coding Supported Factorized Variational Autoencoder for Unsupervised Learning of Disentangled Speech Representations," *ICASSP*, 2021.

[13] S. H. K. Parthasarathi, H. Bourlard, and D. Gatica-Perez, "Wordless Sounds: Robust Speaker Diarization Using Privacy-Preserving Audio Representations," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 1, pp. 85–98, 2012.

[14] S. Ding and R. Gutierrez-Osuna, "Group Latent Embedding for Vector Quantized Variational Autoencoder in Non-Parallel Voice Conversion." in *INTERSPEECH*, 2019, pp. 724–728.

[15] D.-Y. Wu and H.-y. Lee, "One-Shot Voice Conversion by Vector Quantization," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7734–7738.

[16] I. Himawan, S. Aryal, I. Ouyang, S. Kang, P. Lanchantin, and S. King, "Speaker Adaptation of a Multilingual Acoustic Model for Cross-Language Synthesis," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7629–7633.

[17] X. Zhou, H. Che, X. Wang, and L. Xie, "A Novel Cross-Lingual Voice Cloning Approach with a Few Text-Free Samples," *arXiv preprint arXiv:1910.13276*, 2019.

[18] J. Yang and L. He, "Towards Universal Text-to-Speech," *Proc. Interspeech 2020*, pp. 3171–3175, 2020.

[19] B. Li, Y. Zhang, T. Sainath, Y. Wu, and W. Chan, "Bytes are all you need: End-to-End Multilingual Speech Recognition and Synthesis with Bytes," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 5621–5625.

[20] J. Qian, F. Han, J. Hou, C. Zhang, Y. Wang, and X.-Y. Li, "Towards Privacy-Preserving Speech Data Publishing," in *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE, 2018, pp. 1079–1087.

[21] N. Tomashenko, B. M. L. Srivastava, X. Wang, E. Vincent, A. Nautsch, J. Yamagishi, N. Evans, J. Patino, J.-F. Bonastre, P.-G. Noé *et al.*, "Introducing the VoicePrivacy Initiative," *INTERSPEECH*, 2020.

[22] S. Ahmed, A. R. Chowdhury, K. Fawaz, and P. Ramanathan, "Preech: A System for Privacy-Preserving Speech Transcription," in *29th USENIX Security Symposium (USENIX Security 20)*, 2020, pp. 2703–2720.

[23] K. Hashimoto, J. Yamagishi, and I. Echizen, "Privacy-preserving sound to degrade automatic speaker verification performance," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 5500–5504.

[24] J.-P. Goldman, P.-E. Honnet, R. Clark, P. N. Garner, M. Ivanova, A. Lazaridis, H. Liang, T. Macedo, B. Pfister, M. S. Ribeiro *et al.*, "The SIWIS Database: A Multilingual Speech Database With Acted Emphasis," in *Proceedings of Interspeech*, no. CONF, 2016.

[25] J. Yamagishi, C. Veaux, K. MacDonald *et al.*, "CSTR VCTK Corpus: English Multi-Speaker Corpus for CSTR Voice Cloning Toolkit (version 0.92)," 2019.

[26] M. McAuliffe, M. Socolof, S. Mihuc, M. Wagner, and M. Sonderegger, "Montreal Forced Aligner: Trainable Text-Speech Alignment Using Kaldi," *Proc. Interspeech 2017*, 2017.

[27] W. A. Dreschler, H. Verschuure, C. Ludvigsen, and S. Westermann, "IcraNoises: Artificial Noise Signals with Speech-Like Spectral and Temporal Properties for Hearing Instrument Assessment," *Audiology*, vol. 40, no. 3, pp. 148–157, 2001.

11th ISCA Speech Synthesis Workshop (SSW 11)
26-28 August 2021, Budapest, Hungary

Text-to-Speech Synthesis Techniques for MIDI-to-Audio Synthesis

Erica Cooper[†], *Xin Wang*[†], *Junichi Yamagishi*

National Institute of Informatics, Chiyoda-ku, Tokyo, Japan

ecooper@nii.ac.jp, wangxin@nii.ac.jp, jyamagis@nii.ac.jp

Abstract

Speech synthesis and music audio generation from symbolic input differ in many aspects but share some similarities. In this study, we investigate how text-to-speech synthesis techniques can be used for piano MIDI-to-audio synthesis tasks. Our investigation includes Tacotron and neural source-filter waveform models as the basic components, with which we build MIDI-to-audio synthesis systems in similar ways to TTS frameworks. We also include reference systems using conventional sound modeling techniques such as sample-based and physical-modeling-based methods. The subjective experimental results demonstrate that the investigated TTS components can be applied to piano MIDI-to-audio synthesis with minor modifications. The results also reveal the performance bottleneck – while the waveform model can synthesize high quality piano sound given natural acoustic features, the conversion from MIDI to acoustic features is challenging. The full MIDI-to-audio synthesis system is still inferior to the sample-based or physical-modeling-based approaches, but we encourage TTS researchers to test their TTS models for this new task and improve the performance.

Index Terms: music audio synthesis, text to speech synthesis, deep learning, Tacotron

1. Introduction

Speech and music are human universals, and they have been the theme of numerous research topics in the social and natural sciences. From an engineering perspective, both speech and music information processing deal with symbolic and acoustic data, i.e., symbolic music notes or text, and acoustic music or speech audio signals. This similarity makes it possible to share methodologies across disciplines, especially those based on data-driven deep learning. For example, AI music composition, which learns a distribution of music notes to generate new songs, is based on language models for text data. Automatic audio transcription, which converts music into a sequence of notes, uses similar techniques to speech recognition such as Viterbi decoding and DNNs for classification tasks.

From the music notes to instrumental audio signals[1], however, cross-disciplinary techniques are less explored even though the concept is very similar to text-to-speech (TTS) synthesis. It has not been until recently that some deep learning models, such as WaveNet [1, 2], have been used in both tasks. While research in both fields has investigated other related models such as GAN [3, 4] and VAE [5, 6], most studies focus on music audio-to-audio mapping tasks. The question of how and to what extent TTS approaches can be applied to music audio generation remains to be explored.

This study is our initial step to address the aforementioned question. We focus on MIDI-to-audio synthesis for piano because of the available data, but the methodology is expected to be applicable to many other instruments. In Section 2, we compare TTS to music audio generation from MIDI and explain the possibility of using TTS methods for the MIDI synthesis task. In Section 3, we explain the MIDI-to-audio systems that use many components from TTS, including Tacotron [7] and neural source-filter (NSF) waveform model [8]. We introduce modifications to those components to account for the intrinsic differences between music and speech. Furthermore, we introduce MIDI-specific acoustic features and compare them with the Mel-spectrogram for MIDI-to-audio synthesis.

Based on a subjective evaluation, our study tentatively suggests that many TTS techniques can be adapted to music audio generation with slight modifications, and we observed trends similar to those in TTS. For audio generation, the NSF models, which were originally created for speech modeling, can produce high-quality polyphonic piano sound in the copy-synthesis scenario. For acoustic modeling, the Tacotron-based models demonstrated competitive performance to produce acoustic features from the MIDI piano roll input. However, the conversion from MIDI to acoustic features is the most challenging task, similar to the bottleneck in TTS. Hence, we encourage TTS researchers to extend their research outcomes to the music audio generation task.

2. Using TTS techniques for MIDI-to-audio synthesis

This paper focuses on music audio generation from music transcription data in MIDI format. As illustrated in Figure 1, the MIDI-to-audio synthesis process is similar to TTS since both convert symbolic data into audio signals.

In both cases, the front-end converts the input text or MIDI raw data into a representation as input to the acoustic model. In the case of statistical parametric TTS [9], it is a sequence of context vectors $\boldsymbol{x}_{1:N} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_N)$, where N is the number of frames, and where the vector for the n-th frame is $\boldsymbol{x}_n$. Each $\boldsymbol{x}_n$ encodes linguistic information such as the phone and syllable identities. In the case of the MIDI-to-audio generation, the input MIDI contains messages that encode the time of note onset and offset, velocity, and other events. For processing using deep learning models, the MIDI input is usually represented as a piano roll[2] that can also be written as a sequence of vectors $\boldsymbol{x}_{1:N}$, and each $\boldsymbol{x}_n$ is a 128-dimensional vector in which each dimension encodes the velocity for one of the 128 MIDI notes[3]. Figure 2 illustrates the difference.

Accordingly, conversion from $\boldsymbol{x}_{1:N}$ to $\boldsymbol{o}_{1:T}$ can use similar models for both tasks. These methods can be grouped into two

[†]Equal contribution

[1]In this paper, we focus on musical instrument sounds rather than singing voice synthesis.

[2]Although it is called piano roll, it can be used for other instruments.

[3]The sustain pedal information can be reflected as elongation of notes, encoded by extending the note across multiple frames.

10.21437/SSW.2021-23

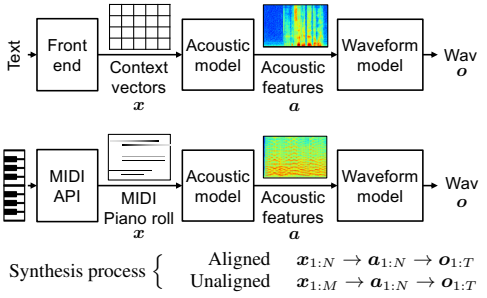


Figure 1: *Comparing TTS and MIDI-to-audio synthesis. Temporal length of synthesized waveform $\boldsymbol{o}_{1:T}$ and acoustic feature $a_{1:N}$ is related by the frame rate L, i.e., $T = N \times L$.*

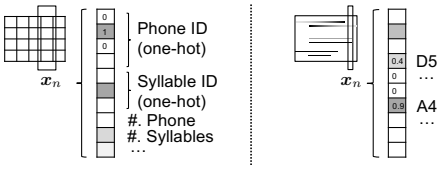


Figure 2: *Comparing contextual vector for statistical parametric TTS (left) and MIDI piano roll (right).*

categories from the perspective of MIDI-to-audio generation. The first group is for applications where the audio is required to be consistent with the timing information in the MIDI input. In implementation, the synthesis process has to be $\boldsymbol{x}_{1:N} \rightarrow \boldsymbol{a}_{1:N} \rightarrow \boldsymbol{o}_{1:T}$, where the MIDI piano roll and acoustic feature sequences have the same length N, and where the audio waveform length T is related to the feature length by a fixed frame shift L, i.e., $T = N \times L$. This pipeline is very similar to neural statistical parametric TTS [9], and we can use various types of neural networks [10, 11] for $\boldsymbol{x}_{1:N} \rightarrow \boldsymbol{a}_{1:N}$ and use neural waveform models [12, 13] or vocoders [14, 15] for $\boldsymbol{a}_{1:N} \rightarrow \boldsymbol{o}_{1:T}$. Note that $\boldsymbol{x}_{1:N}$ for TTS should have obtained the duration information from a duration model, while $\boldsymbol{x}_{1:N}$ for MIDI can retrieve the duration information from raw MIDI.

The second type of MIDI-to-audio application converts MIDI corresponding to a basic musical score into a professional "performance" with rich and dynamic expressivity, or transfers a particular "performance style" to the generated audio. In this case, $\boldsymbol{x}_{1:M}$ from input MIDI may not be aligned with the desired output sequence $\boldsymbol{a}_{1:N}$. Accordingly, the acoustic model should be capable of $\boldsymbol{x}_{1:M} \rightarrow \boldsymbol{a}_{1:N}$, and many attention-based sequence-to-sequence models [7, 16] are suitable for this task.

Note that it is also possible to use a single model to directly convert the the input piano roll into an audio waveform, which is similar to WaveNet for TTS [1].

3. Experimental systems for time-aligned MIDI to piano audio synthesis

In this paper, we focus on systems that convert time-aligned MIDI to musical instrument audio waveforms and choose piano as the target instrument. Figure 3 illustrates the systems as four groups. While all of them use an NSF-based waveform model, the first two types contain intermediate acoustic models, and the remaining two directly convert the input MIDI into an output waveform. Another difference among the systems is the type

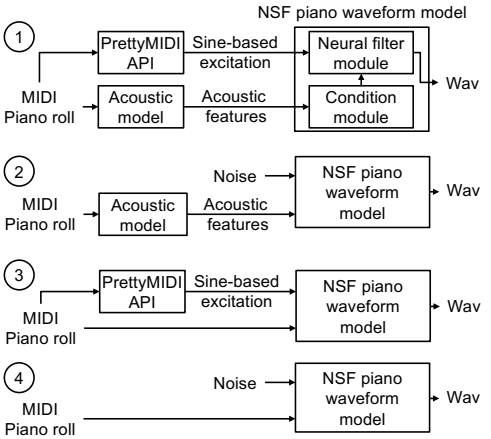


Figure 3: *Four types of experimental MIDI-to-audio systems using NSF piano waveform model.*

of the excitation signal, which will be detailed in Section 3.3. Note that the experimental systems introduced in this section are applicable to other monophonic or polyphonic instruments.

3.1. Acoustic models

For the systems that use separate acoustic models, we can use many types of neural networks that conduct $\mathbb{R}^{N \times D} \rightarrow \mathbb{R}^{N \times D'}$. Although it is originally designed for sequence-to-sequence mapping, we investigate variants of Tacotron, hoping that the outcome can accelerate our research on the unaligned MIDI-to-audio synthesis task in the near future.

Our Tacotron acoustic model implementation was based on [17]. Tacotron was modified to accept a sequence of 128-dimensional MIDI piano roll frames as input instead of a text or phoneme sequence. Tacotron's encoder learns an embedding that maps a symbolic token to an embedding vector, but since piano roll frames are not strictly symbolic but are already a meaningful vector representation of pitch and velocity, we replaced this embedding layer with a dense projection layer. Aside from these initial modifications of the encoder to accept MIDI piano roll input instead of text, the model architecture is otherwise the same as in [17]. The encoder consists of a CBHG module followed by a self-attention block. The outputs of both are input to the decoder, where the CBHG output is input to a forward attention mechanism and the self-attention output is received by an additive attention mechanism. The decoder consists of a decoder recurrent layer followed by self-attention, and finally, a post-net conducts spectral shaping and enhancement for the final spectrogram output.

Because music sequences are much longer than the typical short utterances used for training text-to-speech synthesis models, and in particular, the piano roll input representation is the same length as the output spectrogram, as opposed to text or phoneme sequences which are much shorter, we had to take a number of steps to fit the training sequences into memory and to learn alignments well. First, we found that it was necessary to segment the data into shorter sequences, starting with 200 frames. Next, we wished to reduce the autoregressive dependencies and instead force the model to rely more directly on the inputs. The prenet to the decoder receives

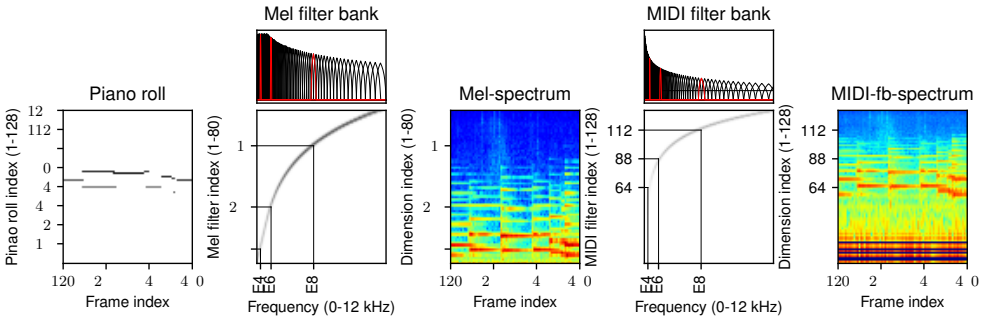


Figure 4: *Mel-spectrogram versus MIDI-filter-bank spectrogram*

the previous predicted spectrogram frame (or the previous actual spectrogram frame, when teacher-forcing is enabled), so we increased dropout at the prenet to the decoder from its default value of 0.5 to higher values of 0.9, 0.95, or 0.99, finding in initial experiments that the 0.99 dropout rate produced the best-aligned predictions. Next, we tried a downsampling approach where we downsampled the input piano roll sequence by a factor of either 2 or 4, and effectively downsampled the output spectrogram as well by setting the reduction factor to 2 or 4, so that the model predicts that many output frames at each timestep. We found that a downsampling factor of 4 produced the best alignments, and that combining this downsampling with the dropout to the decoder prenet at a rate of 0.99 produced stable alignments for sequences as long as 800 frames, so we chose this as our base model, which we call `taco2`.

As an additional method to force the model to use the input sequence more directly, we concatenated the current MIDI piano roll frame with the previous spectrogram frame at the decoder prenet (after that spectrogram frame has been dropped out). We call this model configuration `taco3`, and we warm-start its parameters from `taco2`. Finally, to see whether up-sampling the data again would improve the synthesis quality, model `taco3` uses none of these prenet or downsampling tricks, and its parameters are also warm-started from `taco2`.

As reference, we included a CNN-based network that conducts $\mathbb{R}^{N\times D} \rightarrow \mathbb{R}^{N\times D'}$. This network called PerformanceNet combines U-Net and multi-band convolution blocks to convert MIDI piano rolls into acoustic features and has shown good performance on MIDI-to-audio synthesis [18]. Samples from all systems can be heard online[4].

3.2. MIDI filter-bank features

Many TTS systems use Mel-spectrogram or Mel-cepstral coefficients as the output of the acoustic model, but the Mel scale may not be the best for music applications. Since each MIDI note d and its corresponding frequency f is related by

$$f = 2^{\frac{d-69}{12}} \times 440, \tag{1}$$

where 69 and 440 are the MIDI index and frequency value of note A4, respectively, we can define a new filter bank where the k-th filter is centered around $f = 2^{(k-69)/12} \times 440$ Hz. Figure 4 plots the resulting MIDI-based filter bank.

We apply the MIDI-based filter bank to extract low-dimensional spectral features in a similar manner to the Mel-spectrogram. Figure 4 also compares the Mel-spectrogram and MIDI-filter-bank-based spectra. Notice that the bars in the MIDI-filter-bank-based spectra resemble the corresponding piano roll. Note that the MIDI-filter-bank spectra have empty dimensions in the low frequency region because some of the filters do not cover any discrete frequency bin. These empty dimensions can be filled in if we increase the FFT points.

[4]https://nii-yamagishilab.github.io/samples-xin/main-midi2audio.html

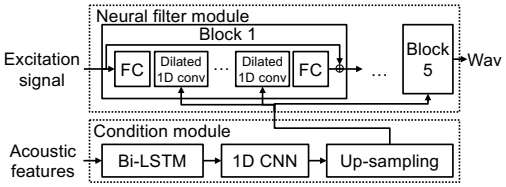


Figure 5: *NSF piano waveform model. Block in neural filter module is based on the simplified dilated-CNN filter block (cf. Figure.4 in [8]). FC denotes a fully-connected layer.*

3.3. NSF-based piano waveform model

Our NSF-based piano waveform model is based on the simplified NSF model [8]. As Fig. 5 illustrates, the NSF piano waveform model contains a condition module that transforms and up-samples the input frame-level acoustic features and a neural filter module that converts the up-sampled features and an excitation signal into an output waveform through multiple dilated convolution blocks.

A major difference between NSF waveform models for piano and speech is the source module. While the source module for speech produces an excitation signal from fundamental frequencies (F0s), such a module cannot be used for polyphonic piano sound. One solution is to use noise as the excitation (e.g., ② and ④ in Fig. 3). An alternative solution is to derive a sine-based excitation signal from the input time-aligned MIDI. In this paper, we use the PrettyMIDI synthesis API [19][5] to produce a polyphonic sine-based excitation signal (e.g., ① and ③ in Fig. 3) from the piano roll notes.

4. Experiments

4.1. Database and protocol

Experiments were conducted using the MAESTRO database (V2.0.0)[6] [2]. This is a large-scale database that contains

[5]https://craffel.github.io/pretty-midi/

[6]https://magenta.tensorflow.org/datasets/maestro

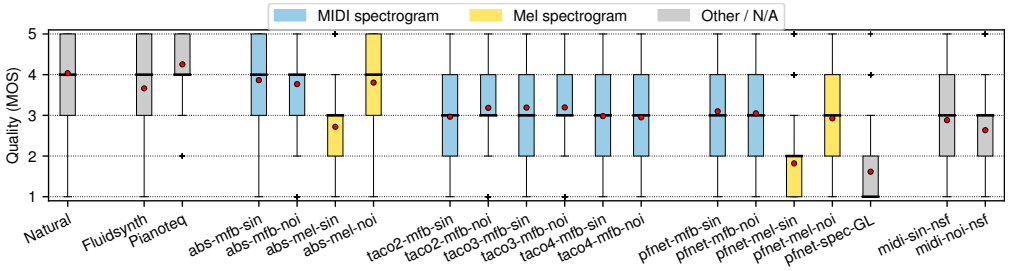


Figure 6: *Boxplots of MOS per system. Red dots denote mean of MOS.*

over 200 hours of piano performances and aligned MIDI data from the International Piano-e-Competition. Both the audio and MIDI data were recorded when the competing virtuoso pianists performed on concert-quality acoustic grand pianos with integrated MIDI capture and playback systems.

The experiments followed the official data protocol: a train set with 161.3 hours of data from 967 performances, a validation set with 19.4 hours of data from 137 performances, and a test set with 20.5 hours of data. Because it is impossible to evaluate the entire test set in subjective evaluation, 192 test segments were manually excerpted from the test set, and each test segment was less than 30 seconds in duration.

4.2. Experimental systems and feature configurations

Table 1 lists the systems investigated in the experiments. The first two are reference software synthesizers, and the next four are copy-synthesis systems that directly use natural acoustic features (i.e., Mel-spectrogram or MIDI-based filter bank features) as the input to the NSF model for piano waveform generation. They simulate the ideal case where acoustic models convert the input MIDI into the acoustic features perfectly. The next 11 systems are pipelines of an acoustic model, which is either a variant of the Tacotron or the PerformanceNet model, and the NSF waveform model. The last two experimental systems, namely `midi-sin-nsf` and `midi-noi-nsf`, directly convert the MIDI and the excitation signals into the waveform through NSF models.

We trained Tacotron models using the MIDI filter bank spectrogram as output, since we found initially that this produced better alignments than using Mel spectrograms.[7] The models were trained on segments of 800 frames using the Adam optimizer, a batch size of 4, and a learning rate of 0.0001. The base model `taco2` was trained for 550k steps until spectrogram loss on the development set had converged. The other two models `taco3` and `taco4` had their parameters warm-started from `taco2`, and were trained for an additional 250k steps and 50k steps to convergence, respectively. The PerformanceNet-based acoustic models (`PFNet`) were trained for 50 epochs using the Adam optimizer with a batch size of 16.

All the NSF models were trained using an Adam optimizer with a learning rate of 1×10^{-4} and [20]. The maximum number of training epochs is 20, and each epoch took around 24 hours. Due to the limited GPU memory size, the input and output data for NSF models was truncated into segments of 3s in duration, and the batch size was set to 5. During generation, the NSF models produced the waveform without truncation.

As a reference, the original PerformanceNet was included in the experiment. This system uses spectrograms as the acoustic feature and produces a waveform from the generated spectrogram through the Griffin-Lim (GL) algorithm [21]. Two software synthesizers were also included for reference: an open-source software called `Fluidsynth`[8] and a commercial one called `Pianoteq`[9]. While both produce piano audio given MIDI input, the former uses a sampling-based approach, and the latter is based on physical modeling of pianos.

Table 1: *Experimental systems. Pitch accuracy is measured using cross-entropy, the lower the better.*

System ID	Acoustic model	Acoustic feature	Excit. signal	Wave. model	Pitch mismatch note	Pitch mismatch chord	MOS (mean)
Natural	-	-	-	-	-	-	4.04
`Fluidsynth`	Sample-based MIDI-to-audio software				5.20	6.77	3.66
`Pianoteq`	Physical-model MIDI-to-audio software				4.82	6.50	4.25
`abs-mfb-sin`	-	midi-fb	sine	NSF	-	-	3.87
`abs-mfb-noi`	-	midi-fb	noise	NSF	-	-	3.77
`abs-mel-sin`	-	mel-spc.	sine	NSF	-	-	2.72
`abs-mel-noi`	-	mel-spc.	noise	NSF	-	-	3.81
`taco2-mfb-sin`	`taco2`	midi-fb	sine	NSF	4.61	6.34	2.97
`taco2-mfb-noi`	`taco2`	midi-fb	noise	NSF	4.66	6.36	3.18
`taco3-mfb-sin`	`taco3`	midi-fb	sine	NSF	4.78	6.48	3.19
`taco3-mfb-noi`	`taco3`	midi-fb	noise	NSF	4.89	6.53	3.19
`taco4-mfb-sin`	`taco4`	midi-fb	sine	NSF	4.86	6.39	2.98
`taco4-mfb-noi`	`taco4`	midi-fb	noise	NSF	4.97	6.42	2.95
`pfnet-mfb-sin`	PFNet	midi-fb	sine	NSF	5.59	7.14	3.10
`pfnet-mfb-noi`	PFNet	midi-fb	noise	NSF	5.78	7.26	3.05
`pfnet-mel-sin`	PFNet	mel-spec.	sine	NSF	5.66	7.17	1.82
`pfnet-mel-noi`	PFNet	mel-spec.	noise	NSF	5.74	7.25	2.93
`pfnet-spec-GL`	PFNet	spec.	-	GL	5.43	6.98	1.62
`midi-sin-nsf`	-	-	sine	NSF	4.32	6.40	2.88
`midi-noi-nsf`	-	-	noise	NSF	4.40	6.08	2.63

The audio waveforms from MAESTRO were down-sampled to 24kHz and encoded through 16-bit PCM. The Mel-spectrogram was then extracted using a frame length of 50ms, a frame shift of 12 ms, FFT with 2048 points, and 80 overlapped triangular filters evenly spaced on the Mel-frequency scale. The MIDI-based filter bank features were extracted using a similar configuration but with a filter bank based on the MIDI notes (Section 3.2). The spectrogram for `pfnet-spec-GL` used the original recipe [18]. The MIDI files were converted into 128-dimensional piano rolls using the PrettyMIDI API.

4.3. Subjective evaluation and results

We conducted a crowdsourced listening test to evaluate the quality of audio from our synthesizers, comparison synthesizers, and natural audio. We included 120 samples from each of the 20 systems, and obtained mean opinion score (MOS) ratings for each sample from five different listeners on a scale from 1 (very bad) to 5 (very good). Listeners were instructed

[7] Alignment errors were reduced from 21% using Mel spectrograms to 3% using MIDI spectrograms.

[8] https://www.fluidsynth.org/

[9] https://www.modartt.com/pianoteq

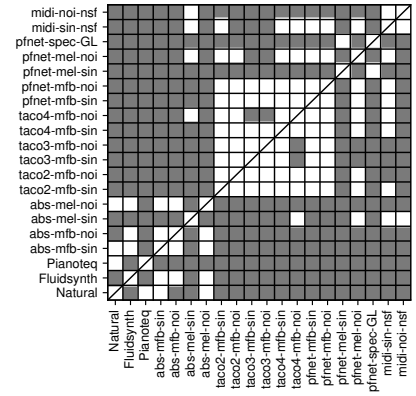


Figure 7: *Results of two-sided Mann-Whitney test with Holm-Bonferroni correction. A grey block indicates a statistically significant difference at* $\alpha = 0.05$.

to evaluate the overall quality of the piano sound subjectively. Each test set contained 30 different samples, balanced to contain at least one sample from each of the 20 systems. Listeners were permitted to complete up to 10 different sets. In total, we received ratings from 224 unique listeners. A box plot of the MOS ratings for each of the 20 systems can be seen in Figure 6, and significant differences from a Holm-Bonferroni-corrected two-sided Mann-Whitney U test at a level of $\alpha = 0.05$ are shown in Figure 7. The mean of MOS is also listed in Table 1.

We found that the `Pianoteq` physical synthesis was significantly preferred over every other synthesis method, and was even rated higher than natural piano audio (although not significantly so). Other systems which were not significantly different from natural audio were two of the analysis-by-synthesis systems, `abs-mfb-sin` and `abs-mel-noi`. There were not many statistically-significant differences between the Tacotron models, with the exception of `taco4-mfb-noi`, which was significantly worse than both of the `taco3` systems. As for PFNet-based systems, the two that used the MIDI filter bank representation had about equivalent performance to the Tacotrons with no significant differences, whereas `pfnet-mel-sin` and `pfnet-spec-GL` were significantly worse than all Tacotrons.

Comparing the standard Mel filter bank to the proposed MIDI filter bank, there are two significant differences favoring MIDI, `pfnet-mel-sin` vs. `pfnet-mfb-sin` and `abs-mel-sin` vs. `abs-mfb-sin`. For noise vs. sine wave excitation, there are two significant differences favoring noise, `pfnet-mel-sin` vs. `pfnet-mel-noi` and `abs-mel-sin` vs. `abs-mel-noi`.

4.4. Objective evaluation and results

We first measured the pitch accuracy of the synthesized audio. We trained a CNN-based F0 estimator called CREPE [22] on the MAESTRO training set. Although the original CREPE is designed for monophonic sound, it can be modified for polyphonic piano sound by replacing the target from one-hot vectors to multi-hot ones. During the pitch detection stage, we extract the pitch probability sequence $\boldsymbol{p}_{1:N} = (\boldsymbol{p}_1, \cdots, \boldsymbol{p}_N)$ from the input audio $\boldsymbol{o}_{1:T}$ by $\boldsymbol{p}_{1:N} = \texttt{CREPE}(\boldsymbol{o}_{1:T})$, where each $\boldsymbol{p}_n = [p_{n,1}, \cdots, p_{n,128}]$, and where $p_{n,k} \in (0, 1)$ indicates the probability of observing the k-th MIDI note at the n-th frame. Then, the cross entropy between $\boldsymbol{p}_{1:N}$ and the input piano roll $\boldsymbol{x}_{1:N}$ can be computed to measure the pitch mismatch $\texttt{CE} = -\sum_{n=1}^{N}\sum_{k=1}^{128} x_{n,k} \log p_{n,k}$. Hence, a lower `CE` indicate less severe mismatch.

We created piano rolls and synthesized audio for around 100 individual notes and chords and synthesized their audio. We measured cross entropy and listed results in Table 1. It can be observed that, when using separate acoustic and waveform models, the systems with sine excitation outperformed their counterparts using noise excitation, but the systems using only the NSF waveform model achieved lower `CE` values. Furthermore, Tacotron models have lower mismatches compared to PFNet systems using the same vocoder. However, lower pitch mismatch does not lead to higher MOS. This indicates that the perceptual quality of piano audio is not only affected by pitch accuracy. Another hypothesis is that amateur listeners may not be able to detect mild pitch mismatch.

5. Discussion

The evaluation results suggest that the physical-model-based approach outperformed the other deep-learning-based MIDI-to-audio systems and is even slightly better than the original audio in MAESTRO. The original audio was recorded over many years, and we observed that the MOS of the audio in year 2008 and 2014 were less than 4.0. This variation of quality may also affect the MAESTRO training set and the deep-learning-based models trained using this data. On the other hand, the physical-model-based approach is free from such artifacts in data. However, the physical model is the outcome of laborious analysis and simulation [23, 24, 25], which does not easily generalize to another piano type or instrument. In contrast, deep-learning-based models are more flexible, and this study showed examples of using TTS models for music generation.

Performance of the investigated deep-learning models is not satisfying yet. Since listeners may be more sensitive to the artifacts in music signals, this sets a high standard for producing natural audio but also leaves large room for improvement. One potential direction for future work is to combine data-driven techniques with physical models of piano sounds. Rather than learning a physical sound from scratch, sound physics may offer effective prior knowledge.

6. Conclusions

This study is our initial step to investigate the possibility of using TTS models for MIDI-to-audio synthesis. The two disciplines differ in many aspects but both deal with the mapping from one sequence of data into another. Based on the similarities and differences, we modified the TTS components, namely Tacotron and NSF, and introduced a MIDI filter-bank acoustic feature set for the MIDI-to-audio task, which improved alignments for Tacotron and resulted in more preferred audio. Based on subjective evaluation of the TTS-like systems, natural audio, physical piano model, and other reference systems, we observed promising results when using TTS components for MIDI-to-audio generation. We also identified the quality bottleneck when converting the MIDI input into acoustic features, a similar bottleneck to that in TTS. Hence, future work will explore more different types of acoustic and waveform models, and we encourage TTS researchers to extend their knowledge and practices to this challenging task of MIDI-to-audio generation.

7. Acknowledgements

A part of the computations were performed on the TSUBAME 3.0 supercomputer at Tokyo Institute of Technology. This work is supported by "TSUBAME Encouragement Program for Young/Female Users" of Global Scientific Information and Computing Center at Tokyo Institute of Technology and by "Joint Usage/Research Center for Interdisciplinary Large-scale Information Infrastructures" in Japan, and by JST CREST Grants (JPMJCR18A6 and JPMJCR20D3), JST AIP Challenge Grant, and MEXT KAKENHI Grants (18H04112, 21H04906, 21K11951, 21K17775), and a grant by KAWAI foundation for sound technology & music (year 2020, No.4), Japan.

8. References

[1] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "WaveNet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[2] C. Hawthorne, A. Stasyuk, A. Roberts, I. Simon, C.-Z. A. Huang, S. Dieleman, E. Elsen, J. Engel, and D. Eck, "Enabling factorized piano music modeling and generation with the MAESTRO dataset," in *Proc. ICLR*, 2018.

[3] J. Engel, K. K. Agrawal, S. Chen, I. Gulrajani, C. Donahue, and A. Roberts, "GANSynth: Adversarial neural audio synthesis," in *Proc. ICLR*, 2019.

[4] K. Kumar, R. Kumar, T. de Boissiere, L. Gestin, W. Z. Teoh, J. Sotelo, A. de Brébisson, Y. Bengio, and A. C. Courville, "MelGAN: Generative adversarial networks for conditional waveform synthesis," in *Proc. NIPS*, 2019, pp. 14 910–14 921.

[5] F. Roche, T. Hueber, S. Limier, and L. Girin, "Autoencoders for music sound modeling: a comparison of linear, shallow, deep, recurrent and variational models," in *Proc. SMC*, 2019. [Online]. Available: http://www.gipsa-lab.fr/{~}fanny.roche/SMC{_}2019.html

[6] A. Bitton, P. Esling, and T. Harada, "Vector-Quantized Timbre Representation," *arXiv preprint arXiv:2007.06349*, 2020.

[7] Y. Wang, R. J. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio, Q. Le, Y. Agiomyrgiannakis, R. Clark, and R. A. Saurous, "Tacotron: Towards End-to-End Speech Synthesis," in *Proc. Interspeech*, 2017, pp. 4006–4010.

[8] X. Wang, S. Takaki, and J. Yamagishi, "Neural Source-Filter Waveform Models for Statistical Parametric Speech Synthesis," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 402–415, 2020. [Online]. Available: https://ieeexplore.ieee.org/document/8915761/

[9] H. Zen, A. Senior, and M. Schuster, "Statistical parametric speech synthesis using deep neural networks," in *Proc. ICASSP*, 2013, pp. 7962–7966.

[10] H. Zen and A. Senior, "Deep mixture density networks for acoustic modeling in statistical parametric speech synthesis," in *Proc. ICASSP*, 2014, pp. 3844–3848.

[11] Y. Fan, Y. Qian, F. Xie, and F. K. Soong, "TTS Synthesis with Bidirectional LSTM Based Recurrent Neural Networks," in *Proc. Interspeech*, 2014, pp. 1964–1968.

[12] A. Tamamori, T. Hayashi, K. Kobayashi, K. Takeda, and T. Toda, "Speaker-dependent WaveNet vocoder," in *Proc. Interspeech*, 2017, pp. 1118–1122.

[13] R. Prenger, R. Valle, and B. Catanzaro, "WaveGlow: A Flow-based Generative Network for Speech Synthesis," in *Proc. ICASSP*, 2019, pp. 3617–3621.

[14] H. Kawahara, I. Masuda-Katsuse, and A. de Cheveigne, "Restructuring speech representations using a pitch-adaptive time-frequency smoothing and an instantaneous-frequency-based F0 extraction: Possible role of a repetitive structure in sounds," *Speech Communication*, vol. 27, pp. 187–207, 1999.

[15] M. Morise, F. Yokomori, and K. Ozawa, "WORLD: A vocoder-based high-quality speech synthesis system for real-time applications," *IEICE Trans. on Information and Systems*, vol. 99, no. 7, pp. 1877–1884, 2016.

[16] N. Li, S. Liu, Y. Liu, S. Zhao, and M. Liu, "Neural speech synthesis with transformer network," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 6706–6713.

[17] Y. Yasuda, X. Wang, S. Takaki, and J. Yamagishi, "Investigation of enhanced Tacotron text-to-speech synthesis systems with self-attention for pitch accent language," in *Proc. ICASSP*. IEEE, 2019, pp. 6905–6909.

[18] B. Wang and Y.-H. Yang, "PerformanceNet: Score-to-audio music generation with multi-band convolutional residual network," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 1174–1181.

[19] C. Raffel and D. P. W. Ellis, "Intuitive analysis, creation and manipulation of midi data with pretty midi," in *Proc. ISMIR late breaking and demo papers*, 2014, pp. 84–93.

[20] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, 2014.

[21] D. Griffin and J. Lim, "Signal estimation from modified short-time Fourier transform," *IEEE Trans. ASSP*, no. 2, pp. 236–243, 1984.

[22] J. W. Kim, J. Salamon, P. Li, and J. P. Bello, "CREPE: A convolutional representation for pitch estimation," in *Proc. ICASSP*. IEEE, 2018, pp. 161–165.

[23] A. Chaigne and A. Askenfelt, "Numerical simulations of piano strings. I. A physical model for a struck string using finite difference methods," *The Journal of the Acoustical Society of America*, vol. 95, no. 2, pp. 1112–1118, 1994.

[24] N. Giordano and M. Jiang, "Physical Modeling of the Piano," *EURASIP Journal on Advances in Signal Processing*, vol. 2004, no. 7, p. 981942, dec 2004. [Online]. Available: https://asp-eurasipjournals.springeropen.com/articles/10.1155/S111086570440105X

[25] J. Chabassier and M. Duruflé, "Physical parameters for piano modeling," Ph.D. dissertation, INRIA, 2012.

11th ISCA Speech Synthesis Workshop (SSW 11)
26-28 August 2021, Budapest, Hungary

Preliminary study on using vector quantization latent spaces for TTS/VC systems with consistent performance

Hieu-Thi Luong[1], *Junichi Yamagishi*[1]

[1]National Institute of Informatics, Tokyo, Japan

{luonghieuthi,jyamagis}@nii.ac.jp

Abstract

Generally speaking, the main objective when training a neural speech synthesis system is to synthesize natural and expressive speech from the output layer of the neural network without much attention given to the hidden layers. However, by learning useful latent representation, the system can be used for many more practical scenarios. In this paper, we investigate the use of quantized vectors to model the latent linguistic embedding and compare it with the continuous counterpart. By enforcing different policies over the latent spaces in the training, we are able to obtain a latent linguistic embedding that takes on different properties while having a similar performance in terms of quality and speaker similarity. Our experiments show that the voice cloning system built with vector quantization has only a small degradation in terms of perceptive evaluations, but has a discrete latent space that is useful for reducing the representation bit-rate, which is desirable for data transferring, or limiting the information leaking, which is important for speaker anonymization and other tasks of that nature.

Index Terms: voice cloning, text-to-speech, voice conversion, vector quantization, variational autoencoder

1. Introduction

When it comes to text-to-speech (TTS) tasks, the deep learning approach has several advantages over the conventional approaches, such as its simple structure and the ability to scale with large data [1, 2]. These characteristics are key for pushing the performance of speech synthesis systems and machine learning systems in general. Recent works have shown that a sequence-to-sequence TTS system [1, 3] trained with a large transcribed speech corpus can synthesize speech with high naturalness directly from text input instead of going through several sub-systems. While such systems provide a high performance and a straightforward solution for TTS, many researchers have shifted their focus to more elaborate systems that aim to give some of the control back to human users [4, 5]. For example, Shen et al. [6] replaced the attention module with a duration prediction to create a more resilient sequence-to-sequence TTS model and enable the ability to control the spacing of generated speech, while Liu et al. [7] explicitly integrated emphatic code into the model so that users can control the emphasis by changing duration, intonation, and energy.

For voice conversion (VC), recent deep learning based systems are formulated around the ability to disentangle speaker and linguistic information of a neural network by using an information bottleneck structure to force the model to learn useful representations [8, 9, 3]. This information bottleneck structure can simply be a layer with a few units [10], a variational autoencoder (VAE) model with its encoder's output regularized to approximate a normal distribution [11, 12], or a jointly trained discrete latent space through vector quantization [13, 14, 9]. In these works, the common hypothesis is that using a certain network structure can help train a representation that takes on information and/or properties that are useful for the task at hand.

Previously, we proposed NAUTILUS [15], a versatile voice cloning system, that is a fusion of TTS and VC. By carefully designing the shared and the exclusive components, NAUTILUS can utilize them to perform elaborate tasks such as cloning unseen voices using untranscribed speech. It also has a consistent performance when switching between TTS and VC. These properties are the result of a unified and robust linguistic latent space achieved by the joint training and the VAE-like structure. However, one may want to use other methods to shape the latent space for many different purposes. Specifically, if we can train a discrete latent space [16, 17] instead of a continuous one, it will be useful for many applications. For example, a low bit-rate representation [13, 18] is ideal for a client-server VC system in which the speech encoder is stored in the client device while the speech decoder is not. Alternatively, by using the vector quantization bottleneck, we can limit the information getting through the speech encoder, which is important for tasks such as speaker anonymization [19, 20] as it helps reduce the leaking of speaker identity through temporal patterns. In this work, we investigate the use of vector quantization variational autoencoder (VQVAE) [17] components to model the linguistic latent space of the NAUTILUS system. In addition to conducting experiments to clarify its effect on subjective evaluations, we discuss how different types of assumption about the latent spaces are useful for different scenarios. We describe the basics of the voice cloning framework with the vector quantization components in Section 2 and discuss our motivation in Section 3. Section 4 lays out the experiment conditions, and Section 5 presents the subjective evaluation results. We conclude in Section 6 with a brief summary of our findings and mention of future works.

2. Vector Quantization Latent Space for Voice Cloning

We adopt the basic concepts of the voice cloning framework proposed in our previous publications [15] and replace the VAE-based encoders with a VQVAE-based counterpart for this work. Readers may want to refer to the original study [15] for more context. Briefly, our voice cloning system is a unified system of TTS and VC, and thanks to this fusion it has the capacity to clone new voices using untranscribed speech. The proposed VQVAE-based system, called NAUTILUS-VQ, is illustrated in Fig. 1. The only difference from the original [15] is the way the text and speech encoders are set up as shown in Fig. 2. More specifically, the vector quantization bottleneck transforms the continuous latent feature z, emitted by the text or speech encoder, into a discrete latent feature q using the jointly trained codebook $\boldsymbol{e}_k, k \in 1...K$, with $q = e_k$ where $k = argmin_j ||z - e_j||$. The speech decoder then consumes q,

10.21437/SSW.2021-24

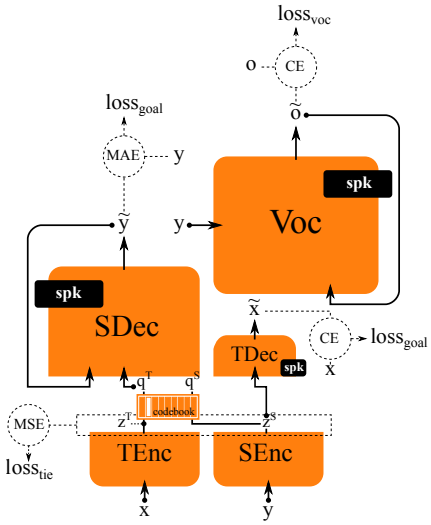


Figure 1: *The modified NAUTILUS-VQ system has a similar structure to the original system, which includes a text encoder ($TEnc$), a speech encoder ($SEnc$), a text decoder ($TDec$), a speech decoder ($SDec$), and a neural vocoder (Voc). The jointly trained codebook is the new addition.* $\boldsymbol{x}$ *is phoneme,* $\boldsymbol{y}$ *is acoustic,* $\boldsymbol{o}$ *is waveform,* $\boldsymbol{z}$ *is continuous latent feature and* $\boldsymbol{q}$ *is the quantized latent feature. The term* loss_{goal} *is a placeholder, depending on the encoder/decoder combination — it can be* loss_{tts} *(Text-to-speech: $TEnc$→codebook→$SDec$),* loss_{sts} *(Speech-to-speech, STS: $SEnc$→codebook→$SDec$),* loss_{stt} *(Speech-to-text, STT: $SEnc$→$TDec$), or* loss_{ttt} *(Text-to-text: $TEnc$→$TDec$) . The loss functions used in training and adaptation are mean absolute error (MAE), mean squared error (MSE) and cross entropy (CE).*

instead of z, to reconstruct the acoustic feature y that is used to synthesize speech waveform o.

2.1. Train the initial model

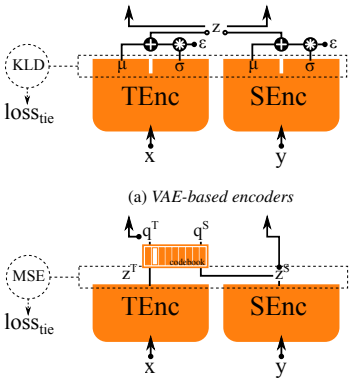
First, we need to jointly train the text/speech encoders/decoders and the codebook in a supervised fashion by using a large-scale transcribed multi-speaker speech corpus and optimizing a designated loss:

$$\text{loss}_{train} = \text{loss}^{tts}_{train} + \alpha_{sts}\ \text{loss}^{sts}_{train} + \alpha_{stt}\ \text{loss}_{stt} + \beta\ \text{loss}_{tie}\ . \quad (1)$$

Please see the caption of Fig. 1 for subscripts of each term. The basic structure of the training loss is not much different from the original [15], but, due to the vector quantization components, the details are a little more complex. Specifically $\text{loss}^{tts}_{train}$ is similar to a typical VQVAE setup [17]:

$$\text{loss}^{tts}_{train} = \text{loss}_{tts} + \delta_{VQ}\ \text{loss}^{T}_{VQ} + \delta_{C}\ \text{loss}^{T}_{C}\ . \quad (2)$$

where $\text{loss}_{tts} = ||\tilde{y}^T - y||$ is the reconstruction loss of the acoustic feature, $\text{loss}^{T}_{VQ} = ||sg(z^T) - q^T||^2_2$ is the codebook training loss in response to the text input, x, from the text encoder, and $\text{loss}^{T}_{C} = ||z^T - sg(q^T)||^2_2$ is the text encoder commitment. The operator $sg()$ indicates the stop gradient operation.



(a) *VAE-based encoders*

(b) *VQVAE-based encoders*

Figure 2: *The VAE (original) and VQVAE (proposed) setups for the encoders. The VAE-based encoders output mean* $\boldsymbol{\mu}$ *and standard deviation* $\boldsymbol{\sigma}$ *and then sample the latent feature* $\boldsymbol{z}$ *using an* $\boldsymbol{\epsilon}$ *value drawn from a normal distribution. Kullback-Leibler divergence (KLD) is used as* loss_{tie} *in this setup. The VQVAE-based encoders output continuous latent feature* $\boldsymbol{z}$ *that is then quantized into the discrete feature q by using the jointly trained codebook. MSE is used for* loss_{tie} *in this case.*

Similarly, we have the optimization loss for the STS stack that handles the speech input:

$$\text{loss}^{sts}_{train} = \text{loss}_{sts} + \delta_{VQ}\ \text{loss}^{S}_{VQ} + \delta_{C}\ \text{loss}^{S}_{C}\ . \quad (3)$$

Unlike the VAE-based setup [15] which used Kullback-Leibler (KL) divergence to "tie" the encoders' outputs, the VQVAE-based system uses MSE as the latent tying loss:

$$\text{loss}_{tie} = ||sg(z^T) - z^S||^2_2\ . \quad (4)$$

Note that we stop the gradient on the text-encoded latent feature, which basically creates an asymmetric tied-layer loss instead of the symmetric KL divergence function as in the original [15]. A multi-speaker WaveNet vocoder, unchanged from the original, is separately initialized using the same corpus:

$$\text{loss}'_{train} = \text{loss}_{voc}\ , \quad (5)$$

The speech decoder, text decoder, and neural vocoder contain speaker dependent (SD) components, which are just one-hot vectors representing speakers in the training set. These SD components will be removed in the voice cloning steps along with the text decoder, which is only included as an auxiliary phone classification regularizer [15].

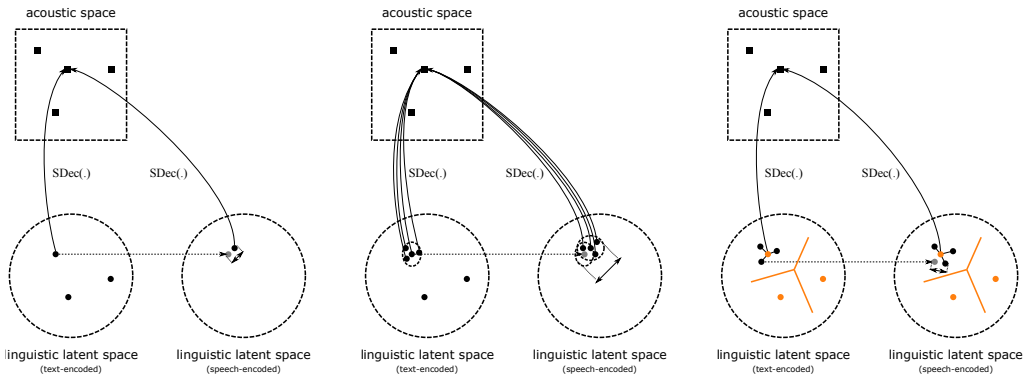
2.2. Clone the target voice

Given the untranscribed speech of an unseen speaker, we adapt the initial model to generate speech with the voice of the new target, using either the TTS or VC interface.

2.2.1. Step 1 - Adaptation

After removing all SD components, we use the STS stack ($SEnc$→codebook→$SDec$) to fine-tune the speech decoder while keep other modules immutable:

$$\text{loss}_{adapt} = \text{loss}_{sts}\ , \quad (6)$$



(a) *Standard* (b) *VAE-based* (c) *VQVAE-based*

Figure 3: *Different ways the linguistic latent spaces are setup and the methods used to enforce the consistency between them.*

The neural vocoder is also adapted using the same strategy (removing SD components, fine-tuning the rest):

$$\text{loss}'_{adapt} = \text{loss}_{voc} \ . \tag{7}$$

As this step is similar to the original framework, reader can refer to Fig. 2a in [15] for details.

2.2.2. *Step 2 - Welding*

The adapted model obtained in step one is already capable of generating speech with the target voice. However, we want to increase the speech decoder and the neural vocoder compatibility by jointly tuning them using the speech to waveform stack ($SEnc$→codebook→$SDec$→Voc):

$$\text{loss}_{weld} = \text{loss}_{sts} + \gamma \ \text{loss}_{voc} \ . \tag{8}$$

The loss_{sts} is included to maintain the acoustic space for the autoregressive speech decoder (see Fig. 2b in [15]).

2.2.3. *Step 3 - Inference*

After the previous steps, the adapted model can generate speech with the voice of the target. The TTS stack and the neural vocoder form a TTS interface that transforms phoneme sequences into a speech waveform, while the STS stack and the neural vocoder form a VC interface that converts utterances spoken by arbitrary source speakers into speech of the same content but with the voice of the target (Fig. 2c in [15]).

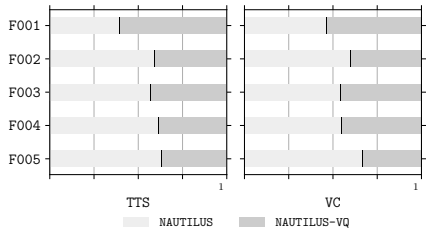
3. Shaping the linguistic latent spaces

Our voice cloning system functions on the assumption of a robust and consistent linguistic latent space. Briefly, we want the latent linguistic embedding (LLE) to contain linguistic information, which is useful for speech reconstruction, but none of the speaker information. Moreover, the consistency between the TTS and VC interfaces is dictated by the consistency between the text-encoded and speech-encoded latent spaces. In other words, the proposed system achieves a perfect consistency if its text and speech encoders produce an identical LLE sequence given a sentence input and a spoken utterance of the same content. Note that, for many practical applications [21], it may not even be desirable to achieve such consistency as it eliminates the ability to synthesize speech content that cannot be represented in written form. However, establishing a straightforward goal about consistency helps to simplify the analysis.

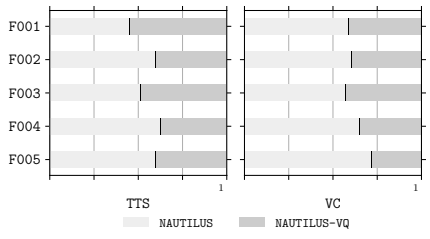
Table 1: *Japanese target speakers for voice cloning task*

Speaker	Gender	Quantity	Duration
F001	female	483 utt.	45.0 min
F002	female	481 utt.	44.4 min
F003	female	484 utt.	47.4 min
F004	female	468 utt.	40.8 min
F005	female	485 utt.	47.6 min
XL10	female	10 utt.	55 s
		125 utt.	10.9 min
		500 utt.	44.5 min
		2000 utt.	2.9 h
		8750 utt.	12.9 h

We can simply use standard latent features [22] as LLEs and assume the text and speech encoders produce identical features when consuming different modalities of the same content, and so we optimize the consistency between them by minimizing the distance between the two latent points [23] as shown in Fig. 3a. The choice of distance function is another decision that could affect the performance [24], but in practice most use Euclidean distance for its simplicity [25]. The flaws of this assumption are the one-to-many relation of text and speech and the scarcity nature of data, which make it difficult to train a robust and consistent latent space. To address this problem, we utilized the VAE-based encoders (Fig. 3b) in our previous works [15]. This modification provides two key benefits: 1) the speech decoder is trained in a denoising fashion due to the sampling process, which can be interpreted as an artificial data argumentation, and 2) we can use a density-wise instead of a point-wise function to connect the text and speech encoders which helps with the consistency. However, it has the common drawback of the VAE model [26] which is the average-ness of the generated features [3]. Therefore, in this paper, we investigate the VQVAE-based modification that has discrete latent spaces, as shown in Fig. 3c. The hypothesis is that the discrete features will allow the speech decoder to learn fine-grain details. Moreover, it has several useful traits as mentioned in previous sections.

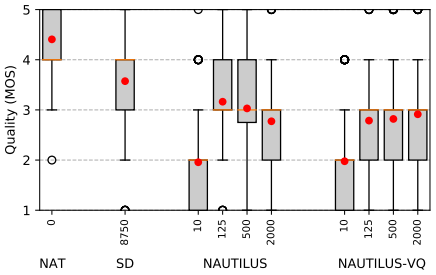


(a) *Quality*

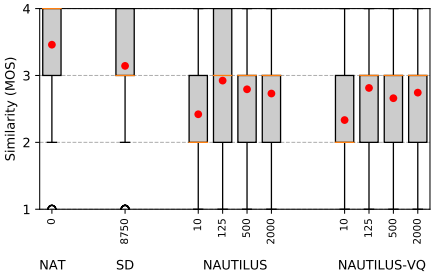


(b) *Similarity*

Figure 4: *Subjective evaluations on quality and speaker similarity between NAUTILUS and NAUTILUS-VQ on voice cloning task for TTS and VC.*



(a) *Quality*



(b) *Similarity*

Figure 5: *Subjective evaluations on quality and speaker similarity of the TTS unsupervised speaker adaption using varying amount of untranscribed speech data of speaker XL10.*

4. Experiments

4.1. Model and training configurations

For the experiments, we compared the performances of the original NAUTILUS system [15] and the new system with vector quantization modification. The network architecture of the original was unchanged from the previous publication, and consists of many layers of causal and non-causal dilated convolution layers. Readers can refer to Fig. 4 and Sec. IV of [15] for details. The NAUTILUS-VQ system also adopts this architecture but with several changes to reflect the vector quantization components. Specifically, the encoders directly output 64-dimensional latent vectors instead of means and standard variances. The 160-code jointly trained codebook was used to transform these continuous features into discrete ones. We chose this size for the codebook because it produces a relatively reliable performance based on several test-runs and relevant publications pertaining to VQVAE [14, 27]. For the hyperparameters, we set $\alpha = 0.1$, $\beta = 0.25$, $\gamma = 0.01$, the same as in [15], and $\delta_C = 1.0$, $\delta_{VQ} = 0.25$, as based on relevant works [17, 14].

4.2. Speech data

Previously, we conducted experiments with English as the target language [15]. In this work, we used Japanese to test our methodology under a new condition. Specifically, several native female Japanese speakers, as listed in Table 1, were selected as the target speakers. The Japanese model was first initialized on a large-scale low-quality transcribed speech corpus with a diverse linguistic content. We used ∼236 hours of speech (978 speakers) from the 16 khz Corpus of Spontaneous Japanese (CSJ) [28] for this purpose. Then, we fine-tuned it on a quality-controlled transcribed speech corpus for the desired sampling rate. We used ∼134 hours of speech (235 speakers) from an in-house 24 khz Japanese Voice Bank Corpus for this step. The same policy was applied for the training of both NAUTILUS and NAUTILUS-VQ, and it is similar to the one we used when we trained the English models [15]. The well-trained modules were used to adapt to the target speakers using their untranscribed speech data. For the first scenario, five speakers with about 45 minutes of speech were used to compare the performances of NAUTILUS and NAUTILUS-VQ. For the second scenario, a different speaker, XL10, was used to investigate the performances when adapting with different amounts of data. This speaker was chosen because she was included in our previous experiments using the conventional TTS system [29].

5. Evaluations

5.1. Vector quantization latent spaces for voice cloning

We first compare the original and the VQ-based systems on the voice cloning task: specifically their perceptive evaluations for TTS and VC[1]. We used about 45 minutes of untranscribed speech of the first five target speakers in Table 1 for this scenario. This amount of data was more than our previous English experiments [15], which were conducted with just five or ten minutes of speech. We used a crowdsourcing service to conduct the survey. A total of 241 native speakers, each of whom did one to five sessions, participated. Listeners were asked to pick the preferred sample between two presented in terms of either quality or speaker similarity (which includes a reference sample). The results are shown in Fig. 4. In total, each speaker/system/task was judged 300 times. Interestingly, the NAUTILUS-VQ system had worse results for most speakers except F001. As most of the differences between the two

[1]Samples are available at `https://nii-yamagishilab.github.io/sample-preliminary-nautilus-vq/`

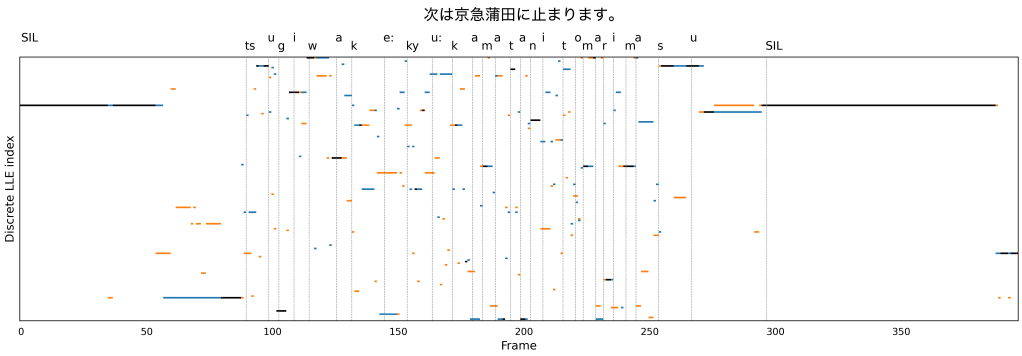


Figure 6: *Examples of the 160-code discrete LLE sequences. One utterance of a source speaker was used to generate the speech-encoded LLE (orange), while text (phoneme) and alignment information extracted from the same utterance was used to generate the text-encoded LLE (blue). The color black indicates the overlap of the speech-encoded and text-encoded LLE sequences, which covers 54.41% of this particular example utterance.*

systems were marginal, we conclude that the proposed vector quantization latent space can be used for voice cloning but has a small degradation in quality.

5.2. TTS speaker adaptation with different amount of untranscribed speech data

Next, we investigate the performances of both NAUTILUS and NAUTILUS-VQ for TTS unsupervised speaker adaptation with varying amount of adaptation data. For this scenario, we use speaker XL10, with whom we have more than 13 hours of speech data. Previously, we found that with this amount of data, the SD system of this speaker does not seem to benefit from the joint training with augmented data from other speakers [29]. In addition to the natural utterances (NAT) and generated utterances from our voice cloning systems, we included generated utterance from the conventional TTS system trained on 12.9 hours of transcribed speech of XL10. Basically, we reused the SD system in [29] as the upper bound for this experiment. The same listeners who participated in the first scenario survey were asked to do this one as well. Specifically, they were asked to judge the naturalness of a speech sample in a typical 5-point scale mean opinion score (MOS) question, and the likelihood that two presented samples were spoken by the same person on a 4-point scale. The results are shown in Fig. 5. As expected, none of the TTS systems were as good as natural speech, but the speaker similarity was not too far behind. Second, none of our voice cloning systems were as good as the SD baseline, which is not surprising as SD was trained on 12.9 hours of transcribed speech [29] while our systems cloned voices with only a small amount of untranscribed utterances. Third, between our systems, NAUTILUS-VQ has worse results than NAUTILUS in most data points, but not significantly so. Finally, the most interesting results were the performances when adapting to varying amounts of data, where we found that the performance of the NAUTILUS system seemed to peak at 125 utterances. These findings demonstrate the potential of our voice cloning system but at the same time reveal its remaining limitations.

5.3. Visualization of the discrete LLE sequences

One advantage of our speech synthesis is its ability to use as TTS and VC while maintaining a relatively consistent performance when switching between the two. Figure 6 shows the LLE sequences generated from text input and an utterance spoken by a source speaker not included in the training and adaptation stages. As the discrete LLE is forced to be one of 160 jointly trained vectors, it is easier and more intuitive to evaluate the consistency between the text-encoded and speech-encoded LLE sequences compared with the continuous representation (Fig. 8 in [15]). Figure 6 shows the discrete LLE sequences generated by the text and speech encoders using text and speech of the same content. For a perfectly consistent TTS/VC system, we expect the two sequences to be perfectly matched but it was not the case as seen in Fig. 6. Most of the overlap occurred at the start and the end of the utterance, which is the silence phoneme, but not much elsewhere. Overall, the LLE sequences were sparse and fragmented. For further improvement, focusing on condensing the latent space and stabilizing the text-encoded and speech-encoded LLE sequences would be a good direction.

6. Conclusion

In this paper, we investigated the feasibility of using VQVAE-based components to train a discrete latent linguistic embedding for a consistent performance TTS/VC system. While the perceptive evaluations showed that the proposed NAUTILUS-VQ system is not as good as the original system, having these different approaches to model the linguistic latent spaces is handy for many practical reasons. Understanding the dynamics of different methods is also important for the development of a more sophisticated speech synthesis system that can solve more complex and elaborate tasks, such as controlling speaking style [5], denoising TTS [30], or generating audio other than speech [31]. As VQVAE is just one way to model a jointly trained discrete latent space, other methods [16, 32] or assumptions [14, 33] about the nature of the latent space may lead to different results and have different utilities for specific application scenarios.

7. Acknowledgements

This work was partially supported by a JST CREST Grant (JPMJCR18A6, VoicePersonae project), Japan, MEXT KAKENHI Grants (18H04112, 21H04906), Japan, and KDDI research, Japan.

8. References

[1] W. Ping, K. Peng, A. Gibiansky, S. O. Arik, A. Kannan, S. Narang, J. Raiman, and J. Miller, "Deep voice 3: Scaling text-to-speech with convolutional sequence learning," in *Proc. ICLR*, 2018, pp. 1–11.

[2] J. Sotelo, S. Mehri, K. Kumar, J. F. Santos, K. Kastner, A. Courville, and Y. Bengio, "Char2wav: End-to-end speech synthesis," in *Proc. ICLR*, 2017, pp. 1–6.

[3] G. Sun, Y. Zhang, R. J. Weiss, Y. Cao, H. Zen, A. Rosenberg, B. Ramabhadran, and Y. Wu, "Generating diverse and natural text-to-speech samples using a quantized fine-grained vae and autoregressive prosody prior," in *Proc. ICASSP*, 2020, pp. 6699–6703.

[4] H.-T. Luong, S. Takaki, G. E. Henter, and J. Yamagishi, "Adapting and controlling dnn-based speech synthesis using input codes," in *Proc. ICASSP*, 2017, pp. 4905–4909.

[5] Y. Wang, D. Stanton, Y. Zhang, R.-S. Ryan, E. Battenberg, J. Shor, Y. Xiao, Y. Jia, F. Ren, and R. A. Saurous, "Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis," in *Proc. ICML*, 2018, pp. 5180–5189.

[6] J. Shen, Y. Jia, M. Chrzanowski, Y. Zhang, I. Elias, H. Zen, and Y. Wu, "Non-attentive tacotron: Robust and controllable neural tts synthesis including unsupervised duration modeling," *arXiv preprint arXiv:2010.04301*, 2020.

[7] L. Liu, J. Hu, Z. Wu, S. Yang, S. Yang, J. Jia, and H. Meng, "Controllable emphatic speech synthesis based on forward attention for expressive speech synthesis," in *Proc. SLT*. IEEE, 2021, pp. 410–414.

[8] C.-C. Hsu, H.-T. Hwang, Y.-C. Wu, Y. Tsao, and H.-M. Wang, "Voice conversion from non-parallel corpora using variational auto-encoder," in *Proc. APSIPA*, 2016, pp. 1–6.

[9] D.-Y. Wu and H.-y. Lee, "One-shot voice conversion by vector quantization," in *Proc. ICASSP*, 2020, pp. 7734–7738.

[10] K. Qian, Y. Zhang, S. Chang, X. Yang, and M. Hasegawa-Johnson, "Autovc: Zero-shot voice style transfer with only autoencoder loss," in *Proc. ICML*, 2019, pp. 5210–5219.

[11] C.-C. Hsu, H.-T. Hwang, Y.-C. Wu, Y. Tsao, and H.-M. Wang, "Voice conversion from unaligned corpora using variational autoencoding wasserstein generative adversarial networks," in *Proc. INTERSPEECH*, 2017, pp. 3364–3368.

[12] H. Kameoka, T. Kaneko, K. Tanaka, and N. Hojo, "Acvae-vc: Non-parallel voice conversion with auxiliary classifier variational autoencoder," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 27, no. 9, pp. 1432–1443, 2019.

[13] A. Tjandra, B. Sisman, M. Zhang, S. Sakti, H. Li, and S. Nakamura, "Vqvae unsupervised unit discovery and multi-scale code2spec inverter for zerospeech challenge 2019," *Proc. INTERSPEECH*, pp. 1118–1122, 2019.

[14] S. Ding and R. Gutierrez-Osuna, "Group latent embedding for vector quantized variational autoencoder in non-parallel voice conversion." in *Proc. INTERSPEECH*, 2019, pp. 724–728.

[15] H.-T. Luong and J. Yamagishi, "Nautilus: a versatile voice cloning system," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 28, pp. 2967–2981, 2020.

[16] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with gumbel-softmax," in *Proc. ICLR*, 2016, pp. 1–12.

[17] A. van den Oord, O. Vinyals, and K. Kavukcuoglu, "Neural discrete representation learning," in *Proc. NIPS*, 2017, pp. 1–10.

[18] E. Dunbar, R. Algayres, J. Karadayi, M. Bernard, J. Benjumea, X.-N. Cao, L. Miskic, C. Dugrain, L. Ondel, A. W. Black *et al.*, "The zero resource speech challenge 2019: Tts without t," *Proc. INTERSPEECH*, pp. 1088–1092, 2019.

[19] F. Fang, X. Wang, J. Yamagishi, I. Echizen, M. Todisco, N. Evans, and J.-F. Bonastre, "Speaker anonymization using x-vector and neural waveform models," in *Proc. SSW*, 2019, pp. 155–160.

[20] N. Tomashenko, B. M. L. Srivastava, X. Wang, E. Vincent, A. Nautsch, J. Yamagishi, N. Evans, J. Patino, J.-F. Bonastre, P.-G. Noé *et al.*, "Introducing the voiceprivacy initiative," in *Proc. INTERSPEECH*, 2020, pp. 1693–1697.

[21] H.-T. Luong and J. Yamagishi, "Latent linguistic embedding for cross-lingual text-to-speech and voice conversion," in *Proc. Joint Workshop for the Blizzard Challenge and Voice Conversion Challenge 2020*, 2020, pp. 150–154.

[22] W.-C. Huang, T. Hayashi, Y.-C. Wu, H. Kameoka, and T. Toda, "Pretraining techniques for sequence-to-sequence voice conversion," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 29, pp. 745–755, 2021.

[23] H.-T. Luong and J. Yamagishi, "Multimodal speech synthesis architecture for unsupervised speaker adaptation," in *Proc. INTERSPEECH*, 2018, pp. 2494–2498.

[24] S. Karita, S. Watanabe, T. Iwata, M. Delcroix, A. Ogawa, and T. Nakatani, "Semi-supervised end-to-end speech recognition using text-to-speech and autoencoders," in *Proc. ICASSP*, 2019, pp. 6166–6170.

[25] M. Zhang, Y. Zhou, L. Zhao, and H. Li, "Transfer learning from speech synthesis to voice conversion with non-parallel training data," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 29, pp. 1290–1302, 2021.

[26] D. P. Kingma, D. J. Rezende, S. Mohamed, and M. Welling, "Semi-supervised learning with deep generative models," in *Proc. NIPS*, 2014, pp. 3581–3589.

[27] J. Williams, Y. Zhao, E. Cooper, and J. Yamagishi, "Learning disentangled phone and speaker representations in a semi-supervised vq-vae paradigm," in *Proc. ICASSP*, 2021, pp. 7053–7057.

[28] K. Maekawa, H. Koiso, S. Furui, and H. Isahara, "Spontaneous speech corpus of japanese." in *Proc. LREC*, vol. 6, 2000, pp. 1–5.

[29] H.-T. Luong, X. Wang, J. Yamagishi, and N. Nishizawa, "Training multi-speaker neural text-to-speech systems using speaker-imbalanced speech corpora," in *Proc. INTERSPEECH*, 2019, pp. 1303–1307.

[30] C. Zhang, Y. Ren, X. Tan, J. Liu, K. Zhang, T. Qin, S. Zhao, and T.-Y. Liu, "Denoispeech: Denoising text to speech with frame-level noise modeling," in *Proc. ICASSP*, 2021, pp. 7063–7067.

[31] J. Engel, K. K. Agrawal, S. Chen, I. Gulrajani, C. Donahue, and A. Roberts, "Gansynth: Adversarial neural audio synthesis," in *Proc. ICLR*, 2019, pp. 1–17.

[32] C. Louizos, M. Welling, and D. P. Kingma, "Learning sparse neural networks through l_0 regularization," in *Proc. ICLR*, 2018, pp. 1–13.

[33] T. V. Ho and M. Akagi, "Non-parallel voice conversion based on hierarchical latent embedding vector quantized variational autoencoder," in *Proc. Joint Workshop for the Blizzard Challenge and Voice Conversion Challenge 2020*, 2020, pp. 140–144.

11th ISCA Speech Synthesis Workshop (SSW 11)
26-28 August 2021, Budapest, Hungary

Low-latency real-time non-parallel voice conversion based on cyclic variational autoencoder and multiband WaveRNN with data-driven linear prediction

Patrick Lumban Tobing[1], *Tomoki Toda*[1]

[1]Nagoya University, Japan

patrick.lumbantobing@g.sp.m.is.nagoya-u.ac.jp, tomoki@icts.nagoya-u.ac.jp

Abstract

This paper presents a low-latency real-time (LLRT) non-parallel voice conversion (VC) framework based on cyclic variational autoencoder (CycleVAE) and multiband WaveRNN with data-driven linear prediction (MWDLP). CycleVAE is a robust non-parallel multispeaker spectral model, which utilizes a speaker-independent latent space and a speaker-dependent code to generate reconstructed/converted spectral features given the spectral features of an input speaker. On the other hand, MWDLP is an efficient and a high-quality neural vocoder that can handle multispeaker data and generate speech waveform for LLRT applications with CPU. To accommodate LLRT constraint with CPU, we propose a novel CycleVAE framework that utilizes mel-spectrogram as spectral features and is built with a sparse network architecture. Further, to improve the modeling performance, we also propose a novel fine-tuning procedure that refines the frame-rate CycleVAE network by utilizing the waveform loss from the MWDLP network. The experimental results demonstrate that the proposed framework achieves high-performance VC, while allowing for LLRT usage with a single-core of 2.1–2.7 GHz CPU on a real-time factor of 0.87–0.95, including input/output, feature extraction, on a frame shift of 10 ms, a window length of 27.5 ms, and 2 lookup frames.

Index Terms: non-parallel voice conversion, low-latency real-time, CycleVAE, multiband WaveRNN, waveform loss

1. Introduction

Voice conversion (VC) [1] is a technique for altering voice characteristics of a speech waveform from an input speaker to that of a desired target speaker while preserving the lingustic contents of the speech. Many real-world and/or research applications benefit from VC, such as for speech database augmentation [2], for recovery of impaired speech [3], for expressive speech synthesis [4], for singing voice [5], for body-conducted speech processing [6], and for speaker verification [7]. As the development of VC has been growing rapidly [8], it is also wise to pursue not only for the highest performance, but also for its feasibility on the constraints of real-world deployment/development, e.g., low-latency real-time (LLRT) [9] constraint with low-computational machines in its deployment and unavailability of parallel (paired) data between source and target speakers in its development.

To develop LLRT VC [9], the costs from input waveform analysis, conversion step, and output waveform generation are taken into account to obtain the acceptable amount of total delay. On the waveform analysis, several works use simple fast Fourier transform (FFT) [9, 10, 11, 12]. On the conversion module, where the spectral characteristics of speech waveform are usually modeled, a Gaussian mixture model is employed in [9], a simple multi layer perceptron is employed in [11, 12], while convolutional neural network (CNN) and recurrent neural network (RNN) are employed in [10]. On the waveform generation, source-filter vocoder based on STRAIGHT [13] is used in [9, 10], while WORLD [14] is used in [11], and direct waveform filtering is utilized in [5]. In all cases, parallel training data is required to develop the conversion model, while the quality of the waveform generation module is still limited. In this paper, we work to achieve flexible and high-quality LLRT VC, where it can be developed with non-parallel data and provide high-quality waveform using also neural network for waveform generation, i.e., neural vocoder.

Neural vocoder could provide high-quality speech waveform in copy-synthesis [15], in text-to-speech (TTS) [16], and in VC [8] systems, albeit, high computational cost impedes most of its use on LLRT applications. Essentially, neural vocoder architectures can be categorized into autoregressive (AR) [17, 18] and non-autoregressive (non-AR) [19, 20] models, on which the former depends on the previously generated waveform samples. In practice, AR models based on RNN (WaveRNN) [17, 18] can be developed with less layers than non-AR ones, which are built with multiple layers (deep) of CNN. In LLRT applications, where waveform synthesis is sequentially performed depending on the availability of input stream, it is more difficult for the deeper non-AR models to achieve this constraint while still preserving high-quality waveform. In this work, to reliably achieve LLRT VC, we utilize a high-quality AR model called multiband WaveRNN with data-driven linear prediction (MWDLP) [21], which has been proven to be capable of producing high-fidelity waveform in the most adverse conditions including on LLRT constraint.

On the other hand, to develop non-parallel VC, a shared space between speakers (speaker-independent) can be utilized as a reference point on which the linguistic contents of speech are generated. For instance, several works have employed the use of explicit text/phonetic space [22, 23]. An alternative way is to employ a linguistically unsupervised latent space that serves as a point of distribution for the content generation, such as in variational autoencoder (VAE) [24, 25] or generative adversarial network [26]. The unsupervised approach has more flexibility in terms of independency from linguistic features in its development, which could be of higher value in situations where reliable transcriptions are difficult to be obtained. In this work, we focus on the use of a robust model based on VAE called cyclic variational autoencoder (CycleVAE) [27] that is capable of handling non-parallel multispeaker data.

To achieve flexible and high-quality LLRT VC, we propose to combine CycleVAE-based spectral model and MWDLP-based neural vocoder. First, we propose to modify the spectral features of CycleVAE to be that of mel-spectrogram. Second, as in [17, 18, 21], we propose to employ sparsification for the CycleVAE network. Finally, to achieve high-performance VC, we propose a novel fine-tuning for the CycleVAE model with the use of waveform domain loss from the MWDLP.

10.21437/SSW.2021-25

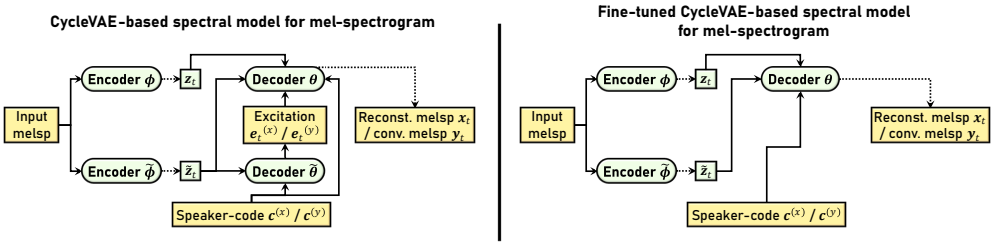


Figure 1: *Diagram of proposed CycleVAE model for mel-spectrogram (melsp) spectral features (left) with its fine-tuned architecture (right), where the second decoder $\tilde{\boldsymbol{\theta}}$ (excitation) is discarded, while keeping the related second encoder $\tilde{\boldsymbol{\phi}}$; Dotted lines denote sampling; Latent features are sampled from estimated posteriors; Reconstructed (reconst.) / converted (conv.) mel-spectrogram is sampled with estimated Gaussian parameters; Paths for speaker classifier (variational posterior of speaker-code) are omitted for brevity.*

2. MWDLP-based neural vocoder

Let $\boldsymbol{s} = [s_1, \ldots, s_{t_s}, \ldots, s_{T_s}]^\top$ be the sequence of speech waveform samples, where t_s and T_s respectively denotes the time indices and the length of the waveform samples. At band-level, the sequence of speech waveform samples is denoted as $\boldsymbol{s}^{(m)} = [s_1^{(m)}, \ldots, s_\tau^{(m)}, \ldots, s_\mathcal{T}^{(m)}]^\top$, where m denotes the mth band index, τ denotes the band-level time index, $\mathcal{T} = T_s/M$ denotes the length of the band-level waveform samples, which is downsampled from T_s by a factor of M [28], and the total number of bands is denoted as M. At frame-level, the sequence of conditioning feature vectors is denoted as $\boldsymbol{x} = [\boldsymbol{x}_1^\top, \ldots, \boldsymbol{x}_t^\top, \ldots, \boldsymbol{x}_T^\top]^\top$, where T denotes the length of the frame-level conditioning feature vector sequence, and at band-level, the sequence of conditioning feature vectors is denoted as $\tilde{\boldsymbol{x}} = [\tilde{\boldsymbol{x}}_1, \ldots, \tilde{\boldsymbol{x}}_\tau, \ldots, \tilde{\boldsymbol{x}}_\mathcal{T}]$.

In MWDLP [21], the likelihood of the sequence of waveform samples $\boldsymbol{s}$ is defined by the probability mass function (p.m.f.) of the discrete waveform samples as follows:

$$p(\boldsymbol{s}) = \prod_{m=1}^{M}\prod_{\tau=1}^{\mathcal{T}} p(s_\tau^{(m)}|\boldsymbol{s}_{1:\tau-1}^{(M)}, \tilde{\boldsymbol{x}}_\tau) = \prod_{m=1}^{M}\prod_{\tau=1}^{\mathcal{T}} \boldsymbol{p}_\tau^{(m)\top}\boldsymbol{v}_\tau^{(m)}, \quad (1)$$

where $\boldsymbol{s}_{1:\tau-1}^{(M)}$ denotes the past samples of all band-levels waveform, $\boldsymbol{p}_\tau^{(m)} = [p_\tau^{(m)}[1], \ldots, p_\tau^{(m)}[b], \ldots, p_\tau^{(m)}[B]]^\top$denotes the probability vector, the number of sample bins is denoted as B, and $\boldsymbol{v}_\tau^{(m)}$ denotes a one-hot vector. Of the probability vector $\boldsymbol{p}_\tau^{(m)}$, the probability of each sample bin $p_\tau^{(m)}[b]$ is given by

$$p_\tau^{(m)}[b] = \frac{\exp(\hat{o}_\tau^{(m)}[b])}{\sum_{j=1}^{B}\exp(\hat{o}_\tau^{(m)}[j])}, \quad (2)$$

where $\exp(\cdot)$ denotes the exponential function, $\hat{o}_\tau^{(m)}[b]$ is the unnormalized probability (logit) of the bth sample bin for the mth band, and the vector of logits is denoted as $\hat{\boldsymbol{o}}_\tau^{(m)} = [\hat{o}_\tau^{(m)}[1], \ldots, \hat{o}_\tau^{(m)}[b], \ldots, \hat{o}_\tau^{(m)}[B]]^\top$.

The linear prediction (LP) [29] is performed in the logit space of the discrete waveform samples as follows:

$$\hat{\boldsymbol{o}}_\tau^{(m)} = \sum_{k=1}^{K} a_\tau^{(m)}[k]\boldsymbol{r}_{\tau-k}^{(m)} + \boldsymbol{o}_\tau^{(m)}, \quad (3)$$

where the residual logit vector is denoted as $\boldsymbol{o}_\tau^{(m)}$, the kth data-driven LP coefficient of the mth band is denoted as $a_\tau^{(m)}[k]$, k denotes the index of LP coefficient, and the total number of coefficients is denoted as K. $\{\boldsymbol{r}_{\tau-1}^{(m)}, \ldots, \boldsymbol{r}_{\tau-K}^{(m)}\}$ are the trainable logit basis vectors corresponding to past K discrete samples. In Eq. 3, the network outputs are $a_\tau^{(m)}[k]$ and $\boldsymbol{o}_\tau^{(m)}$.

3. Proposed LLRT VC based on CycleVAE spectral model and MWDLP

3.1. CycleVAE model with mel-spectrogram features

To realize LLRT VC, in this work, we propose to use mel-spectrogram as the spectral features for CycleVAE model, where we extend the CycleVAE [25, 27] to incorporate estimation of intermediate excitation features, e.g., fundamental frequency (F0). Diagram of the proposed model is illustrated in the left side of Fig. 1.

Let $\boldsymbol{x}_t = [x_1[1], \ldots, x_t[d], \ldots, x_t[D]]^\top$ and $\boldsymbol{y}_t = [y_1[1], \ldots, y_t[d], \ldots, y_t[D]]^\top$ be the D-dimensional spectral feature vectors of an input speaker x and that of a converted speaker y at time t, respectively. The likelihood function of the input spectral feature vector $\boldsymbol{x}_t$ is defined as follows:

$$p_{\boldsymbol{\theta},\tilde{\boldsymbol{\theta}}}(\boldsymbol{x}_t, \boldsymbol{e}_t^{(x)}|\boldsymbol{c}_t^{(x)}) = \iint p_{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{z}_t, \tilde{\boldsymbol{z}}_t, \boldsymbol{c}_t^{(x)}, \boldsymbol{e}_t^{(x)}) p_{\tilde{\boldsymbol{\theta}}}(\boldsymbol{e}_t^{(x)}|\tilde{\boldsymbol{z}}_t, \boldsymbol{c}_t^{(x)}) p_{\boldsymbol{\theta}}(\boldsymbol{z}_t) p_{\tilde{\boldsymbol{\theta}}}(\tilde{\boldsymbol{z}}_t)\mathrm{d}\tilde{\boldsymbol{z}}_t\mathrm{d}\boldsymbol{z}_t, \quad (4)$$

where $\{\boldsymbol{z}_t, \tilde{\boldsymbol{z}}_t\}$ denotes the latent feature vectors, $\boldsymbol{c}_t^{(x)}$ denotes a speaker-code vector of the input speaker x, and $\boldsymbol{e}_t^{(x)}$ denotes the excitation features. In VAE [30], posterior form of latent features $p_{\boldsymbol{\theta},\tilde{\boldsymbol{\theta}}}(\boldsymbol{z}_t, \tilde{\boldsymbol{z}}_t|\boldsymbol{x}_t) = \frac{p_{\boldsymbol{\theta},\tilde{\boldsymbol{\theta}}}(\boldsymbol{x}_t, \boldsymbol{z}_t, \tilde{\boldsymbol{z}}_t)}{p_{\boldsymbol{\theta},\tilde{\boldsymbol{\theta}}}(\boldsymbol{x}_t)}$ is utilized to handle the likelihood of Eq. (4) with Gibbs' inequality as follows:

$$\log p_{\boldsymbol{\theta},\tilde{\boldsymbol{\theta}}}(\boldsymbol{x}_t, \boldsymbol{e}_t^{(x)}|\boldsymbol{c}_t^{(x)}) \geq \mathcal{L}(\boldsymbol{\Psi}; \boldsymbol{x}_t, \boldsymbol{c}_t^{(x)}, \boldsymbol{e}_t^{(x)}), \quad (5)$$

where $\boldsymbol{\Psi} = \{\boldsymbol{\theta}, \tilde{\boldsymbol{\theta}}, \boldsymbol{\phi}, \tilde{\boldsymbol{\phi}}\}$ and the variational/evidence lower bound (ELBO) $\mathcal{L}(\boldsymbol{\Psi}; \boldsymbol{x}_t, \boldsymbol{c}_t^{(x)}, \boldsymbol{e}_t^{(x)})$ is given by

$$\begin{aligned}\mathbb{E}_{q_{\boldsymbol{\phi},\tilde{\boldsymbol{\phi}}}(\boldsymbol{z}_t,\tilde{\boldsymbol{z}}_t|\boldsymbol{x}_t)}[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{z}_t,\tilde{\boldsymbol{z}}_t,\boldsymbol{c}^{(x)},\boldsymbol{e}_t^{(x)})] - \mathrm{KL}(q_{\boldsymbol{\phi}}(\boldsymbol{z}_t|\boldsymbol{x}_t)||p_{\boldsymbol{\theta}}(\boldsymbol{z}_t)) \\ + \mathbb{E}_{q_{\tilde{\boldsymbol{\phi}}}(\tilde{\boldsymbol{z}}_t|\boldsymbol{x}_t)}[\log p_{\tilde{\boldsymbol{\theta}}}(\boldsymbol{e}_t^{(x)}|\tilde{\boldsymbol{z}}_t,\boldsymbol{c}^{(x)})] - \mathrm{KL}(q_{\tilde{\boldsymbol{\phi}}}(\tilde{\boldsymbol{z}}_t|\boldsymbol{x}_t)||p_{\tilde{\boldsymbol{\theta}}}(\tilde{\boldsymbol{z}}_t)), \quad (6)\end{aligned}$$

and $\boldsymbol{c}^{(x)}$ denotes a time-invariant speaker-code of the input speaker x. The sets of encoder and decoder parameters are respectively denoted as $\{\boldsymbol{\phi}, \tilde{\boldsymbol{\phi}}\}$ and $\{\boldsymbol{\theta}, \tilde{\boldsymbol{\theta}}\}$. The prior distributions of latent features are denoted as $p_{\boldsymbol{\theta}}(\boldsymbol{z}_t)$ and $p_{\tilde{\boldsymbol{\theta}}}(\tilde{\boldsymbol{z}}_t)$. The variational posteriors are denoted as $q_{\boldsymbol{\phi}}(\boldsymbol{z}_t|\boldsymbol{x}_t)$ and $q_{\tilde{\boldsymbol{\phi}}}(\tilde{\boldsymbol{z}}_t|\boldsymbol{x}_t)$. In addition, to improve the latent disentanglement performance, we also utilize variational posterior $q_{\boldsymbol{\phi},\tilde{\boldsymbol{\phi}}}(\boldsymbol{c}_t^{(x)}|\boldsymbol{x}_t)$.

From Eq. (6), the conditional probability density function (p.d.f.) of the input spectral features $\boldsymbol{x}_t$, as well as of the converted spectral features $\boldsymbol{y}_t$, are given by

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{z}_t, \tilde{\boldsymbol{z}}_t, \boldsymbol{c}^{(x)}, \boldsymbol{e}_t^{(x)}) = \mathcal{N}(\boldsymbol{x}_t; \boldsymbol{\mu}_t^{(x)}, \boldsymbol{\Sigma}_t^{(x)}), \quad (7)$$

$$p_{\boldsymbol{\theta}}(\boldsymbol{y}_t|\boldsymbol{z}_t, \tilde{\boldsymbol{z}}_t, \boldsymbol{c}^{(y)}, \boldsymbol{e}_t^{(y)}) = \mathcal{N}(\boldsymbol{y}_t; \boldsymbol{\mu}_t^{(y)}, \boldsymbol{\Sigma}_t^{(y)}), \quad (8)$$

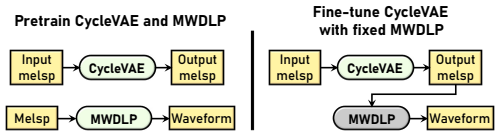


Figure 2: *Proposed model development steps: separately pre-train CycleVAE spectral model and MWDLP neural vocoder (left), then fine-tune CycleVAE modules with fixed MWDLP to utilize its waveform domain loss (right).*

where $\boldsymbol{c}^{(y)}$ denotes the time-invariant speaker-code of the converted speaker y, $\boldsymbol{e}_t^{(y)}$ denotes the converted excitation features, e.g., linearly converted log-F0 [31], and

$$\boldsymbol{z}_t = \boldsymbol{\mu}_t^{(z)} - \boldsymbol{\sigma}_t^{(z)} \odot \boldsymbol{\epsilon},\ \tilde{\boldsymbol{z}}_t = \boldsymbol{\mu}_t^{(\tilde{z})} - \boldsymbol{\sigma}_t^{(\tilde{z})} \odot \boldsymbol{\epsilon},\ \boldsymbol{\epsilon} \sim \mathcal{L}(\mathbf{0}, \mathbf{1}), \quad (9)$$

the Hadamard product is denoted as $\odot$, $\mathcal{L}(\mathbf{0}, \mathbf{1})$ denotes the standard Laplacian distribution. The Gaussian distribution with a mean vector $\boldsymbol{\mu}$ and a covariance matrix $\boldsymbol{\Sigma}$ is denoted as $\mathcal{N}(;\boldsymbol{\mu}, \boldsymbol{\Sigma})$. The output of encoders $\{\boldsymbol{\phi}, \tilde{\boldsymbol{\phi}}\}$ are denoted as $\{\boldsymbol{\mu}_t^{(z)}, \boldsymbol{\sigma}_t^{(z)}, \boldsymbol{\mu}_t^{(\tilde{z})}, \boldsymbol{\sigma}_t^{(\tilde{z})}\}$, while the output of decoder $\boldsymbol{\theta}$ is denoted as $\{\boldsymbol{\mu}_t^{(x)}, \mathrm{diag}(\boldsymbol{\Sigma}_t^{(x)})\}$ or $\{\boldsymbol{\mu}_t^{(y)}, \mathrm{diag}(\boldsymbol{\Sigma}_t^{(y)})\}$. To improve the conversion performance, we also utilize the p.d.f. of converted excitation $p_{\tilde{\boldsymbol{\theta}}}(\boldsymbol{e}_t^{(y)}|\tilde{\boldsymbol{z}}_t, \boldsymbol{c}^{(y)})$ in a similar manner as in Eq. (6) of the excitation of input speaker. The reconstructed/converted mel-spectrogram is generated from sampling the Gaussian p.d.f. in Eq.(7) or (8), respectively.

To provide network regularization with cycle-consistency, an auxiliary for the likelihood of Eq. (4) is defined as follows:

$$p_{\boldsymbol{\theta},\tilde{\boldsymbol{\theta}}}(\boldsymbol{x}_t, \boldsymbol{e}_t^{(x)}|\boldsymbol{e}_t^{(y)}, \boldsymbol{c}_t^{(x)}, \boldsymbol{c}_t^{(y)}) = \iiint p_{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{y}_t, \boldsymbol{z}_t, \tilde{\boldsymbol{z}}_t, \boldsymbol{e}_t^{(x)}, \boldsymbol{c}_t^{(x)}) p_{\boldsymbol{\theta}}(\boldsymbol{y}_t|\boldsymbol{x}_t, \boldsymbol{z}_t, \tilde{\boldsymbol{z}}_t, \boldsymbol{e}_t^{(y)}, \boldsymbol{c}_t^{(y)}) p_{\tilde{\boldsymbol{\theta}}}(\boldsymbol{e}_t^{(x)}|\tilde{\boldsymbol{z}}_t, \boldsymbol{c}_t^{(x)}) \mathrm{d}\tilde{\boldsymbol{z}}_t \mathrm{d}\boldsymbol{z}_t \mathrm{d}\boldsymbol{y}_t, \quad (10)$$

where by taking the expected values of the converted spectral $\boldsymbol{y}_t$ through sampling from Eq. (8), Eq. (10) is rewritten as

$$p_{\boldsymbol{\theta},\tilde{\boldsymbol{\theta}}}(\boldsymbol{x}_t, \boldsymbol{e}_t^{(x)}|\boldsymbol{e}_t^{(y)}, \boldsymbol{c}_t^{(x)}, \boldsymbol{c}_t^{(y)}) = \iint \mathbb{E}_{p_{\boldsymbol{\theta}}(\boldsymbol{y}_t|\boldsymbol{x}_t, \boldsymbol{e}_t^{(y)}, \boldsymbol{c}_t^{(y)})}[p_{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{y}_t, \boldsymbol{z}_t, \tilde{\boldsymbol{z}}_t, \boldsymbol{e}_t^{(x)}, \boldsymbol{c}_t^{(x)})] p_{\tilde{\boldsymbol{\theta}}}(\boldsymbol{e}_t^{(x)}|\tilde{\boldsymbol{z}}_t, \boldsymbol{c}_t^{(x)}) p_{\boldsymbol{\theta}}(\boldsymbol{z}_t) p_{\tilde{\boldsymbol{\theta}}}(\tilde{\boldsymbol{z}}_t) \mathrm{d}\tilde{\boldsymbol{z}}_t \mathrm{d}\boldsymbol{z}_t. \quad (11)$$

Therefore, as in Eq. (5), we approximate the true posterior $p_{\boldsymbol{\theta},\tilde{\boldsymbol{\theta}}}(\boldsymbol{z}_t, \tilde{\boldsymbol{z}}_t|\boldsymbol{x}_t, \boldsymbol{y}_t)$ through the following form

$$\log p_{\boldsymbol{\theta},\tilde{\boldsymbol{\theta}}}(\boldsymbol{x}_t, \boldsymbol{e}_t^{(x)}, \boldsymbol{e}_t^{(y)}|\boldsymbol{c}_t^{(x)}, \boldsymbol{c}_t^{(y)}) \geq \mathcal{L}(\boldsymbol{\Psi}; \boldsymbol{x}_t, \boldsymbol{y}_t, \boldsymbol{e}_t^{(x)}, \boldsymbol{e}_t^{(y)}, \boldsymbol{c}_t^{(x)}, \boldsymbol{c}_t^{(y)}) \quad (12)$$

where the ELBO $\mathcal{L}(\boldsymbol{\Psi}; \boldsymbol{x}_t, \boldsymbol{y}_t, \boldsymbol{e}_t^{(x)}, \boldsymbol{e}_t^{(y)}, \boldsymbol{c}_t^{(x)}, \boldsymbol{c}_t^{(y)})$ is given by

$$\mathbb{E}_{p_{\boldsymbol{\theta}}(\boldsymbol{y}_t|\boldsymbol{x}_t, \boldsymbol{e}_t^{(y)}, \boldsymbol{c}_t^{(y)})}\big[\mathbb{E}_{q_{\boldsymbol{\phi},\tilde{\boldsymbol{\phi}}}(\boldsymbol{z}_t, \tilde{\boldsymbol{z}}_t|\boldsymbol{x}_t, \boldsymbol{y}_t)}[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{z}_t, \tilde{\boldsymbol{z}}_t, \boldsymbol{c}^{(x)}, \boldsymbol{e}_t^{(x)})]] - \mathrm{KL}(q_{\boldsymbol{\phi}}(\boldsymbol{z}_t|\boldsymbol{x}_t, \boldsymbol{y}_t)||p_{\boldsymbol{\theta}}(\boldsymbol{z}_t)) - \mathrm{KL}(q_{\tilde{\boldsymbol{\phi}}}(\tilde{\boldsymbol{z}}_t|\boldsymbol{x}_t, \boldsymbol{y}_t)||p_{\tilde{\boldsymbol{\theta}}}(\tilde{\boldsymbol{z}}_t)) + \mathbb{E}_{q_{\tilde{\boldsymbol{\phi}}}(\tilde{\boldsymbol{z}}_t|\boldsymbol{y}_t)}[\log p_{\tilde{\boldsymbol{\theta}}}(\boldsymbol{e}_t^{(x)}|\tilde{\boldsymbol{z}}_t, \boldsymbol{c}^{(x)})]\big]. \quad (13)$$

Hence, the optimization of network parameters $\hat{\boldsymbol{\Psi}} = \{\hat{\boldsymbol{\theta}}, \hat{\tilde{\boldsymbol{\theta}}}, \hat{\boldsymbol{\phi}}, \hat{\tilde{\boldsymbol{\phi}}}\}$ is performed with Eqs. (5) and (12) as follows:

$$\hat{\boldsymbol{\Psi}} = \underset{\boldsymbol{\theta},\tilde{\boldsymbol{\theta}},\boldsymbol{\phi},\tilde{\boldsymbol{\phi}}}{\mathrm{argmax}} \sum_{t=1}^{T} \mathcal{L}(\boldsymbol{\Psi}; \boldsymbol{x}_t, \boldsymbol{y}_t, \boldsymbol{e}_t^{(x)}, \boldsymbol{e}_t^{(y)}, \boldsymbol{c}_t^{(x)}, \boldsymbol{c}_t^{(y)}) + \mathcal{L}(\boldsymbol{\Psi}; \boldsymbol{x}_t, \boldsymbol{c}_t^{(x)}, \boldsymbol{e}_t^{(x)}) \quad (14)$$

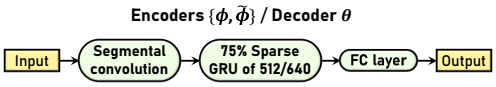


Figure 3: *Network structure of encoders $\{\boldsymbol{\phi}, \tilde{\boldsymbol{\phi}}\}$ and decoder $\boldsymbol{\theta}$ with a base GRU size of 512 and 640, respectively, which are sparsified to 75% density. Segmental convolution is made to take into account p previous and n succeeding frames, as in [21], with $p = 3, n = 1$ and $p = 4, n = 0$ for encoders and decoders, respectively.*

3.2. Fine-tuning with MWDLP-based waveform loss

As illustrated on the right side of Fig. 1 and Fig. (2), to perform fine-tuning with MWDLP loss, we discard the estimation of excitation, where the likelihood in Eq. (4) is rewritten as follows:

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{c}_t^{(x)}) = \iint p_{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{z}_t, \tilde{\boldsymbol{z}}_t, \boldsymbol{c}_t^{(x)}) p_{\boldsymbol{\theta}}(\boldsymbol{z}_t) p_{\boldsymbol{\theta}}(\tilde{\boldsymbol{z}}_t) \mathrm{d}\tilde{\boldsymbol{z}}_t \mathrm{d}\boldsymbol{z}_t. \quad (15)$$

As in Eqs. (5) and (6), the inequality form to approximate the true posterior $p_{\boldsymbol{\theta}}(\boldsymbol{z}_t, \tilde{\boldsymbol{z}}_t|\boldsymbol{x}_t)$ is as follows:

$$\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{c}_t^{(x)}) \geq \mathcal{L}(\boldsymbol{\Lambda}; \boldsymbol{x}_t, \boldsymbol{c}_t^{(x)}) \quad (16)$$

where $\boldsymbol{\Lambda} = \{\boldsymbol{\theta}, \boldsymbol{\phi}, \tilde{\boldsymbol{\phi}}\}$, and the ELBO $\mathcal{L}(\boldsymbol{\Lambda}; \boldsymbol{x}_t, \boldsymbol{c}_t^{(x)})$ is given by

$$\mathbb{E}_{q_{\boldsymbol{\phi},\tilde{\boldsymbol{\phi}}}(\boldsymbol{z}_t, \tilde{\boldsymbol{z}}_t|\boldsymbol{x}_t)}[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{z}_t, \tilde{\boldsymbol{z}}_t, \boldsymbol{c}^{(x)})] - \mathrm{KL}(q_{\boldsymbol{\phi}}(\boldsymbol{z}_t|\boldsymbol{x}_t)||p_{\boldsymbol{\theta}}(\boldsymbol{z}_t)) - \mathrm{KL}(q_{\tilde{\boldsymbol{\phi}}}(\tilde{\boldsymbol{z}}_t|\boldsymbol{x}_t)||p_{\boldsymbol{\theta}}(\tilde{\boldsymbol{z}}_t)). \quad (17)$$

Likewise, following Eq. (11), the auxiliary form of Eq. (15), to provide cycle-consistency, is defined as follows:

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{c}_t^{(x)}, \boldsymbol{c}_t^{(y)}) = \iint \mathbb{E}_{p_{\boldsymbol{\theta}}(\boldsymbol{y}_t|\boldsymbol{x}_t, \boldsymbol{c}_t^{(y)})}[p_{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{y}_t, \boldsymbol{z}_t, \tilde{\boldsymbol{z}}_t, \boldsymbol{c}_t^{(x)})] p_{\boldsymbol{\theta}}(\boldsymbol{z}_t) p_{\boldsymbol{\theta}}(\tilde{\boldsymbol{z}}_t) \mathrm{d}\tilde{\boldsymbol{z}}_t \mathrm{d}\boldsymbol{z}_t. \quad (18)$$

Following Eqs. (12) and (13), the inequality form to approximate the true posterior $p_{\boldsymbol{\theta}}(\boldsymbol{z}_t, \tilde{\boldsymbol{z}}_t|\boldsymbol{x}_t, \boldsymbol{y}_t)$ is defined as

$$\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{c}_t^{(x)}, \boldsymbol{c}_t^{(y)}) \geq \mathcal{L}(\boldsymbol{\Lambda}; \boldsymbol{x}_t, \boldsymbol{y}_t, \boldsymbol{c}_t^{(x)}, \boldsymbol{c}_t^{(y)}) \quad (19)$$

where the ELBO $\mathcal{L}(\boldsymbol{\Lambda}; \boldsymbol{x}_t, \boldsymbol{y}_t, \boldsymbol{c}_t^{(x)}, \boldsymbol{c}_t^{(y)})$ is given by

$$\mathbb{E}_{p_{\boldsymbol{\theta}}(\boldsymbol{y}_t|\boldsymbol{x}_t, \boldsymbol{c}_t^{(y)})}\big[\mathbb{E}_{q_{\boldsymbol{\phi},\tilde{\boldsymbol{\phi}}}(\boldsymbol{z}_t, \tilde{\boldsymbol{z}}_t|\boldsymbol{x}_t, \boldsymbol{y}_t)}[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{z}_t, \tilde{\boldsymbol{z}}_t, \boldsymbol{c}^{(x)})]] - \mathrm{KL}(q_{\boldsymbol{\phi}}(\boldsymbol{z}_t|\boldsymbol{x}_t, \boldsymbol{y}_t)||p_{\boldsymbol{\theta}}(\boldsymbol{z}_t)) - \mathrm{KL}(q_{\tilde{\boldsymbol{\phi}}}(\tilde{\boldsymbol{z}}_t|\boldsymbol{x}_t, \boldsymbol{y}_t)||p_{\boldsymbol{\theta}}(\tilde{\boldsymbol{z}}_t))\big]. \quad (20)$$

Finally, the set of updated parameters $\hat{\boldsymbol{\Lambda}} = \{\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\phi}}, \hat{\tilde{\boldsymbol{\phi}}}\}$ is obtained by combining Eqs. (16), (19), and Eq. (1), i.e., the likelihood of the waveform samples from MWDLP, as follows:

$$\{\hat{\boldsymbol{\Lambda}}\} = \underset{\boldsymbol{\theta},\boldsymbol{\phi},\tilde{\boldsymbol{\phi}}}{\mathrm{argmax}} \sum_{t=1}^{T} \mathcal{L}(\boldsymbol{\Lambda}; \boldsymbol{x}_t, \boldsymbol{y}_t, \boldsymbol{c}_t^{(x)}, \boldsymbol{c}_t^{(y)}) + \mathcal{L}(\boldsymbol{\Lambda}; \boldsymbol{x}_t, \boldsymbol{c}_t^{(x)}) + \sum_{m=1}^{M} \sum_{\tau=1}^{\mathcal{T}} \log p(s_\tau^{(m)}|\boldsymbol{s}_{1:\tau-1}^{(M)}, \tilde{\boldsymbol{x}}_\tau), \quad (21)$$

where the conditioning feature vector $\tilde{\boldsymbol{x}}_\tau$ is built from the sampled reconstructed mel-spectrogram $\boldsymbol{x}_t$ of the input speaker x.

Table 1: *Results of accuracy (acc.) measurement on log-global-variance distance of mel-cepstrum (LGD), mel-cepstral distortion (MCD), unvoiced/voiced decision error (U/V), and root-mean-square-error of F0 between converted and target waveform on intra-lingual pairs.*

Intra-lingual acc.	LGD	MCD [dB]	U/V [%]	F0 [Hz]
ASR+TTS [22]	0.29	6.91	16.20	**22.29**
CycVAE+PWG [27]	0.34	**6.67**	**14.36**	24.91
NU T23 [32]	**0.28**	7.50	18.94	23.20
LLRT CycVAE	0.36	**7.41**	**15.35**	25.74
LLRT CycVAE+FT	**0.28**	7.51	17.27	**25.17**

Table 2: *Results of accuracy (acc.) measurement on log-global-variance distance of mel-cepstrum (LGD), mel-cepstral distortion (MCD), unvoiced/voiced decision error (U/V), and root-mean-square-error of F0 between converted and target waveform on cross-lingual pairs.*

Cross-lingual acc.	LGD	MCD [dB]	U/V [%]	F0 [Hz]
ASR+TTS [22]	0.39	8.78	14.84	**21.12**
CycVAE+PWG [27]	0.34	**7.56**	**13.86**	22.83
NU T23 [32]	**0.24**	8.50	16.33	22.68
LLRT CycVAE	0.39	**8.22**	**15.25**	20.91
LLRT CycVAE+FT	**0.30**	8.44	15.81	**20.91**

3.3. Network architecture and sparsification

The network architecture of the encoders and decoders of the proposed CycleVAE is illustrated in Fig.3. As in [21], a segmental convolution is utilized to take into account p preceding and n succeeding frames. To realize LLRT VC, we use $p = 3, n = 1$ for encoders of CycleVAE, $p = 4, n = 0$ for decoder of CycleVAE, and $p = 5, n = 1$ for the MWDLP neural vocoder, which yields a total of 2 lookup frames.

In addition, a sparsification procedure for CycleVAE network is also performed, as in [18, 21], with 75% target density for the gated recurrent unit (GRU) modules of encoders $\{\boldsymbol{\phi}, \tilde{\boldsymbol{\phi}}\}$ and decoder $\boldsymbol{\theta}$. The base hidden units size of GRU encoders is 512, while that of the decoder is 640. The target density ratios for each reset, update, and new gates of the GRU recurrent matrices are respectively $0.685, 0.685, 0.88$.

4. Experimental evaluation

4.1. Experimental conditions

We used the Voice Conversion Challenge (VCC) 2020 [8] dataset, which consisted of 8 English speakers, 2 German speakers, 2 Finnish speakers, and 2 Mandarin speakers, each uttered 70 sentences in their languages. For the training set, 60 sentences were used, while the remaining 10 sentences were for the development set. Additional 25 English utterances from each speaker were provided for evaluation. In the evaluation, we utilized two baseline systems of VCC 2020: cascaded automatic speech recognition (ASR) with TTS (ASR+TTS) [22] and CycleVAE with Parallel WaveGAN (CycVAE+PWG) [27], as well as Nagoya University (NU) T23 system [32]. 2 English source, 2 English target (intra-lingual), and 2 German target (cross-lingual) speakers were utilized in the evaluation.

As spectral features, we used 80-dimensional mel-spectrogram, which was extracted frame-by-frame from magnitude spectra. The number of FFT length in analysis was 2048. 27.5 ms Hanning window with 10 ms frame shift were used. The sampling rate was $24,000$ Hz. As the target intermediate excitation features used in Section 3.1, we used F0, aperiodicities, and their voicing decisions, which were extracted from the speech waveform using WORLD [14]. The excitation $\boldsymbol{e}_t^{(y)}$ of converted speaker y was set to linearly converted log-F0 [31].

Table 3: *Result on automatic speech recognition accuracy (ASR acc.) on intra- and cross-lingual conversions with word error rate (WER) and character error rate (CER) measurements.*

ASR acc.	Intra-lingual WER	Intra-lingual CER	Cross-lingual WER	Cross-lingual CER
Source	18.5	3.7	-	-
Target	17.5	3.0	19.2	4.1
ASR+TTS [22]	**25.1**	**7.5**	30.3	12.2
CycVAE+PWG [27]	28.2	9.6	29.6	10.3
NU T23 [32]	37.3	14.9	**25.2**	**7.6**
LLRT CycVAE	33.8	13.6	34.0	12.4
LLRT CycVAE+FT	**25.2**	**7.9**	**26.1**	**7.9**

Other than the configuration of segmental convolution in Section 3.3, the hyperparameters of MWDLP neural vocoder was the same as in [21] with the use of $K = 8$ data-driven LP coefficients and STFT loss. As well as for the CycleVAE-based spectral model, the encoders $\{\boldsymbol{\phi}, \tilde{\boldsymbol{\phi}}\}$ and the decoder $\boldsymbol{\theta}$ were set the same as in 3.3. On the other hand, the excitation decoder $\tilde{\boldsymbol{\theta}}$ described in Section 3.1 used the same structure as the other encoders/decoder, but utilizing a dense GRU with 128 hidden units. A classifier network with similar structure utilizing a GRU with 32 hidden units was employed to handle the variational speaker posteriors $q(\boldsymbol{c}_t^{(x)}|\boldsymbol{x}_t)$ and $q(\boldsymbol{c}_t^{(y)}|\boldsymbol{y}_t)$. Additionally, each of the encoders was also set to estimate the speaker posteriors along with the latent posteriors.

The training procedure was as described in Sections 3.1 and 3.2, where the standard Laplacian prior was replaced with the posterior of the pretrained CycleVAE. In addition, we performed final fine-tuning of CycleVAE by fixing the encoders and updating only decoder $\boldsymbol{\theta}$ (LLRT CycVAE+FT). In all CycleVAE optimizations, the spectral loss included Gaussian p.d.f. term and the loss of the sampled mel-spectrogram. Further, in the fine-tuning steps, we included loss from full-resolution magnitude spectra, which was obtained using inverted mel-filterbank and the sampled mel-spectrogram. The waveform domain loss included the set of loss in [21] and the differences of the output of all MWDLP layers when fed with original spectra and generated spectra (layer-wise loss).

We used a single-core of Intel Xeon Gold 6230 2.1 GHz, Intel Xeon Gold 6142 2.6 GHz, and Intel i7-7500U 2.7 GHz CPUs to measure the real-time factor (RTF), which respectively yield 0.87, 0.87, and 0.95 RTFs. The total delay is 23.75 ms, which was the sum of the half of the window length (1st frame) and one frame shift, i.e., 2 lookup frames. The model development software, real-time implementation, and audio samples have been made available at `https://github.com/patrickltobing/cyclevae-vc-neuralvoco`.

4.2. Objective evaluation

In the objective evaluation, we measured the accuracies of the generated waveforms to the target ground truth and the accuracies of automatic speech recognition (ASR) output. The former was measured with the use of mel-cepstral distortion (MCD), root-mean-square error of F0, unvoiced/voiced decision error (U/V), and log of global variance [31] distance of the mel-cepstrum (LGD). The latter was measured with word error rate (WER) and character error rate (CER). 28-dimensional mel-cepstral coefficients were extracted from WORLD [14] spectral envelope to compute the MCD. For ASR, we used ESPnet's [33] latest pretrained model on LibriSpeech [34] data.

Table 4: *Result of mean opinion score (MOS) test on naturalness for intra- and cross-lingual conversions in same-gender (SGD) and cross-gender (XGD) pairs. * denotes systems with statistically significant different values ($\alpha < 0.05$) compared to LLRT CycleVAE+FT in each conversion categories.*

MOS	All	Intra-lingual		Cross-lingual	
		SGD	XGD	SGD	XGD
Source	4.68	-	-	-	-
Target	4.69	-	-	-	-
ASR+TTS [22]	4.01	**4.32***	4.15*	3.84	3.72
CycVAE+PWG [27]	3.85*	3.85*	3.75	3.94	3.87
NU T23 [32]	**4.23***	4.30*	**4.21***	**4.23***	**4.21***
LLRT CycVAE	3.33*	3.30*	3.19*	3.48*	3.19*
LLRT CycVAE+FT	**3.96**	**3.99**	**3.85**	**4.02**	**3.96**

Table 5: *Result of speaker similarity [%] test for intra- and cross-lingual conversions in same-gender (SGD) and cross-gender (XGD) pairs. * denotes systems with statistically significant different values ($\alpha < 0.05$) compared to LLRT CycleVAE+FT in each conversion categories.*

Speaker similarity [%]	All	Intra-lingual		Cross-lingual	
		SGD	XGD	SGD	XGD
Source	8.01	-	-	-	-
Target	90.05	-	-	-	-
ASR+TTS [22]	**89.43***	91.80	87.10*	**84.12***	**87.90***
CycVAE+PWG [27]	78.63	85.25	77.42	74.19	77.78
NU T23 [32]	80.24*	**93.50***	**89.60***	71.77	66.13*
LLRT CycVAE	70.22	76.99	70.49	67.20	66.13*
LLRT CycVAE+FT	**77.55**	**86.18**	**74.16**	**75.24**	**74.60**

The results on the accuracies of the generated waveforms are shown in Tables 1 and 2, which correspond to the intra- and the cross-lingual conversion pairs, respectively. It can be observed that the proposed LLRT system based on CycleVAE and MWDLP utilizing fine-tuning with waveform domain loss (LLRT CycVAE+FT) achieves better LGD values (less oversmoothed) in intra- and cross-lingual conversions than the proposed system without fine-tuning (LLRT CycleVAE) with values of 0.28 and 0.36, respectively, in intra-lingual, and 0.30 and 0.39, respectively, in cross-lingual. Furthermore, it beats the LGD values of CycVAE+PWG that uses non-LLRT system, and beats NU T23 system in MCD, U/V, and F0 for cross-lingual, which uses non-LLRT CycleVAE with WaveNet.

Lastly, the ASR result is shown in Table 3, which shows WERs and CERs for the intra- and the cross-lingual conversions. It can be clearly observed that the proposed LLRT CycVAE+FT outperforms the proposed system without fine-tuning LLRT CycVAE with WER and CER values of 26.1 and 7.9 in intra-lingual and of 25.2 and 7.9 in cross-lingual. These values are also lower than the non-LLRT CycleVAE system of the VCC 2020 baseline (CycVAE+PWG) and similar to that of the non-LLRT CycleVAE of NU T23 in cross-lingual conversions.

4.3. Subjective evaluation

In the subjective evaluation, we conducted two listening tests, each to judge the naturalness of speech waveform and the speaker similarity to a reference target speech. The former is conducted with a mean opinion score (MOS) test using a 5-scaled score ranging from 1 (very bad) to 5 (very good). The latter is conducted with a speaker similarity test as in [8], where "same" or "not-same" decision had to be chosen along with "sure" or "not-sure" decision as a confidence measure. 10 utterances from the evaluation set was used. The number of participants on Amazon Mechanical Turk was 19 and 13, respectively, for MOS and speaker similarity tests.

The result of MOS test on naturalness is shown in Table 4. It can be observed that the proposed LLRT VC system benefits from the fine-tuning approach (LLRT CycleVAE+FT), yielding significantly higher naturalness in all categories than the LLRT CycleVAE system, with values of 3.96, 3.99, 3.85, 4.02, and 3.96 for all, intra-lingual same-gender (SGD), intra-lingual cross-gender (XGD), cross-lingual SGD and cross-lingual XGD, respectively. On the other hand, the result of speaker similarity test is shown in Table 5. The tendency is also similar, where the proposed LLRT CycleVAE+FT system has better speaker accuracy than the LLRT CycleVAE in all categories, while achieving similar accuracies to the non-LLRT CycleVAE systems: CycVAE+PWG and NU T23 (cross-lingual).

5. Discussion

The proposed method of fine-tuning the CycleVAE-based spectral model with MWDLP-based waveform modeling significantly improves the converted speech waveform. From our investigation, the use of mel-spectrogram sampling from Gaussian p.d.f. in Eqs.(7) and (8) works very well with the waveform domain loss. In addition, we also found that layer-wise loss from neural vocoder helps to provide more natural outcome. Our reasoning is that the generated spectra will not be exactly the same as the natural spectra that corresponds to the natural waveform, but we assume that there is a domain for generated spectra that could provide quite reasonable approximation for generating the natural waveform by explicitly guiding through all layers of the neural vocoder in addition of the waveform loss.

The largest average RTF factors for each module are as follows: 0.14 for two encoders, 0.13 for decoder, 0.56 for MWDLP, and 0.12 for others including input/output, memory allocation, etc. The total of these RTF values, i.e., $\sim$9.5 ms, should be lower than the length of the frame shift, which is 10 ms. However, in practical situation, a larger margin is required to avoid glitching caused by outliers of RTF values that are larger than the frame shift. In future work, we will investigate lower size of MWDLP and/or 8-bit model quantization.

6. Conclusions

We have presented a novel low-latency real-time (LLRT) non-parallel voice conversion (VC) framework based on cyclic variational autoencoder (CycleVAE) and multiband WaveRNN with data-driven linear prediction (MWDLP). The proposed system utilizes mel-spectrogram features as the spectral parameters of the speech waveform, which are used in the CycleVAE-based spectral model and the MWDLP neural vocoder. To realize LLRT VC, CycleVAE modules undergo a sparsification procedure with respect to their recurrent matrices. In addition, we propose to use waveform domain loss from a fixed pretrained MWDLP to fine-tune the CycleVAE modules. The experimental resuts have demonstrated that the proposed system is capable of achieving high-performance VC, while allowing its usage for LLRT applications with 0.87–0.95 real-time factor using a single-core of 2.1–2.7 GHz CPU on 27.5 ms window length, 10 ms frame shift, and 2 lookup frames.

7. Acknowledgements

This work was partly supported by JSPS KAKENHI Grant Number 17H06101 and JST, CREST Grant Number JPMJCR19A3.

8. References

[1] D. B. Childers, B. Yegnanarayana, and K. Wu, "Voice conversion: Factors responsible for quality," in *Proc. ICASSP*, Florida, USA, Mar. 1985, pp. 748–751.

[2] M. Abe, S. Nakamura, K. Shikano, and H. Kuwabara, "Voice conversion through vector quantization," *J. of the Acoust. Soc. of Japan (E)*, vol. 11, no. 2, pp. 71–76, 1990.

[3] K. Tanaka, T. Toda, G. Neubig, S. Sakti, and S. Nakamura, "A hybrid approach to Electrolaryngeal speech enhancement based on spectral subtraction and statistical voice conversion," in *Proc. INTERSPEECH*, Lyon, France, Sep. 2013, pp. 3067–3071.

[4] O. Türk and M. Schröder, "Evaluation of expressive speech synthesis with voice conversion and copy resynthesis techniques," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 18, no. 5, pp. 965–973, 2010.

[5] K. Kobayashi, T. Toda, and S. Nakamura, "Intra-gender statistical singing voice conversion with direct waveform modification using log-spectral differential," *Speech Commun.*, vol. 99, pp. 211–220, 2018.

[6] T. Toda, M. Nakagiri, and K. Shikano, "Statistical voice conversion techniques for body-conducted unvoiced speech enhancement," *IEEE Trans. Audio Speech Lang. Process.*, vol. 20, no. 9, pp. 2505–2517, 2012.

[7] T. Kinnunen, Z.-Z. Wu, K. A. Lee, F. Sedlak, E. S. Chng, and H. Li, "Vulnerability of speaker verification systems against voice conversion spoofing attacks: The case of telephone speech," in *Proc. ICASSP*, Kyoto, Japan, Mar. 2012, pp. 4401–4404.

[8] Y. Zhao, W.-C. Huang, X. Tian, J. Yamagishi, R. K. Das, T. Kinnunen, Z. Ling, and T. Toda, "Voice Conversion Challenge 2020 intra-lingual semi-parallel and cross-lingual voice conversion," in *ISCA Joint Workshop for the Blizzard Challenge and Voice Conversion Challenge 2020*, Shanghai, China, Oct. 2020, pp. 80–98.

[9] T. Toda, T. Muramatsu, and H. Banno, "Implementation of computationally efficient real-time voice conversion," in *Proc. INTERSPEECH*, Portland, OR, USA, Sep. 2012.

[10] K. Kobayashi and T. Toda, "Implementation of low-latency Electrolaryngeal speech enhancement based on multi-task CLDNN," in *Proc. EUSIPCO*, Amsterdam, Netherlands, Jan. 2021, pp. 396–400.

[11] R. Arakawa, S. Takamichi, and H. Saruwatari, "Implementation of DNN-based real-time voice conversion and its improvements by audio data augmentation and mask-shaped device," in *Proc. SSW10*, Vienna, Austria, Sep. 2019, pp. 93–98.

[12] T. Saeki, Y. Saito, S. Takamichi, and H. Saruwatari, "Real-time, full-band, online DNN-based voice conversion system using a single CPU," *Proc. INTERSPEECH*, pp. 1021–1022, Oct. 2020.

[13] H. Kawahara, I. Masuda-Katsuse, and A. De Cheveigné, "Restructuring speech representations using a pitch-adaptive time-frequency smoothing and an instantaneous-frequency-based F0 extraction: Possible role of a repetitive structure in sounds," *Speech Commun.*, vol. 27, pp. 187–207, 1999.

[14] M. Morise, F. Yokomori, and K. Ozawa, "WORLD: A vocoder-based high-quality speech synthesis system for real-time applications," *IEICE Trans. Inf. Syst.*, vol. 99, no. 7, pp. 1877–1884, 2016.

[15] A. Tamamori, T. Hayashi, K. Kobayashi, K. Takeda, and T. Toda, "Speaker-dependent WaveNet vocoder," in *Proc. INTERSPEECH*, Stockholm, Sweden, Aug. 2017, pp. 1118–1122.

[16] X. Zhou, Z.-H. Ling, and S. King, "The Blizzard Challenge 2020," in *Proc. Joint Workshop for the Blizzard Challenge and Voice Conversion Challenge 2020*, Shanghai, China, Oct. 2020, pp. 1–18.

[17] N. Kalchbrenner, E. Elsen, K. Simonyan, S. Noury, N. Casagrande, E. Lockhart, F. Stimberg, A. v. d. Oord, S. Dieleman, and K. Kavukcuoglu, "Efficient neural audio synthesis," *arXiv preprint arXiv:1802.08435*, 2018.

[18] J.-M. Valin and J. Skoglund, "LPCNet: Improving neural speech synthesis through linear prediction," in *Proc. ICASSP*, Brighton, UK, May 2019, pp. 5891–5895.

[19] R. Prenger, R. Valle, and B. Catanzaro, "WaveGlow: A flow-based generative network for speech synthesis," in *Proc. ICASSP*, Brighton, UK, May 2019, pp. 3617–3621.

[20] R. Yamamoto, E. Song, and J.-M. Kim, "Parallel WaveGAN: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram," in *Proc. ICASSP*, Barcelona, Spain, May 2020, pp. 6199–6203.

[21] P. L. Tobing and T. Toda, "High-fidelity and low-latency universal neural vocoder based on multiband WaveRNN with data-driven linear prediction for discrete waveform modeling," *arXiv preprint arXiv:2105.09856*, 2021.

[22] W.-C. Huang, T. Hayashi, S. Watanabe, and T. Toda, "The sequence-to-sequence baseline for the Voice Conversion Challenge 2020: Cascading ASR and TTS," in *Proc. Joint Workshop for the Blizzard Challenge and Voice Conversion Challenge 2020*, Shanghai, China, Oct. 2020, pp. 160–164.

[23] J.-X. Zhang, L.-J. Liu, Y.-N. Chen, Y.-J. Hu, Y. Jiang, Z.-H. Ling, and L.-R. Dai, "Voice conversion by cascading automatic speech recognition and text-to-speech synthesis with prosody transfer," in *Proc. Joint Workshop for the Blizzard Challenge and Voice Conversion Challenge 2020*, Shanghai, China, Oct. 2020, pp. 121–125.

[24] C.-C. Hsu, H.-T. Hwang, Y.-C. Wu, Y. Tsao, and H.-M. Wang, "Voice conversion from non-parallel corpora using variational auto-encoder," in *Proc. APSIPA*, Jeju, South Korea, Dec. 2016, pp. 1–6.

[25] P. L. Tobing, Y.-C. Wu, T. Hayashi, K. Kobayashi, and T. Toda, "Non-parallel voice conversion with cyclic variational autoencoder," in *Proc. INTERSPEECH*, Graz, Austria, Sep. 2019, pp. 674–678.

[26] T. Kaneko, H. Kameoka, K. Tanaka, and N. Hojo, "StarGAN-VC2: Rethinking conditional methods for StarGAN-based voice conversion," in *Proc. INTERSPEECH*, Graz, Austria, Sep. 2019, pp. 679–683.

[27] P. L. Tobing, Y.-C. Wu, and T. Toda, "Baseline system of Voice Conversion Challenge 2020 with cyclic variational autoencoder and Parallel WaveGAN," in *Proc. Joint Workshop for the Blizzard Challenge and Voice Conversion Challenge 2020*, Shanghai, China, Oct. 2020, pp. 155–159.

[28] T. Q. Nguyen, "Near-perfect-reconstruction pseudo-QMF banks," *IEEE Trans. Sig. Process.*, vol. 42, no. 1, pp. 65–76, 1994.

[29] B. S. Atal and S. L. Hanauer, "Speech analysis and synthesis by linear prediction of the speech wave," *J. Acoust. Soc. Amer.*, vol. 50, no. 2B, pp. 637–655, 1971.

[30] D. P. Kingma and J. Ba, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[31] T. Toda, A. W. Black, and K. Tokuda, "Voice conversion based on maximum-likelihood estimation of spectral parameter trajectory," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 8, pp. 2222–2235, 2007.

[32] W.-C. Huang, P. L. Tobing, Y.-C. Wu, K. Kobayashi, and T. Toda, "The NU voice conversion system for the Voice Conversion Challenge 2020: On the effectiveness of sequence-to-sequence models and autoregressive neural vocoders," in *Proc. Joint Workshop for the Blizzard Challenge and Voice Conversion Challenge 2020*, Shanghai, China, Oct. 2020, pp. 165–169.

[33] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. Enrique Yalta Soplin, J. Heymann, M. Wiesner, N. Chen, A. Renduchintala, and T. Ochiai, "ESPnet: End-to-end speech processing toolkit," in *Proc. INTERSPEECH*, Hyderabad, India, Sep. 2018, pp. 2207–2211.

[34] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "LibriSpeech: an ASR corpus based on public domain audio books," in *Proc. ICASSP*, South Brisbane, Queensland, Australia, Apr. 2015, pp. 5206–5210.

11th ISCA Speech Synthesis Workshop (SSW 11)
26-28 August 2021, Budapest, Hungary

Factors Affecting the Evaluation of Synthetic Speech in Context

Johannah O'Mahony, Pilar Oplustil-Gallegos, Catherine Lai, Simon King

The Centre for Speech Technology Research, University of Edinburgh, United Kingdom
j.o'mahony-1@sms.ed.ac.uk

Abstract

Text-to-Speech synthesis is approaching the limit of naturalness that is possible from an isolated sentence. The focus of research is shifting to modelling contextual information, typically with the goal of producing better prosodic realisations by accounting for longer-range text dependencies from preceding sentences. But current evaluation methods were developed for single sentences and it is not yet clear how the evaluation of longer texts should be approached. Previous work suggests that evaluation of utterances in context can lead to an increase in Mean Opinion Score ratings, even when the synthesis technique is not context-aware. We investigated several factors that might explain this increase. Three experiments manipulated: the wording of instructions that participants received; the textual characteristics of context-stimulus pairs; and the prosodic realisation of the synthetic speech. We found that the wording of instructions has an impact on listeners' ratings of stimuli presented in context. The between-sentence context dependency of stimulus text has no impact on ratings. Listeners are, however, sensitive to prosodic differences, both in context and in isolation.

Index Terms: long-form Text-to-Speech, Text-to-Speech evaluation, context-aware Text-to-Speech

1. Introduction

Recent improvements in Text-to-Speech (TTS) modelling have paved the way for approaches which can take textual [1, 2] or acoustic context beyond the current utterance into account [3, 4]. Accounting for context has the potential to capture long-range text dependencies, discourse [5] and paragraph information [6], which are known to affect the prosodic realisation of an utterance. Context-sensitive prosody should exhibit increased variation compared to 'default' prosody generated for isolated sentences, and thus better long-form TTS [7]. However, in previous work, TTS output has been almost exclusively generated utterance-by-utterance and has therefore also been evaluated using isolated utterances [8, 9]. For context-sensitive TTS, appropriate evaluation paradigms are not yet fully developed.

One difficulty when rating prosodic variability is that countless realisations may be equally valid given a specific context [9]. Rating an utterance in context is a fundamentally different task to rating an utterance in isolation: varying the context can change the rating of the utterance. Conversely, in isolation the listener does not have access to any contextual information [8] (although participants might be able to imagine it [10]). This could potentially cause marked prosodic forms, elicited by a very specific context, to be rated lower when presented out of context, where listeners would expect default prosody. The opposite could also be true: perfectly natural and well-spoken utterances are rated highly in isolation, but when heard in an infelicitous context they are rated lower.

Clark et al. [8] found that utterances presented in isolation vs. in context had significantly different Mean Opinion Scores (MOS), with those heard in context receiving a higher rating when both the context and target were synthetic speech. Importantly, the synthetic speech used in their study was *not* context-sensitive. This boost in MOS score calls into question whether the MOS paradigm is the right way to evaluate synthetic speech in context.

The goal of this study is to discover what factors lead to such differences in MOS ratings. We conducted three experiments investigating various factors of interest.

- In experiment one, we test whether the **instructions** have an effect on MOS ratings of utterances presented in context.
- In experiment two, we assess whether between-sentence **textual context dependency** has an effect on MOS ratings.
- In experiment three, we test whether the MOS paradigm is suitable for rating **prosodically varied** synthetic speech.

Although Clark et al. tested a range of presentation types, including paragraphs, we will focus on a comparison between isolated utterances and context-target pairs in which an utterance is presented after a single context utterance. Finally, although prosodic realisation changes as a function of much more than the preceding sentence, e.g., pragmatic context, emotional state of the speaker, etc [9], we will concentrate on prosodic realisations which are determined by the textual context alone.

2. Related Work

As Text-to-Speech synthesis approaches its limit of naturalness, there is more and more focus on prosodic variability [10, 11, 12, for example] including the use of surrounding context to condition the realisation of the current utterance [1, 2, 3]. There is, however, little agreement on the best method for evaluating such prosodically-varied synthetic speech.

Some opt to use a qualitative approach. After testing whether prosodic realisations were perceptually distinct using a discriminative task, Hodari et al. asked participants to judge what effect different prosodic renditions had on the interpretation of the sentence, i.e., subtle differences in meaning or intent [10]. They found that participants were able to describe different contexts or situations where the prosodic variant would be found. A different qualitative approach was taken by Xu et al. who constructed different textual contexts and used these to generate different prosodic realisations of a single sentence in order to determine what effect their BERT-based context-aware model had on the prosody of a sentence [2].

Others opt for quantitative subjective evaluation using a MUSHRA-like paradigm. For example, Tyagi et al. used linguistic information, such as syntactic information and word embeddings to generate richer prosodic variability and evaluated both isolated utterances and long-form material [12]. In order to assess the quality of the prosodic output of individual sentences, they asked ten linguists to judge the appropriateness of the prosody in isolation. They stated that judging prosody requires domain-specific knowledge. This raises an issue with

10.21437/SSW.2021-26

devising appropriate metrics for prosodic felicity, if using non-expert listeners requires them to have an awareness of this dimension of the speech signal. Even among experts, however, it has been shown that inter-annotator agreement can be quite low [13]. For long-form evaluation, Tyagi et al. used crowd-sourced listeners and asked them to rate whole news stories for the *suitability* of the speaker's style, which they said would assess naturalness. As we will see from the results of experiment one, changing just one word in the task instructions can lead to different ratings. Many studies have used instructions such as *suitability* and *appropriateness* as synonyms for naturalness when they are in fact asking something quite different [8, 12].

Another option is a preference test to determine which system or prosodic realisation listeners prefer. For example, Aubin et al. tested the difference between a TTS system using discourse relations and a baseline system, using a preference test in which the target sentence was presented in a natural speech context [5]. Oplustil et al. also used a preference test in order to evaluate whether systems which take acoustic context into account from the preceding sentence perform better than a non-context-aware baseline [3]. While preference tests and MUSHRA both ask participants to make *direct comparisons* of stimuli with differing prosodic renditions, MOS tests do not. By asking listeners to provide 'absolute' ratings, many stimuli could receive the same MOS score.

Clark et al. [8] were the first to systematically evaluate the use of MOS for long-form evaluation. They compared differences in MOS ratings for utterances presented in isolation, in a context-target pair, or in a paragraph. They asked participants to rate the *naturalness* of utterances presented in isolation, but for context-target pairs, they asked participants to rate *appropriateness* of the target utterance given the context. The type of context was also varied, being either text, synthetic speech, or natural speech. They found that target utterances presented in context were rated significantly higher than the same utterances presented in isolation, when the context was in the form of text or synthetic speech. It is important to re-iterate that the synthetic speech was *not* context-dependent.

Clark et al. postulated that the increase in rating might be due to the task specification, and indeed other work has found that instructions can have an impact on MOS rating [14]. They also suggested that this may be due to 'the fact that the content of a paragraph non-initial sentence sounds less natural when presented out of context.' [8, Section 5.1]. They found no increase in ratings when the preceding context utterance was (non-vocoded) natural speech, reasoning that mismatches in quality between natural and synthetic speech make the synthetic speech sound of lower quality.

In the study reported in this paper, we focus exclusively on the MOS paradigm and investigate what factors lead to differing MOS scores between utterances presented in isolation vs. in context. Clark et al. used different wording of instructions when presenting isolated utterances than when presenting them in context. One of our experiments investigates the effect of wording alone, to avoid this confound. We restrict the investigation to the case of both target and context being synthetic speech. We also investigate whether the paradigm is sensitive enough to differentiate prosodically-different renditions of a sentence by a single system, something that Clark et al. did not do.

3. Research Questions

3.1. Effect of instructions

As noted in [8], the increase in MOS rating between the isolated condition and the TTS context condition was rather unexpected, given that the TTS model in question was not context-aware. One factor that might have influenced MOS was the task specification. Specifically, participants were asked to rate the *naturalness* of isolated utterances but the *appropriateness* of utterances presented in context. By wording the instructions to ask for either naturalness or appropriateness ratings, our first experiment tests whether this difference leads to changes in rating, independent of how the stimuli are presented.

3.2. Effect of between-sentence textual context dependency

Although [8] suggested that the increase in MOS rating may have been due to the task, they also suggested that utterances from non-paragraph-initial position may benefit from being presented with a preceding context. This is because non-initial sentences more often contain anaphoric references, such as pronouns, and therefore need a context in order to be fully understood. In experiment two, we manipulate the context-dependency of the target sentence text to test whether sentences containing anaphora receive higher MOS ratings when presented in a context that provides the referent, than non-anaphoric versions that do not need context in order to be fully understood.

3.3. Sensitivity of MOS to prosodic differences

While [8] investigated the effect of synthetic spoken context, natural spoken context and text context, they did not investigate whether participants are sensitive to changes in prosodic realisation when both context and target are synthetic and differ only in their prosody. [12] suggests that rating speech in context is difficult because there is no *correct* realisation and multiple variations will be equally acceptable. Therefore, in experiment three, we make one stimulus obviously non-canonical and ill-fitting to the context, in order to evaluate whether such a mismatch is salient for participants. If participants rate both the non-canonical and canonical highly in context, that would be evidence that this task is ill-suited to evaluating prosodic variation.

4. Methods

4.1. Data and models

We used the LJ Speech corpus, which consists of roughly 13 000 sentences read by a female speaker [15], for training all models. The model used in all experiments was the Ophelia implementation [16] of DC-TTS [17] . For experiment 3, we needed to manipulate prosody. We used the publicly-available training data used in [18] which is the LJ Speech corpus marked up with prosodic labels automatically generated using continuous wavelet transform (CWT) features which correlate with prosodic attributes such as prominence and boundaries. By marking up the training data with these labels, we obtained a model that offered control over prosody during inference, simply by changing the labels. Suni et al. used three strength levels of both accent and boundary labels, with accent level 0 signifying a de-accented word and boundary level 0 signifying no prosodic boundary. Level 2 accent signifies an emphasised word and level 2 boundary is roughly equivalent to an intona-

	Condition	
	Context-dependent	Context-independent
Context	**Storms** have been named in the US since the 1700s, for the UK it's a relatively new thing.	**Storms** have been named in the US since the 1700s, for the UK it's a relatively new thing.
Target	The first **one** to receive a name in the UK was storm Abigail in 2015.	The first **storm** to receive a name in the UK was storm Abigail in 2015.

Table 1: *Example of context-dependent (left column) and context-independent (right column) sentence pairs.*

Please, read the instructions carefully:
- You will be presented with **one sentence at a time**.
- We want you to rate how **natural** the sentence **sounds**.

(a) *Rating naturalness of utterances presented in isolation*

Please, read the instructions carefully:
- You will listen to **two sentences**.
- The second sentence will be highlighted in **bold text**.
- We want you to rate how **natural** the second sentence *sounds*, given the first sentence.

(b) *Rating naturalness of target utterances presented in context*

Please, read the instructions carefully:
- You will listen to **two sentences**.
- The second sentence will be highlighted in **bold text**.
- We want you to rate how **appropriate** the second sentence *sounds*, given the first sentence.

(c) *Rating appropriateness of target utterances presented in context*

Table 2: *Participant instructions.*

tional phrase boundary [18]. The LJ Speech recordings contain some background noise and reverberation, which we mitigated by post-processing all generated synthetic speech with the Automatic Sound Engineer (ASE) [19].

4.2. Stimuli

We created 110 pairs[1] of sentences each comprising a context sentence followed by a target sentence, using facts from Wikipedia. An example of two context-target pairs is given in Table 1. The same sentences were used in all experiments. We did not create test material using held out utterances from LJ Speech because this was too restrictive for carefully crafting suitable sentence pairs. All sentences were phonetised using [20] manually corrected, then synthesised.

Stimuli comprising a context-target pair were created by synthesising the two sentences separately then concatenating them into a single audio file separated by a 400 ms pause, a duration chosen through informal listening. This differs from [8], who asked listeners to click separate buttons to play context and target utterances.

4.2.1. Text manipulation

Each stimulus is the synthesised speech of a context sentence followed by one of two possible sentences: either the context-dependent follow-up (CD) or context-independent follow-up (CI). Table 1 provides an example. The CD target sentence needs the context sentence for the listener to resolve the anaphoric reference, such as *it* or *they*. In the CI condition, the target sentence has the referent filled in. The only difference between CI and CD conditions is the referent. Any two-word referents were matched with a two-word anaphoric reference so that the number of words in both conditions is the same.

4.2.2. Prosodic manipulation

To achieve prosodic manipulation, we manually modified the CWT labels on the input to the TTS model in order to create a *canonical* and a *non-canonical* rendition of each target sentence. Non-canonical renditions (as judged by one of the authors) were created by changing the accent and phrase boundary structure of the target utterances such that accents were placed on unexpected words (e.g., function words) or placing prosodic phrase boundaries in unexpected places. Figure 1 provides an example: *first* is de-accented in the non-canonical renditions, but accented in the canonical renditions; *and* receives a strong emphasis in the non-canonical renditions, but is de-accented in the canonical renditions. The creation of prosodic variants was constrained by the ability of the model, which did not render intelligible speech for every possible combination of accents and boundaries.

4.3. Participants

Listeners who self-reported to have no hearing impairment, be resident in the United States and have English as their first language were recruited through Prolific.[2] No other demographic information was asked for. None were allowed to participate more than once within this study. They received monetary compensation for taking part. Participants were asked whether they were using headphones. The responses from anyone who answered *no* were removed from analysis, following [8], as were those from participants who took less than 10 minutes (the minimum time required to listen to all stimuli).

4.4. MOS task

We implemented the MOS task in Qualtrics.[3] Following [8], participants were asked to rate stimuli on a scale of 1-5 in 0.5 increments (i.e., a 9-point scale). Points 1 to 5 were labelled as *poor*, *bad*, *fair*, *good* and *excellent*.

4.4.1. Experiment 1 - Effect of instructions

Each participant was assigned to one of 3 conditions. All participants in any given condition rated the same stimuli.

[1] Stimuli can be found: `https://johannahom.github.io/SSW-samples/index.html`

[2] https://www.prolific.co

[3] https://www.qualtrics.com/

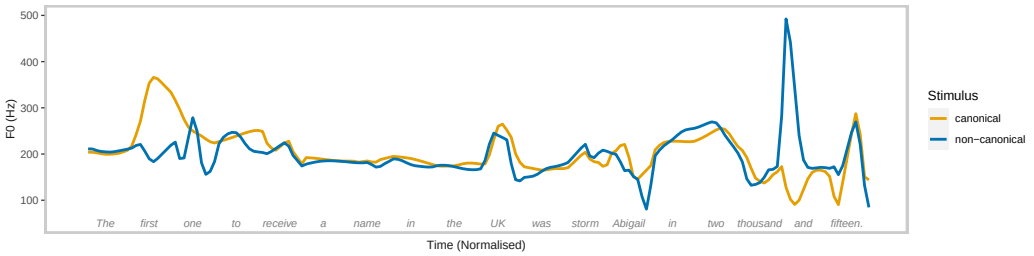


Figure 1: *Time-normalised F0 contour of a canonical and non-canonical stimulus from experiment 3.*

Condition 1: each participant was given the instructions in Table 2a, then rated 110 isolated sentences comprising all 55 unique context sentences, and 55 target sentences (a mixture of CI and CD) presented in randomised order. Condition 2: each participant was given the instructions in Table 2b then rated 55 context-target pairs presented in a random order. Condition 3: identical to condition 2, except using the instructions in Table 2c.

4.4.2. Experiment 2 - Effect of between-sentence textual context dependency

Each participant rated one of 4 sets of stimuli: Set 1: each participant was given the instructions in Table 2a, then rated 110 isolated sentences comprising all 55 unique context sentences, and 55 target sentences (a mixture of CI and CD) presented in randomised order. (*Since this is identical to experiment 1 condition 1, the same participant responses were re-used.*) Set 2: identical to set 1, and also using all 55 unique context sentences, except now using the remaining 55 target sentences not presented in condition 1 (also a mixture of CI and CD), to counterbalance. Set 3: each participant was given the instructions in Table 2c then rated all 55 context-target pairs presented in a random order. (*Since this is identical to experiment 1 condition 3, the same participant responses were re-used.*) Set 4: identical to set 3, except using the remaining 55 sentence pairs not presented in set 3, to counterbalance.

4.4.3. Experiment 3 - Sensitivity of MOS to prosodic differences

Each participant rated one of 4 sets of stimuli: Set 1: each participant was given the instructions in Table 2a, then rated 110 isolated sentences comprising all 55 unique context sentences rendered canonically, and 55 target sentences of which around half were rendered canonically and the rest rendered non-canonically, all presented in randomised order. Set 2: identical to set 1, with the same canonical renditions of all 55 unique context sentences, except with the canonical vs. non-canonical renditions of the target sentences swapped, to counterbalance. Set 3: each participant was given the instructions in Table 2c then rated 55 context-target pairs presented in a random order. Context sentences were always rendered canonically. Around half the target sentences were rendered canonically and the rest rendered non-canonically. Set 4: identical to set 3, except with the canonical vs. non-canonical renditions of the target sentences swapped, to counterbalance.

5. Results

All analyses were done in a by-items fashion such that, for each stimulus, the MOS rating is the mean of all participants' ratings for that stimulus. All data were found to be normally distributed following an insignificant Sharpiro-Wilk test and we therefore used two-tailed paired t-tests. Whenever making multiple pairwise comparisons, p-values were adjusted with Bonferroni coefficients.

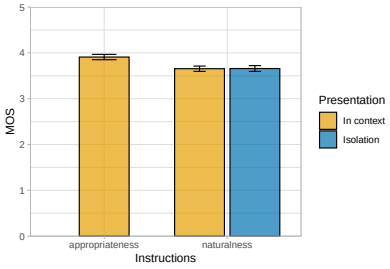


Figure 2: *Results for experiment one: MOS ratings of appropriateness and naturalness for utterances presented in isolation and in context.*

5.1. Experiment one

The experiment tests whether the instruction to listeners affects their ratings. A total of 108 participants took part of which 8 (7.4%) were removed using exclusion criteria from Section 4.3. As we see in Figure 2, stimuli were rated lower on the 5-point MOS scale when presented in isolation (M = 3.66 SD = 0.239) than in context. However this is only the case when using the instructions in Table 2c which asked them to rate how *appropriate* they sounded in context (M = 3.91 SD = 0.220) but *not* when using the instructions in Table 2b which asked them to rate how *natural* they sounded in context (M = 3.65 SD = 0.219). Ratings obtained with the 'how appropriate' instructions were significantly higher than those obtained with the 'how natural' instructions: t(54) = 9.94, $p < 0.001$. When using the 'how natural' instructions, there is no significant difference in ratings for stimuli presented in isolation vs. in context: t(54) = -0.16, $p = 1$. This refutes Clark et al.'s [8] hypothesis that it is the quality of the context and the match in quality (i.e., both context and target are synthetic speech) which leads to an increase in MOS rating.

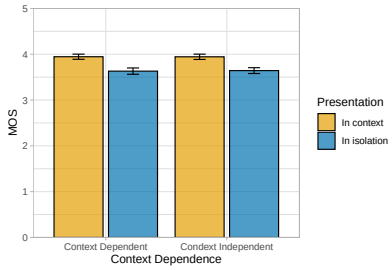


Figure 3: *Results for experiment two: MOS ratings for context-dependent and context-independent utterances presented in isolation and in context.*

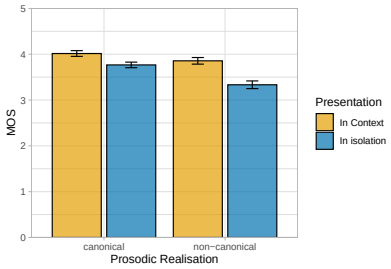


Figure 4: *Results for experiment three: MOS ratings for prosodically canonical and non-canonical renditions, presented in isolation and in context.*

A better explanation, also mentioned in [8], is that differences in ratings arise because participants interpret 'appropriate' differently to 'natural'. This implies that, in the condition from [8] where a synthetic utterance is presented after a natural spoken context utterance, listeners were rating the target as less *appropriate* rather than less natural: it is not appropriate for speech to change from natural to synthetic. We conclude that asking for ratings of *appropriateness* is different to asking for ratings of *naturalness*, for stimuli presented in context.

5.2. Experiment two

This experiment tests whether ratings of appropriateness are affected by textual dependence between the target and its context. A total of 144 participants took part of which 10 (6.9%) were removed using the exclusion criteria in Section 4.3. Results are shown in Figure 3. First, for utterances presented in isolation, there is no significant difference in ratings of naturalness for context-dependent (M = 3.63 SD = 0.262) and context-independent (M = 3.64 SD = 0.240) sentences (t(54) = -0.34, $p = 1$). When rated in context, there is no significant difference in ratings of appropriateness between context-dependent (M = 3.95 SD = 0.212) and context-independent utterances (M = 3.94 SD = 0.219): t(54) = 0.048, $p = 1$. Finally, consistent with the results from experiment 1, there is a significant difference between ratings of isolated utterances and utterances presented in context. This is true regardless of whether the utterance is context-dependent or is context-independent: t(54) = -8.30, $p < 0.001$ and t(54) = -10.48, $p < 0.001$ respectively. We conclude that textual context dependence does not affect listeners' ratings. However, as in experiment one, ratings of appropriateness for utterances presented in context are higher than ratings of naturalness for utterances presented in isolation.

5.3. Experiment three

This experiment tests whether MOS rating is sensitive to differences in prosodic realisation. A total of 144 participants took part of which 13 (9.0%) were removed using the exclusion criteria in Section 4.3. Results are shown in Figure 4. When presented in isolation, naturalness ratings of non-canonical renditions (M = 3.33, SD = 0.318) were significantly lower than of canonical renditions (M = 3.77 , SD = 0.231), t(54) = 9.41, $p < 0.0001$. This also holds true when these stimuli were presented in context and rated for appropriateness, although ratings of non-canonical (M= 3.86 SD= 0.273) and canonical (M = 4.02 SD = 0.237) are closer: t(54) = 3.86, $p = 0.001$. Both canonical renditions and non-canonical renditions received higher appropriateness ratings when presented in context than naturalness ratings when presented in isolation: t(54) = -7.29, $p < 0.001$ and t(54) = -18.33, $p < 0.001$ respectively. This is consistent with the findings reported in [8] and our results in experiments one and two. We conclude that MOS is sensitive enough to measure prosodic differences. As in experiments one and two, we once again conclude that appropriateness ratings for utterances presented in context are higher than naturalness ratings for utterances presented in isolation.

6. Discussion

Like Clark et al. [8], we found that utterances presented in context receive higher ratings of appropriateness than when presented in isolation, across all three experiments. In experiment one, we concluded asking whether an utterance sounds *appropriate* in context is not the same as asking whether it sounds *natural*. We believe the boost in rating is caused by the task specification, as Clark et al. suggested. This could be because the term *appropriate* is open to interpretation by listeners as textual appropriateness or prosodic appropriateness.

We tested whether context dependent targets received a boost in rating when their context was provided. The results from experiment two suggest this is not the case: this context-dependency of text does not play a significant role in listeners' ratings. This does not mean that participants were not taking the text into account at all. All our sentence pairs (an example is in Table 1) fitted together contextually, whether the target contained anaphoric reference or not: so all target sentences were appropriate in context, and listeners' ratings may reflect that. Of course, if they were *only* rating the text, we would expect the same high MOS across all stimuli, which was not the case: the speech did also matter. A future experiment could manipulate semantic or syntactic mismatch between context and target.

In experiment three, we tested whether MOS is sufficiently sensitive to measure differences in prosodic realisation. Clark et al [8] showed that varying the contexts between natural speech, synthetic speech and just text led to changes in MOS rating. They postulated that this was due to quality mismatches, with natural speech lowering the perceived quality of the following synthetic target. Our experiments exclusively used synthetic speech and did not vary the context utterance, so we can rule out any effects caused by differing contexts. We found that participants rated prosodically non-canonical targets as significantly less natural in isolation than canonical targets: so MOS

is sensitive to the difference. Our stimuli generally had substantial prosodic differences (the non-canonical renditions were very different to the canonical ones), so we are unable to say whether MOS would be sensitive to more subtle differences.

But, unexpectedly, *both* non-canonical and canonical target utterance received significantly higher ratings for appropriateness when presented in context, than identical utterances presented in isolation and rated for naturalness. Sometimes, a non-canonical form may indeed sound unnatural if heard in isolation, unless a very specific context is provided in which it sounds felicitous. Our stimuli, however, were constructed to ensure that the non-canonical renditions were *infelicitous* to their contexts, which is why we did not expect ratings of appropriateness to still be higher.

We would also like to extend this study to implicit measures of speech processing, such as reaction time word monitoring tasks, for example to evaluate which prosodic realisation leads fastest processing.

7. Conclusions

We replicated the most interesting finding in [8]: that synthetic speech is rated more highly in context. We investigated the source of this effect, considering the instructions to listeners, textual context-dependence and prosodic felicity. We found that the wording of instructions had a significant effect on the final MOS score. Instructions that asked listeners to rate *naturalness* resulted in the same rating regardless of whether utterances were presented in isolation or in context. In contrast, asking listeners to rate *appropriateness* of utterances presented in context resulted in a rating higher than the naturalness score, as in [8]. Naturalness and appropriateness are fundamentally different things. It is important, when reporting listening test results, to also report the exact wording of instructions to listeners.

To understand how listeners are interpreting appropriateness, we manipulated the target sentence text. We found no significant difference in the ratings of context-dependent and context-independent text. This does not mean that text plays no role in appropriateness rating. Future research could manipulate semantic and syntactic factors to gain a better understanding.

We investigated whether MOS is sensitive to prosody, which will be the main difference between the output of a context-aware model and a context-independent one. We found that, for utterances presented in isolation, participants exhibited a greater preference for canonical renditions, a preference that was maintained for utterances presented in context. MOS is an appropriate paradigm for evaluating prosodic differences. This increase in MOS was also found for non-canonical items, although they were constructed to be less felicitous in context. It is therefore still unclear what is exactly taken into account in the appropriateness rating. We would like to extend this work by including other variations in the instructions to participants, such as attempting to focus their attention on prosody.

Acknowledgements: we thank Rob Clark for providing additional details about the work in [8]. This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 859588 and was supported in part by: ANID, Becas Chile, nº 72190135.

8. References

[1] H. Guo, S. Zhang, F. K. Soong, L. He, and L. Xie, "Conversational End-to-End TTS for Voice Agents," in *IEEE Spoken Language Technology Workshop (SLT)*, vol. 2, 2021, pp. 403–409.

[2] G. Xu, W. Song, Z. Zhang, C. Zhang, X. He, and B. Zhou, "Improving Prosody Modelling with Cross-Utterance BERT Embeddings for End-to-end Speech Synthesis," 2020, pp. 2–6. [Online]. Available: http://arxiv.org/abs/2011.05161

[3] P. Oplustil-Gallegos and S. King, "Using previous acoustic context to improve Text-to-Speech synthesis," in *arXiv preprint*, arXiv:2012.03763, 2020.

[4] P. Oplustil-Gallegos, J. O'Mahony, and S. King, "Comparing acoustic and textual representations of previous linguistic context for improving Text-to-Speech," in *SSW 2021*, 2021.

[5] A. Aubin, A. Cervone, O. Watts, and S. King, "Improving speech synthesis with discourse relations," in *Interspeech*. Graz: ISCA, 2019, pp. 4470–4474.

[6] M. Farrús, C. Lai, and J. D. Moore, "Paragraph-based prosodic cues for speech synthesis applications," in *Proceedings of the International Conference on Speech Prosody*, 2016, pp. 1143–1147.

[7] S. Prevost and M. Steedman, "Specifying intonation from context for speech synthesis," *Speech Communication*, vol. 15, no. 1, pp. 139–153, 1994.

[8] R. Clark, H. Silen, T. Kenter, and R. Leith, "Evaluating Long-form Text-to-Speech: Comparing the Ratings of Sentences and Paragraphs Rob," in *The 10th ISCA Speech Synthesis Workshop*, Vienna, 2019.

[9] P. Wagner, J. Beskow, S. Betz, J. Edlund, J. Gustafson, G. Eje Henter, S. Le Maguer, Z. Malisz, E. Szekely, C. Tannander, and J. Vosse, "Speech Synthesis Evaluation — State-of-the-Art Assessment and Suggestion for a Novel Research Program," in *The 10th ISCA Speech Synthesis Workshop*, 2019, pp. 105–110.

[10] Z. Hodari, C. Lai, and S. King, "Perception of prosodic variation for speech synthesis using an unsupervised discrete representation of F0," in *Proceedings of the International Conference on Speech Prosody*, 2020, pp. 965–969.

[11] V. Klimkov, A. Nadolski, A. Moinet, B. Putrycz, R. Barra-Chicote, T. Merritt, and T. Drugman, "Phrase break prediction for long-form reading TTS: Exploiting text structure information," in *Interspeech*. Stockholm: ISCA, 2017, pp. 1064–1068.

[12] S. Tyagi, M. Nicolis, J. Rohnke, T. Drugman, and J. Lorenzo-Trueba, "Dynamic prosody generation for speech synthesis using linguistics-driven acoustic embedding selection," in *Interspeech*, H. Meng, B. Xu, and T. F. Zheng, Eds. Shanghai: ISCA, 2020, pp. 4407–4411.

[13] A. K. Syrdal and J. T. McGory, "Inter-transcriber reliability of toBI prosodic labeling," in *Sixth International Conference on Spoken Language Processing*. Beijing, China,: ISCA, 2000, pp. 235–238.

[14] R. Dall, J. Yamagishi, and S. King, "Rating naturalness in speech Synthesis: The effect of style and expectation," *Proceedings of the International Conference on Speech Prosody*, pp. 1012–1016, 2014.

[15] I. Keith and J. Linda, "The LJ Speech Dataset," 2017. [Online]. Available: https://keithito.com/LJ-Speech-Dataset/

[16] CSTR-Edinburgh, "Ophelia," 2018. [Online]. Available: https://github.com/CSTR-Edinburgh/ophelia

[17] H. Tachibana, K. Uenoyama, and S. Aihara, "Efficiently Trainable Text-to-Speech System Based on Deep Convolutional Networks with Guided Attention," in *IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP*, 2018, pp. 4784–4788.

[18] A. Suni, S. Kakouros, M. Vainio, and J. Šimko, "Prosodic Prominence and Boundaries in Sequence-to-Sequence Speech Synthesis," in *Proceedings of the International Conference on Speech Prosody*, Tokyo, 2020, pp. 940–944.

[19] C. Chermaz and S. King, "A sound engineering approach to near end listening enhancement," in *Interspeech*. Shanghai: ISCA, 2020, pp. 1356–1360.

[20] K. Park and J. Kim, "g2pE," 2019. [Online]. Available: https://github.com/Kyubyong/g2p

11th ISCA Speech Synthesis Workshop (SSW 11)
26-28 August 2021, Budapest, Hungary

Non-native English lexicon creation for bilingual speech synthesis

Arun Baby, Pranav Jawale, Saranya Vinnaitherthan, Sumukh Badam, Nagaraj Adiga, Sharath Adavanne

Zapr Media Labs (Red Brick Lane Marketing Solutions Pvt. Ltd.), India

arun.baby@zapr.in

Abstract

Bilingual English speakers speak English as one of their languages. Their English is of a non-native kind, and their conversations are of a code-mixed fashion. The intelligibility of a bilingual text-to-speech (TTS) system for such non-native English speakers depends on a lexicon that captures the phoneme sequence used by non-native speakers. However, due to the lack of non-native English lexicon, existing bilingual TTS systems employ native English lexicons that are widely available, in addition to their native language lexicon. Due to the inconsistency between the non-native English pronunciation in the audio and native English lexicon in the text, the intelligibility of synthesized speech in such TTS systems is significantly reduced.

This paper is motivated by the knowledge that the native language of the speaker highly influences non-native English pronunciation. We propose a generic approach to obtain rules based on letter to phoneme alignment to map native English lexicon to their non-native version. The effectiveness of such mapping is studied by comparing bilingual (Indian English and Hindi) TTS systems trained with and without the proposed rules. The subjective evaluation shows that the bilingual TTS system trained with the proposed non-native English lexicon rules obtains a 6% absolute improvement in preference.

Index Terms: Bilingual speech synthesis, non-native English, L2 English, lexicon creation, Common phones

1. Introduction

Developing a bilingual text-to-speech (TTS) system [1] is necessary for countries like India where the majority of the population speak more than one language. Generally, this population speaks their native language as the first and English as their second language. The pronunciation of English words by a non-native speaker is strongly influenced by their native language and is most often different from the native English pronunciation [2]. Indian languages, which have a high grapheme to phoneme correlation (phonemic language), derive pronunciation directly from the spellings of the word. On the contrary, English is an alphabetic and highly non-phonemic language. Hence native phonemic language speakers whose pronunciation is influenced by the spelling of the word often pronounce English words differently from native English speakers. This mispronunciation is further enhanced for native speakers from languages whose phonemes are different from the English language. These speakers generally replace the English phoneme with the closest phoneme in their native language. Given these challenges, building a TTS system for such non-native English bilingual speakers requires a lexicon that handles the influence of the first language on the native English lexicon. However, due to the lack of availability of such a non-native English lexicon [3], existing bilingual TTS systems employ widely available native English lexicon, in addition to their native language lexicon, which results in reduced intelligibility and heavily accented synthesized speech.

Yarra *et al.* [4] proposed to collect Indian English lexicon from the mispronunciations of Indian speakers recording sentences from the TIMIT database. Multiple phonemes- and letter-specific context rules were manually identified through observing pronunciation variations between Indian and original TIMIT speakers. The proposed lexicon was evaluated on the speech recognition task and shown to improve the overall recognition. However, collecting such parallel data for a large vocabulary and manually finding the mispronunciation between the two is a tedious and expensive task. Anju *et al.* [5] proposed a transliteration-based method for developing speech synthesizers for Bilingual Indian English. A mapping was obtained between native English CMU dictionary phonemes and the Indian common phone Label Set (CLS) [6] shared across the Indian languages. The stress markers of the CMU phoneset were removed before this mapping. Further, the words not part of the CMU dictionary were first transliterated to a phonemic Indian language. The CLS phoneme sequence was obtained using the unified parser [7] grapheme to phoneme (G2P) model. In [8], the phoneme sequence of the CMU dictionary was manually corrected for the Assamese language accent to develop an Assamese English TTS system. However, the manual effort of editing the lexicon is an expensive and time-consuming operation. In [9], a sequence labelling approach is employed to generate a pronunciation dictionary using Conditional Random Fields (CRFs). However, this method needs a substantial parallel corpus of phone sequence mapping between native and non-native lexicon to train the CRF network.

In this paper, we propose a generic framework to obtain the rules for mapping the phone sequence of a native English lexicon to a non-native one. Specifically, the framework aims to derive non-native English pronunciation for speakers from native languages that follow phonemic orthography. Although we study the framework on native Hindi language speaker, the framework itself should be adaptable to speakers of other phonemic languages. As the first step, we identify a subset of highly frequent English words. For these words, a three-way alignment is obtained between a) the English letter sequence, b) the CMU phoneme sequence from native English CMU dictionary, and c) the CLS phoneme sequence obtained using unified parser [7] from the (manually curated) transliterated version of the English word. Repeating patterns of the aligned triplets of letter-CMU-CLS phonemes that produce mispronunciations are manually identified. After that, rules are devised which, wherever relevant, will substitute an original CMU phoneme in a word pronunciation with a new phoneme that produces the correct non-native English pronunciation. These rules are applied on the entire native English lexicon to obtain the corresponding non-Native English lexicon. We show that the proposed framework provides better non-native English pronunciation than the

10.21437/SSW.2021-27

existing frameworks through subjective listening tests.

2. Proposed approach

The proposed approach is studied on a bilingual Indian English-Hindi dataset [10]. This dataset contains monolingual recordings in English and Hindi from the same Male speaker (More about the dataset in Section 3.1). The English transcripts are in Roman script, while Hindi is in Devanagari script. The pronunciations for English words were obtained from g2p_en[1] G2P model which uses the 39 CMU phoneset. Similarly, the Unified Parser [7] was employed to obtain the pronunciations for the Hindi words, that used the 59 CLS phonemes, amounting to 98 phonemes in total for the bilingual dataset. Further, motivated by the works of [5], we merged acoustically similar CMU and CLS phonemes [11]. This merging is intended to address non-native English speakers substituting similar phonemes from their native language in place for native English phonemes. Additionally, reduced phoneset results in increased training data per phoneme, and consequently better phoneme modelling. This reduced phoneset has 73 phonemes and is referred to "EngHinCommon" hereafter. While merging, care was taken not to merge phonemes whose realization is audibly distinct across Hindi and English languages. The merged CMU and CLS phonemes are listed in Table 1.

As the baseline for the proposed approach, we trained separate bilingual TTS systems with the above 98 CMU and CLS phoneset (Model 1 in Section 3) and the reduced EngHinCommon phoneset (Model 2 in Section 3). The results showed that the quality of synthesis of Hindi words remained consistent between the two models. However, for the English words, contrary to the motivation of employing the reduced phoneset, the non-native English pronunciation improved only for a few English words compared to Model 1. More details of these experiments are discussed in Section 3.4.3. As discussed earlier, a non-native English speaker (whose native language follows phonemic orthography) takes a cue from the spelling of the word for pronunciation. For example, The CMU phone sequence for word /CITED/ is /S AY T **AH** D/, whereas a phonemic language speaker, takes a cue from the spelling and pronounces it as /S AY T **EH** D/. However, similar to EngHinCommon phoneset creation, a universal mapping of all /AH/ to /EH/ will create more problems than it solves. We need a nuanced approach that goes beyond one-to-one phoneme mapping. Motivated by this, in the following section, we explain the proposed approach to map phonemes based on additional information such as English letter identity, position within a syllable, and letter context.

2.1. Selection of words

The bilingual dataset has a vocabulary of about 7 k words, of these, around 6.5 k words are English words that are not proper nouns of Indian origin. A subset of 2 k words from 6.5 k words of the dataset is chosen based on their being present in the top 10 k most frequent words in an independently obtained Indian newspaper text archive. It is crucial to select a large number of words to arrive at accurate and exhaustive rules.

This paper does not study proper nouns of Indian origin because the native English G2P models fail on these words, making it impossible to recover from these errors even after using a rule-based phoneme mapping. The best way to derive pronunciation for such Indian origin proper nouns is to transliterate them into a phonemic language script and obtain the phoneme sequence using a G2P model like Unified Parser [7].

[1]https://pypi.org/project/g2p-en/

Serial No.	English Word Example	CMU	EngHinCommon	CLS	Hindi Word Example
1	b**a**lm	AA	**aa**	aa	काम (k**ā**m)
2	b**a**t	AE	**ae**	ae	कैद (k**ae**d)
3	st**a**lk	AO	**ax**	ax	कॉल (k**au**l)
4	b**i**t	IH	**i**	i	इमली (**i**mlī)
5	b**oa**t	OW	**ou**	ou	कौन (k**au**n)
6	b**oo**k	UH	**u**	u	तुम (t**u**m)
7	**b**uy	B	**b**	b	बाल (**b**āl)
8	**ch**ina	CH	**c**	c	चम्मच (**c**amma**c**)
9	**d**ie	D	**dx**	dx	डर (**ḍ**ar)
10	**th**y	DH	**DH**	d	दवा (**d**avā)
11	**f**ight	F	**f**	f	तरफ़ (tara**f**)
12	**g**uy	G	**g**	g	गाना (**g**ānā)
13	**h**igh	H	**h**	h	हाथ (**h**āth)
14	**j**ive	JH	**j**	j	जेब (**j**eb)
15	**k**ite	K	**k**	k	कल (**k**al)
16	**l**ie	L	**l**	l	लिखना (**l**ikhanā)
17	**m**y	M	**m**	m	मां (**m**āṃ)
18	**n**igh	N	**n**	n	नाक (**n**āk)
19	si**ng**	NG	**NG**	ng	सिंग (si**ng**)
20	**p**ie	P	**p**	p	पतंग (**p**atang)
21	**r**ye	R	**r**	r	रात (**r**āt)
22	**s**igh	S	**s**	s	सूरज (**s**ūraj)
23	**sh**y	SH	**sx**	sx	शादी (**sh**ādī)
24	**t**ie	T	**tx**	tx	टमाटर (**ṭ**amā**ṭ**ar)
25	**th**igh	TH	**th**	th	थका (**th**akā)
26	**y**acht	Y	**y**	y	यादें (**y**ādein)

Table 1: *Subset of EngHinCommon phonemes that are obtained by merging CMU and CLS phoneset. Rest of the CMU and CLS phonemes are used as unique phonemes of EngHinCommon.*

2.2. Process of deriving rules

The process of deriving phoneme mapping rules from the above subset of 2 k English words is described in Figure 1. To obtain the CLS phoneme sequence for these English words as pronounced by native Hindi speakers we used an online English-Hindi Dictionary[2], which has Devanagari transliteration for several of these words. Four native Hindi speakers conducted a manual verification to match the transliteration with the actual pronunciation of native Hindi speakers. To avoid any bias toward the speaker in the training dataset, the verification of transliteration was done purely based on text. Finally, the CLS phoneme sequences were obtained from these transliterations using the Unified Parser [7]. We employ the transliterations only for the creation of rules. Once the rules are created, they can be directly applied to any native English lexicon to obtain their non-native English versions without the requirement of any transliteration.

Next, we used an alignment algorithm (explained in section 2.3) to align the English letters in the words, the CMU phonemes for the word, and the target CLS phonemes. While in a large number of cases, source CMU phonemes are exclusively aligned with a single CLS phoneme. In some cases, the mapping was not exclusive, i.e., the same CMU phoneme was mapped to different CLS phonemes which are acoustically distant (e.g. /AH/ → /a/ in some words and /AH/ → /o/ in other

[2]https://dict.hinkhoj.com/english-to-hindi/

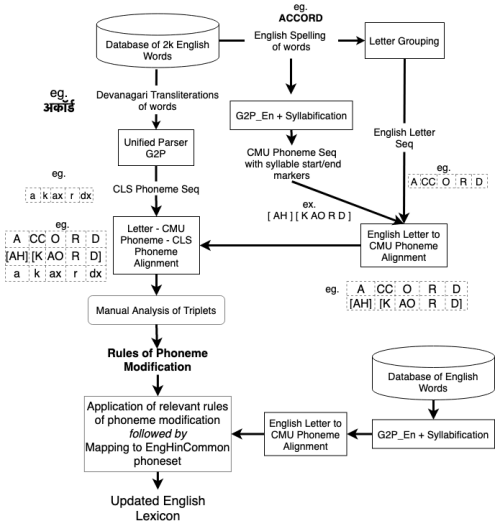


Figure 1: *Block diagram showing the process of deriving rules and the creation of non-native English lexicon.*

words). This ambiguity is the primary reason behind the inconsistency between native and non-native English lexicon. In such cases, we analyzed them further to devise rules for correcting the lexicons. These rules are based on knowledge of the source English letter, its position within a syllable, source CMU phone, and letter sequence context. We aim to propose rules that will reduce this phoneme mapping ambiguity and come up with conditions under which a particular target phoneme should be used (explained in the following sections). Finally, after correcting the lexicon using the proposed rules, all the ambiguous conditions discussed above are handled. After that, the remaining phonemes in the lexicon that are not yet modified with the proposed rules, are directly mapped to EngHinCommon phoneset as shown in Table 1.

2.3. Alignment Algorithm

One of the key steps of the proposed approach is to align the phoneme sequences with the letters. If the number of letters in English words were precisely the same as the number of phonemes, their alignment would be straightforward. However, this is not always the case. Hence we made use of **a.** Letter Grouping Heuristics and **b**. Heuristics for zero distance letter-phoneme pairs.

a. Letter Grouping: The aim of this grouping is to obtain units of letter sequences that usually correspond to a single phoneme. The following consecutive letters are always considered as a single unit - *ph*, *ch*, *ng*, *sh*, *th*, *er*, *ow*. Further, any duplicated consecutive non-vowel letters (Regex: $[\wedge aeiou]\{2\}$), and sequence of a single vowel letter followed by a vowel letter or letter *y* (Regex: $[aeiou][aeiouy]$), are also considered as single unit.

b. Heuristics for zero distance letter-phoneme pairs: Certain letter-phoneme pairs are considered as equivalent (eg. letter $p \leftrightarrow$ CMU phoneme P, letter unit *ph* $\leftrightarrow$ CMU phoneme F, letter $c \leftrightarrow$ CMU phoneme K, and so on). This equivalence information helps the dynamic programming-based alignment algorithm to output more accurate alignments with better causal correspondence between letters and phonemes.

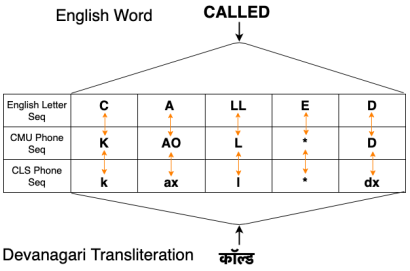


Figure 2: *The example illustrates the three-way alignment between English letters, CMU phone sequence, and CLS phone sequence obtained from the transliterated English word in Devanagari script*

Similar equivalence relation between certain CMU and CLS phoneme pairs is used to finally get a three-way alignment between English letter sequence $\leftrightarrow$ CMU phoneme sequence $\leftrightarrow$ CLS Phoneme Sequence. A sample illustration of our alignment algorithm is shown in Figure 2 for the word /CALLED/. Each column of the alignment in Figure 2 is hereafter referred to as a triplet.

2.4. Triplet Analysis

We analyse the triplets which contain non-exclusive CMU to CLS phone mapping to identify phoneme mapping patterns. For example, Figure 3 illustrates patterns where CMU phoneme /AA/ was aligned either with CLS phoneme /ax/ or with CLS phoneme /aa/ based on letter context present in triplets. We check if this knowledge about letters aligned with the CMU phoneme /AA/ can resolve this ambiguity. After going through a significant number of words resulting in such triplets, we develop a rule that applies to a large subset of words.

In some cases, to develop a specific rule, we have to refer to the position of the CMU phoneme within the syllable. The

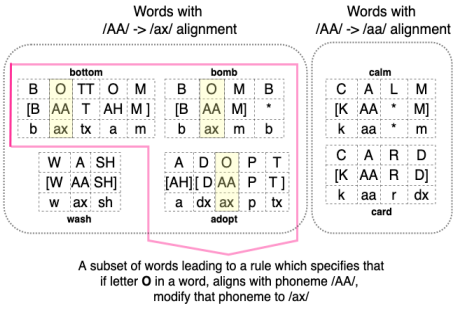


Figure 3: *Simplified illustration of triplet analysis used to derive the rule which states the condition under which we can modify phoneme /AA/ to phoneme /ax/*

Rules	Syll rule1							Syll rule2							Syll rule3								Syll rule4				
Word	academy							admonish							advocate								tibet				
Letter sequence	A	C	A	D	**E**	M	Y	A	D	M	**O**	N	I	SH	A	D	V	**O**	C	A	T	E	T	**I**	B	E	T
Current Map	[AH]	[K	AE]	[D	**AH]**	[M	IY]	[AE	D]	[M	**AA**	N	IH	SH]	[AE	D]	[V	**AH]**	[K	AH	T]	*	[T	**AH]**	[B	EH	T]
New Map	AH	K	AE	D	**i**	M	IY	AE	D	M	**ax**	N	IH	SH	AE	D	V	**o**	K	AH	T		T	**i**	B	EH	T
EngHinCommon	AH	k	ae	dx	**i**	m	IY	ae	dx	m	**ax**	n	i	sx	ae	dx	V	**o**	k	AH	tx		tx	**i**	b	EH	tx

Rules	Syll rule5								Seq rule1									Prefix rules							Suffix rules						
Word	ordinary								ambulance									**auto**mate							alert**ness**							
Letter sequence	O	R	D	**I**	N	A	R	Y	A	M	B	*****	**U**	**L**	A	N	C	E	**AU**	**T**	**O**	M	A	T	E	A	L	ER	T	**N**	**E**	**SS**
Current Map	[AO	R]	[D	**AH]**	[N	EH]	[R	IY]	AE	M	B	**Y**	**AH**	**L**	AH	N	S	*	**AO**	**T**	**AH**	M	EY	T	*	AH	L	ER	T	**N**	**AH**	**S**
New Map	AO	R	D	**i**	N	AH	R	IY	AE	M	B	*****	**u**	**l**	AH	N	S		**ax**	**tx**	**o**	M	EY	T		AH	L	ER	T	**N**	**EH**	**S**
EngHinCommon	ax	r	dx	**i**	n	AH	i	IY	ae	m	b		**u**	**l**	AH	n	s		**ax**	**tx**	**o**	m	EY	tx		AH	l	ER	tx	**n**	**EH**	**s**

Table 2: *Illustration of sample examples showing sub-set of rules and corrections at phone-level with triplet letter-CMU-CLS phoneme alignment. Modifications at the phoneme level are displayed in bold font.*

Table 3: *Summary of proposed rules to map native English lexicon to corresponding non-native version.*

	Source English Letter	Source Phoneme Sequence(CMU)	Target Phoneme Sequence(CLS)	Position in syllable	Corrected words in CMUdict (%)
Syll rule1	E	AH	i	end	1.9
Syll rule2	O	AA	ax	anywhere	6.4
Syll rule3	O	AH	o	end	2.7
Syll rule4	I	AH	i	end	2.5
Syll rule5	A	EH	AH	end	0.7
Seq rule1	(* U L)	(Y AH L)	(* u l)	anywhere	0.2
Prefix rules	sub word	old sequence	new sequence	NA	1
Suffix rules	sub word	old sequence	new sequence	NA	19.8

syllable boundaries are obtained using tsylb2 tool[3]. Syllable start and end boundaries within a CMU phone sequence are denoted using square brackets in Figure 1 and Figure 3. Apart from syllable level corrections, in some cases, the rule is dependent upon the letter context (letters to the left and right) and whether that letter sequence occurs within the word or at suffix/prefix locations. From the triplet analysis, it was observed that for most high-frequency suffixes and prefixes (e.g. *auto-*, *-ted*, *-less*, *-ness*, *-ment*), the Indian English speakers' pronunciation followed a consistent pattern. Therefore, additional rules for these suffixes and prefixes (40 rules) were identified from this analysis. A subset of these rules are depicted in Table 2 and Table 3.

In general, we have derived three kinds of rules – **1.** Syllable level rules, **2.** Letter sequence rules, and **3.** Prefix/Suffix-based rules. Finally, using these rules the native English lexicon is mapped to non-native English lexicon for the Indian English - Hindi bilingual speaker as shown in Figure 1. It is observed from these rules that most of the differences between native and Indian English pronunciation are among vowels, this finding is consistent with prior work [11, 12].

3. Experiments

3.1. Dataset

We selected a native Hindi Bilingual Male speaker from IndicTTS [10] database for our experimentation. This dataset consists of two parts, 9 hours each of monolingual Hindi and monolingual English dataset, amounting to 18 hours of studio-quality recording by a professional voice-over artist. The original recordings are at a 48 kHz sampling rate. However, all the studies in this paper are performed at 16 kHz. The genre of the spoken text is fiction and children's stories. The transcripts for English is in Roman, and Hindi is in Devanagri script.

3.2. TTS modeling

A neural network-based TTS system is used for all the experiments. A neural TTS system is generally comprised of a front end and a vocoder. As the fronted, we use the Tacotron2 (v3) recipe of ESPnet [13], which is an auto-regressive-based sequence-to-sequence model with a location-sensitive and guided attention mechanism [14]. The front end is trained with the phone sequence of the input text and the 80-band Mel spectrogram feature of the corresponding audio, computed with a 1024 point discrete Fourier transform and 256 sample hop-length. The front end is trained for 250 epochs with a batch size of 56 on 4 GPUs. The Mel spectrogram output of the front end is mapped to waveform using the parallel wavegan (PWG) vocoder [15]. The PWG vocoder is a non-autoregressive variant of the WaveNet [16] vocoder that has a significantly faster inference time. We use the publicly available implementation of PWG, whose code is accessible here[4].

3.3. Systems

We train three separate Tacotron2 models with different phone-sets to evaluate the efficacy of the proposed method while keeping the vocoder fixed across all our experiments.

3.3.1. Model 1

As the first baseline TTS model, we employ CMU phoneset (unstressed) for English data and CLS phoneset for Hindi. The unstressed version of CMU is used because Indian languages are syllable-timed and Indian speakers don't differentiate for different stress level [17]. Here a total of 98 unique phonemes (39 CMU + 59 CLS) are used for training the Tacotron model and develop bilingual TTS. Refer to Section 2 for more details.

3.3.2. Model 2

A second TTS model is trained to study the effect of using 73 phonemes EngHinCommon phoneset (Refer to Section 2 for more details) on the pronunciation of English words. Apart from the number of phonemes, all other model parameters are identical to Model 1.

[3]https://www.nist.gov/itl/iad/mig/tools

[4]https://github.com/kan-bayashi/ParallelWaveGAN

Figure 4: *Sample of subjective evaluation webpage employed to compare the pronunciation quality between two TTS models.*

3.3.3. Model 3

Finally, we train a TTS model with the proposed phone mapping rules as described in Section 2. After that, we map the exclusively aligned CMU and CLS phones as discussed in Section 2.2 to the EngHinCommon phoneset. The proposed rules correct around 40% of the training data. Model 3 has an identical phoneset as Model 2.

3.4. Evaluations

Three different subjective evaluations are performed to assess the above TTS models. The evaluation is restricted to isolated words instead of a sentence to avoid any influence on the evaluator's rating by other words in the sentence. During each evaluation, a subset of English words are synthesized using two of the above models, and a preference test between the two is carried out as shown in Figure 4. During the test, ten native Hindi speakers (also proficient in English) evaluate the synthesis of the models based on the closeness to the pronunciation of a bilingual Indian English-Hindi speaker. If the pronunciation of both the models were comparable, the listeners could choose the 'Both Good' option, on the other hand, if both the pronunciations were bad, they could choose the 'Both Bad' option. The synthesized recordings of the two models were shuffled and did not appear in the same order across the test. All evaluations are done with headphones to ensure clear perception, and the listeners were allowed to playback the audio any number of times. The synthesized recordings used for the subjective evaluations in our paper are available at https://www.zapr.in/ssw2021/samples.

3.4.1. Evaluation 1

As the first subjective evaluation, we compare Model 1 and Model 2 to study the effectiveness of the EngHinCommon phoneme mapping. The listeners rate a randomly chosen subset of 200 English words.

3.4.2. Evaluation 2

As the second subjective evaluation, Model 2 is compared against Model 3 to study the effectiveness of the proposed non-native English lexicon creation. For this evaluation, the listening test is carried out on three different subsets of 200 words: a) English words part of the training vocabulary, hereafter referred to as 'Dict' words. b) English words not in the training vocabulary, but the rules have been applied ('Rules'). c) English words that were neither part of the training vocabulary nor modified by any rules ('OOR': Out-of-rules).

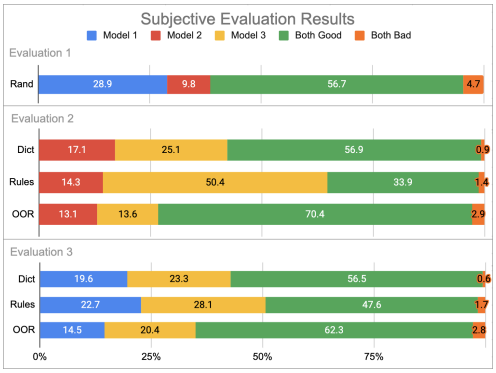


Figure 5: *Results of subjective evaluation indicating the listener's preference. The number indicates the percentage of words in a given evaluation set, falling in one of the categories – Words for which first model synthesis is preferred; Words for which second model synthesis is preferred; Words for which both the models are good; Words for which both the models are bad.*

3.4.3. Evaluation 3

Finally, to complete the comparisons between the three models, Model 3 is compared with the baseline Model 1. Similar to Evaluation 2, three comparisons are made with Dict words, Rules words, and OOR words, respectively.

4. Results and Discussion

All the proposed phoneme mapping rules in this paper directly influence the lexicon of only the English words. However, to study if this has affected the modelling of Hindi words during the bilingual TTS system training, we conducted listening tests on randomly chosen 200 Hindi words for the three Evaluations. We observed that the pronunciation was comparable across the three evaluations.

The summary of the three evaluations conducted on the English words are shown in Figure 5. In evaluation 1, for 28.9% of words the listeners preferred synthesis using the CMU and CLS phonemes, compared to only 9.8% of the words for which they preferred synthesis using EngHinCommon phoneset. This suggests that directly mapping individual CMU phonemes to similar-sounding CLS phonemes does not fix the pronunciation errors in non-Native English lexicon.

The three listening tests of evaluation 2 shown in Figure 5 prefer pronunciations of the proposed rules (Model 3) over the EngHinCommon phonemes (Model 2). The Model 3 preference is significantly higher for words not part of training vocabulary ('Rules'), achieving an absolute improvement of 35% over Model 2. This indicates the effectiveness of the proposed rules in correcting the native English lexicon for non-native English speakers. The words in OOR, which had not undergone any of these rules, also improved slightly. This might be a result of reduced confusion between lexicon and pronunciation in the recordings in Model 2. In the Dict words case as well listeners preferred Model 3 over Model 2 (25.1% vs. 17.1%). Ideally, since the 'dict' words are phonetically corrected for non-native English, Model 3 should have got 100% preference over

Model 2. However, since the lexicon corrections are not speaker specific, the inconsistency in the speaker's pronunciation with the lexicon can be the explanation for this result.

In evaluation 3, we compare the proposed phone rules (Model 3) with the baseline phones without any mapping (Model 1). From Figure 5 it is clear that the listeners preferred the proposed phone rules over the baseline across the three categories of words. On average the proposed rules obtain a 6% absolute improvement compared to the baseline.

A statistics of the percentage of the words undergoing each category of the proposed rules are shown in the last column of Table 3. We generated the statistics on the standard CMU dictionary of 130 k unique words. Here we see that around 35% of the words are getting corrected by the proposed rules (these numbers are excluding the EngHinCommon phoneme mapping). A category-wise statistics are also shown in that table. Further, the actual percentage of training corpus words that get changed depends on the frequency of these dictionary words in the corpus.

5. Conclusion

In this work, we have proposed a method to create a non-native English lexicon for Bilingual TTS. We came up with a systematic approach to do triplet analysis which helped uncover inconsistency between native and non-native English lexicons. The proposed method involved limited manual effort in the transliteration of English words from Roman script to Devanagari script. An additional manual effort was required for the analysis of the mismatch between the phoneme sequence of native and non-native pronunciation, to create multiple rules. Formulating these rules based on this mismatch doesn't require a phonetician's expertise. Finally, we showed that proposed non-native lexicon creation helped to improve the synthesis quality of bilingual TTS models with Indian-English and Hindi language pairs. This method can be easily applied to any phonetically orthographic language. Moreover, a reduced phoneset, as used in the proposed method, will improve the speech synthesis in low resource languages. With more data per phoneme, the trained model will be robust. While we have tried to come up with an exhaustive set of rules, some undiscovered rules may still be there, which can be considered as part of future work. The proposed lexicon generation approach can be extended to improve the performance of an Indian English Automatic Speech Recognition system.

6. References

[1] M. Chu, H. Peng, Y. Zhao, Z. Niu, and E. Chang, "Microsoft Mulan-a bilingual TTS system," in *Proc. ICASSP*, vol. 1. IEEE, 2003, pp. I–I.

[2] Y. Lee, S. Shon, and T. Kim, "Learning pronunciation from a foreign language in speech synthesis networks," *arXiv preprint arXiv:1811.09364*, 2018.

[3] R. Kumar, R. Gangadharaiah, S. Rao, K. Prahallad, C. P. Rosé, and A. W. Black, "Building a better Indian English voice using "more data"," in *Proc. the 6th ISCA workshop on speech synthesis, Germany*, 2007.

[4] C. Yarra, R. Aggarwal, A. Rajpal, and P. K. Ghosh, "Indic TIMIT and Indic English lexicon: A speech database of Indian speakers using TIMIT stimuli and a lexicon from their mispronunciations," in *Proc. Oriental COCOSDA*. IEEE, 2019, pp. 1–6.

[5] A. L. Thomas, A. Prakash, A. Baby, and H. Murthy, "Code-switching in Indic speech synthesisers," in *Proc. Interspeech*, 2018, pp. 1948–1952. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2018-1178

[6] B. Ramani, S. L. Christina, G. A. Rachel, V. S. Solomi, M. K. Nandwana, A. Prakash, S. A. Shanmugam, R. Krishnan, S. K. Prahalad, K. Samudravijaya *et al.*, "A common attribute based unified HTS framework for speech synthesis in Indian languages," in *ISCA Workshop on Speech Synthesis*, 2013.

[7] A. Baby, N. L. Nishanthi, A. L. Thomas, and H. A. Murthy, "A unified parser for developing Indian language text to speech synthesizers," in *International Conference on Text, Speech and Dialogue*, 2016.

[8] D. Mahanta, B. Sharma, P. Sarmah, and S. M. Prasanna, "Text to speech synthesis system in Indian English," in *Proc. TENCON*. IEEE, 2016, pp. 2614–2618.

[9] R. Saikia and S. R. Singh, "Generating manipuri english pronunciation dictionary using sequence labelling problem," in *Proc. International Conference on Asian Language Processing*. IEEE, 2016, pp. 67–70.

[10] A. Baby, A. L. Thomas, N. L. Nishanthi, and T. Consortium, "Resources for Indian languages," in *Community-Based Building of Language Resources*, 2016.

[11] S. S. VijayaRajSolomon, V. Parthasarathy, and N. Thangavelu, "Exploiting acoustic similarities between Tamil and Indian English in the development of an HMM-based bilingual synthesiser," *IET Signal Processing*, vol. 11, no. 3, pp. 332–340, 2016.

[12] O. Maxwell and J. Fletcher, "Acoustic and durational properties of Indian English vowels," *World Englishes*, vol. 28, no. 1, pp. 52–69, 2009.

[13] T. Hayashi, R. Yamamoto, K. Inoue, T. Yoshimura, S. Watanabe, T. Toda, K. Takeda, Y. Zhang, and X. Tan, "ESPnet-TTS: Unified, Reproducible, and Integratable open source End-to-End Text-to-Speech Toolkit," *arXiv:1910.10909*, 2019.

[14] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan *et al.*, "Natural TTS synthesis by conditioning wavenet on mel spectrogram predictions," in *Proc. ICASSP*. IEEE, 2018.

[15] R. Yamamoto, E. Song, and J.-M. Kim, "Parallel WaveGAN: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram," in *Proc. ICASSP*. IEEE, 2020.

[16] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *arXiv:1609.03499*, 2016.

[17] H. Sirsa and M. A. Redford, "The effects of native language on Indian English sounds and timing patterns," *Journal of Phonetics*, vol. 41, no. 6, pp. 393–406, 2013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0095447013000399

11th ISCA Speech Synthesis Workshop (SSW 11)
26-28 August 2021, Budapest, Hungary

Cross-lingual Transfer of Phonological Features for Low-resource Speech Synthesis

Dan Wells, Korin Richmond

The Centre for Speech Technology Research,
University of Edinburgh, United Kingdom
{dan.wells, korin.richmond}@ed.ac.uk

Abstract

Previous work on cross-lingual transfer learning in text-to-speech has shown the effectiveness of fine-tuning phonemic representations on small amounts of target language data. In other contexts, phonological features (PFs) have been suggested as a more suitable input representation than phonemes for sharing acoustic information between languages, for example in multilingual model training or for code-switching synthesis where an utterance may contain words from multiple languages. Starting from a model trained on 14 hours of English, we find that cross-lingual fine-tuning with 15 minutes of German data can produce speech with subjective naturalness ratings comparable to a model trained from scratch on 4 hours of German, using either phonemes or PFs. We also find a modest but statistically significant improvement in naturalness ratings using PFs over phonemes when training from scratch on 4 hours of German.

Index Terms: speech synthesis, low-resource, cross-lingual, transfer learning

1. Introduction

Phonemes are often used as atomic input symbols to text-to-speech (TTS) systems as an explicit representation of the pronunciation of input text [1]. This is useful even for large neural sequence-to-sequence models which have the capacity to learn implicit pronunciation models directly from text inputs but which may make mistakes compared to grapheme-to-phoneme (g2p) conversion models trained on high-quality lexicons [2, 3]. Such large TTS models are typically trained using tens of hours of audio data with associated text transcriptions, which alongside the specialist linguistic knowledge required to convert raw text into phoneme strings are expensive resources to attain and limit the application of these models to a small proportion of the world's 7,000 languages.

For languages with minimal data resources for TTS model training, we might instead consider fine-tuning an existing model from another language with much more data available. In a phoneme-based system, input embeddings for phonemes common to both languages may be initialised in the target-language model by copying source-language parameters directly. For phonemes unique to the target language, however, some additional method is required to determine whether any particular source phoneme may provide a suitable starting point. In [4], a learned mapping is compared to a unified symbol space constructed by aligning phoneme symbols in each language using linguistic expertise, with both approaches achieving similar naturalness ratings when initialised from a model of 24 hours of English speech and fine-tuned with 15 minutes of Mandarin data. These fine-tuning approaches outperform a baseline with random initialisation of Mandarin phoneme embeddings.While the learned phoneme mappings were found largely to correspond with expert mappings, some target-language phonemes went unmapped due to low confidence in the suggested source phoneme and still had to be initialised from scratch. This follows from the atomic nature of phonemic input symbols, such that automatic phoneme mapping is an all-or-nothing approach.

An alternative approach is to decompose phonemic symbols into sets of distinctive phonological features (PFs) corresponding to articulatory attributes such as tongue position, degree of closure and voicing [5]. This representation reveals shared characteristics between phonemes which are not evident when considering only their atomic symbols in a transcription system such as the International Phonetic Alphabet (IPA) [6], and makes it possible to transfer learned embeddings for individual features between languages and so compose representations for target-language phonemes completely unseen during source model training. Previous work has used PF representations to share acoustic information between languages during multilingual model training for LSTM-RNN [7, 8] or feed-forward [9] neural network acoustic models. These models typically include PFs as part of a wider set of linguistic features, sometimes including phoneme labels as well, drawn from a unified symbol space across all training languages. In [7], for example, this data pooling approach using PFs was found to improve naturalness ratings for low-resource languages relative to individual voices trained using only data from those languages.

Our work is closest to that of [10], who use PFs in an encoder-decoder model with attention based on [11] to enable zero-shot synthesis of code-switched speech. They showed that a model trained on one language can be used to generate intelligible speech in a completely unseen target language with no acoustic training data available. Although they evaluated their system in an extreme setting with entire utterances comprised of target-language words, the work was motivated by the need to handle individual vocabulary items being embedded within source-language utterances, for example foreign names. We are directly interested in synthesising full utterances in the target language, and so apply a similar method in a transfer learning context, starting from a high-resource English source model and fine-tuning with either 15 minutes or 4 hours of transcribed German data. Also similar to [10], we rely on considerable lexical resources for g2p conversion prior to PF expansion, so that 'low-resource' in our case refers primarily to this relatively limited availability of transcribed speech data.

2. Phonological features

We use a set of binary phonological features derived from those introduced in Chomsky and Halle's *Sound Pattern of English* (SPE) [5]. In this formalism, each phoneme is represented as a

10.21437/SSW.2021-28

binary vector of 24 features as listed in Table 1. Of these, 19 are a selection of SPE features which adequately describe the phonetic inventories of English and German, and are essentially phonological in nature. We also add 5 features to capture aspects of input text strings, for example representing the end of a sentence or other prosodically-relevant punctuation types.

Table 1: *SPE-style phonological features.*

Category	Features
Major class	syllabic, consonantal, sonorant
Cavity	coronal, anterior, high, low, front, back, round, nasal, lateral, constricted glottis
Manner	continuant, tense, delayed release
Source	voice, strident, subglottal pressure
Text	space, end of sentence, question, exclamation, other punctuation

Following discussion in [5, pp. 353–355] on the treatment of glides relative to high vowels, e.g. /j/ vs. /i/, and to account for syllabic consonants, e.g. /n̩/ in 'button', we replace the original SPE *vocalic* feature with *syllabic*. We also add an explicit *front* feature for horizontal tongue body position alongside *back* to allow for distinction of central vowels in our feature system, e.g. open-mid front /ɛ/ [+*front*, −*back*] vs. central /ɜ/ [−*front*, −*back*]. All other features and mappings between phonetic segments and phonological feature vectors follow closely with those laid out in [5].

For a concrete example, consider our scenario of fine-tuning a high-resource English model using a small amount of German data. Both languages' phonemic inventories include an unvoiced velar plosive /k/, while only German natively makes use of an unvoiced velar fricative /x/. These two sounds share many features, both being produced at the same place of articulation in the mouth with the back of the tongue raised and without vibration of the vocal folds. The main difference between the two is the degree of closure in the oral cavity, with transient but complete interruption of airflow in the case of /k/ compared to narrowing of the vocal tract enough to generate turbulent airflow and constant noise for /x/. If we were only to consider the atomic symbols /k/ and /x/, for example by converting them to one-hot indices in a neural embedding table, these similarities may not be apparent, and we would have to make hard decisions about a possible mapping between these sounds if we wanted to transfer acoustic information learned on English data to our German model, as in [4]. In our PF representation, on the other hand, these two phonemes differ only in the specification of the feature *continuant*, which is − for /k/ and + for /x/. As such, at the beginning of our fine-tuning regime the encoder of our German model is initialised with a representation of /x/ which already contains much information learned from the English /k/, supplemented by [+*continuant*] English phonemes such as /s/. Although we do not test it formally here, we find these initial representations to produce somewhat intelligible German speech even before any target-language data has been seen by the model, as in [10], albeit retaining our English source speaker's vocal quality and accent.

Our binary feature representation largely overlaps with that used in PanPhon [12], and differs from the multi-valued features used in [10], which map more directly to IPA categories such as *vowel frontness* or *consonant place*. While our feature set gives a more compact representation, with 24 features vs. 60 in [10] (after conversion to binary vectors), it is perhaps less interpretable in familiar linguistic terms, for example with the *palatal* place of articulation feature in a multi-valued representation instead being composed from [+*high*, −*low*, −*back*] feature specifications in our system. Previous work on phonological feature detection from speech [13] found similar performance between an SPE-style binary feature system like ours and multi-valued features, and [8] showed improvements for multilingual TTS training using inputs augmented with PFs of both kinds, suggesting that either formalism may be adequate for speech processing tasks.

3. Methodology

3.1. Speech data

For our English voice we use part of the M-AILABS Speech Dataset [14], from the female US speaker *mary_ann*. We only use recordings from the *northandsouth* text, as other recordings from this speaker have a slight reverberant quality. For German, we use the CSS10 dataset [15], which provides a single female speaker. Both corpora are drawn from non-professional audiobook recordings made as part of the LibriVox project [16].

The CSS10 German corpus comprises 16 hours of speech sampled at 22.05 kHz, whereas M-AILABS provides 18 hours sampled at 16 kHz. For our English source models we randomly sample 14 hours (hereafter labelled 840 minutes) from M-AILABS as a training set and 90 minutes for validation. For German, we sample training sets of 15 minutes and 4 hours (240 minutes) and validation sets of 5 and 20 minutes respectively to match the low-resource training setting [17]. A disjoint set of 70 utterances is held out to synthesise listening test stimuli. All German utterances are downsampled to 16 kHz to match the English data. Table 2 summarises these data partitions.

Table 2: *Dataset summary: total number of utterances, average length in phonemes and average duration in seconds.*

Dataset	Utterances	Phones	Duration
EN-train-840	6975	97	7.23
EN-val-90	754	98	7.16
DE-train-240	1698	102	8.48
DE-val-20	153	94	7.85
DE-train-15	103	106	8.76
DE-val-5	38	98	8.12
DE-test	70	87	7.51

As part of dataset selection, we exclude from the English data any utterances with raw text transcriptions longer than 200 characters, and from the German any transcripts longer than 170 characters. This only serves to remove outliers from each dataset, and does not affect the overall distribution of observed transcript lengths. We also exclude any utterances from M-AILABS with digits in their raw transcripts, since we found the normalised transcripts provided did not match the words spoken in several instances. For German test utterances, we select only those with transcripts ending in some kind of intonational phrase-final punctuation $p \in \{.!?;\}$. We do this to increase the proportion of test stimuli which correspond to complete sentences, given that the CSS10 corpus was created by automatically segmenting long audiobook chapters and is not guaranteed

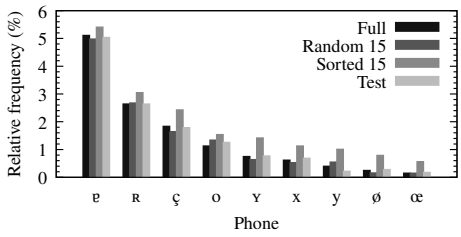


Figure 1: *German-specific phoneme frequencies for different subsets of CSS10 data.*

to have a one-to-one correspondence between segmented utterances and source text sentences.

When sampling German training subsets, we first sort utterances by how many phonemes they contain which are specific to German and therefore unseen during English source model training. We then select utterances starting with the most unseen phoneme types (out of 9 total) until the target dataset duration is met, so maximising training examples for these phones in our low-resource setting. We consider this a valid approach when some lexical resources are available in the target language, since prompt selection in this way can be done before recording any audio. Sorting by unseen phoneme type counts tends to give a greater increase in relative frequency of the least frequent phones, whereas sorting by token counts instead boosts the most frequent unseen phones. Figure 1 shows the effect of this procedure when sampling 15 minutes of German audio; the effect is reduced for 240 minute subsets, as even random sampling begins to exhaust the supply of the least frequent phones in the data. Type counts also restrain the tendency to select longer utterances compared to token counts, although as seen in Table 2 German training utterances are still slightly longer on average than validation utterances, which are not sorted by unseen phoneme counts before sampling.

3.2. Grapheme-to-phoneme conversion

To encode inputs using phonological features, we first need to convert input text to IPA phoneme strings. Where possible, we look up pronunciations in a lexicon: the General American surface form of Combilex [18] for English and the German lexicon from MaryTTS [19], mapping their individual phone sets to IPA symbols. To handle out-of-vocabulary items in each language we train g2p models from these lexicons using the Phonetisaurus toolkit [20].

3.3. Model details

We use a modified Tacotron 2 [11] architecture to predict acoustic features from text, based on the Mozilla TTS implementation [21]. Following [10], in our PF-based models we replace phoneme embeddings with a single linear layer over binary feature inputs, with matching 512-dimensional hidden representations. Mozilla TTS retains the reduction factor used in the original Tacotron [22], predicting r output frames per decoder step. We had better results training our English source model with $r = 2$, predicting frames in pairs rather than individually as in [11], and use the same reduction factor when fine-tuning German models. All other architectural details match [11].

We train English source models for 100k steps, using a Rectified Adam optimiser [23] with batch size 32 and learning rate 1×10^{-4}. German-only models use the same training hyperparameters but run for 60k steps, and fine-tuned models run for 60k steps with a learning rate of 3×10^{-5}. In this way, all German models using the same data split have equal exposure to training examples in that language, and we can evaluate the potential of each model and training scheme in matched data settings. As 240 minutes of speech is much less than is typically used to train sequence-to-sequence neural TTS models such as ours, we were concerned to ensure that our German-only models were adequately trained for fair comparison with the fine-tuned models which also see 14 hours of English data. The cutoff at 60k training steps was chosen to enable strong alignments between input and output timesteps to be learned by the German-only models, which we found to be the major factor preventing gross synthesis errors for those systems.

When fine-tuning phonological feature-based models, which we label *F-{15,240}-ft* depending on amount of German data used, all model parameters are copied directly from the English source model, since PF inputs are completely shared between the two languages. For phoneme-based models (*P-{15,240}-ft*), we copy learned English embeddings directly for all shared phonemes. For German-specific phonemes, we follow [10] and initialise their embeddings with the closest English phoneme largely according to PF specifications. This presents a stronger baseline to test PF systems against compared to leaving them with untouched random initialisations from the English pre-training stage. Figure 3(b) indicates the English phonemes selected to initialise German-specific phoneme embeddings.

We found that stop token prediction did not fare well when transferring from English to German. Fine-tuning this component led to 69% of synthesised utterances from *240-ft* systems and 17% from *15-ft* hitting an upper limit on decoder steps, often producing audible 'babbling' for the additional duration following synthesis of text prompts. This may be caused by mismatches in utterance-final prosody or other acoustic differences between English and German, or perhaps the increased proportion of sentence-fragment utterances in the German data compared to English. Models trained from scratch on our German data didn't exhibit this issue to the same degree, and reinitialising stop token projection weights rather than transferring from English source parameters during fine-tuning largely addresses the problem. Synthesis of our final *15-ft* test stimuli saw no utterances reaching the maximum decoder steps, while the proportion in *240-ft* systems was reduced to 17%.

We also train a Parallel WaveGAN vocoder [24] on our English dataset to generate audio from predicted acoustic features (implementation based on [25]). This model is trained as described in [24], for 400k training steps. We find the vocoder to transfer well to the unseen speaker in our German data without additional fine-tuning (cf. discussion in [26]), though since vocoder training requires only audio and extracted acoustic features and not aligned text transcripts, target-language vocoder training could be viable even in a low-resource setting.

3.4. Listening tests

We evaluate system performance by conducting MUSHRA-style listening tests [27]. Each test panel comprises multiple versions of the same utterance synthesised by each system under test, plus a natural speech reference (recorded by the same speaker used in training) and vocoded speech using mel spectral features extracted from the reference (copy synthesis). Natural speech is presented as an explicit reference and also included as

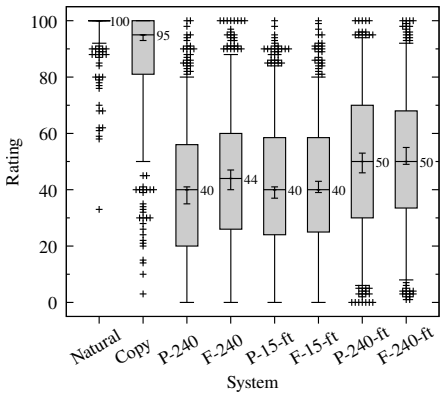


Figure 2: *MUSHRA naturalness ratings per system. Central bars indicate median ratings with 99% confidence intervals, boxes span 25–75% quartiles and whiskers cover 95% of results for each system. Outliers are marked with* $+$.

a hidden reference among other test samples, randomly ordered. Given the difficulty in identifying a suitable 'anchor' stimulus to serve as a lower bound for expected quality in speech synthesis, no such stimulus is included in our tests; each panel therefore contains 9 audio samples in total. Participants are asked to listen to the reference and then to provide a rating from 0–100 for each test sample reflecting how 'natural' they sound compared to the reference. To proceed to the next panel, at least one sample must be rated at 100 on the naturalness scale.

We recruited 40 participants through Prolific, filtering for native speakers of German, and conducted listening tests on the Qualtrics survey platform. Each participant completed 16 panels randomly allocated from our held-out set of 70 test utterances, with each utterance being rated by 9 or 10 participants in total. The average test duration was 35 minutes, and participants were paid £5 for their time.

4. Results

4.1. Subjective evaluation

The MUSHRA naturalness ratings for each system gathered through our subjective listening tests are shown in Figure 2. All systems present a wide range of participant ratings, including copy synthesis and even the hidden reference natural speech to some extent. We did not find any systematic source for this (e.g. particular stimuli or participants), and attribute it to natural variation in subjective ratings. Audio samples of test stimuli are available online.[1]

We test for significant differences between systems using double-sided pairwise Wilcoxon signed-rank comparisons, applying the Bonferroni correction with $\alpha = 0.01$ (for 28 pairwise comparisons, significance is found at $p < 0.00036$). Both *F-240-ft* and *P-240-ft* are significantly more natural than all other TTS systems, but there is no significant difference between them. The two systems fine-tuned with 15 minutes of German data are not significantly different from each other or either of the two systems trained on 240 minutes of German data only. The German-only system trained with PF inputs (*F-240*) is significantly more natural than the equivalent system using phonemes (*P-240*).

From these results, we see that by fine-tuning a source model trained on a high-resource language with as little as 15 minutes of annotated speech data in the target language, it is possible to match peformance against a system trained on 240 minutes of data from the target language alone. Furthermore, significant improvements in naturalness of the synthesised voice can be found by increasing the amount of fine-tuning data to 240 minutes. This is true for both phoneme- and PF-based systems, confirming previous results on fine-tuning from phoneme inputs in [4] and effectively extending the method to PFs with their more flexible and straightforward method for initialising target-language encoder representations compared to atomic phoneme mappings. We also find that, in the absence of a source model in another language, PFs can give a significant boost to naturalness ratings compared to phonemes in a low-resource setting with 240 minutes of target-language data.

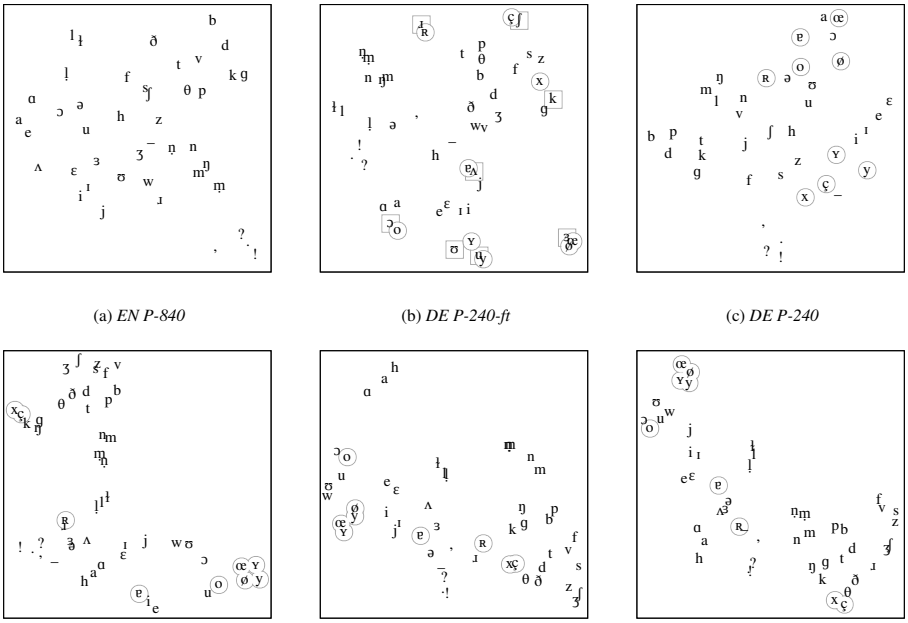
4.2. Input embeddings

To analyse the learned representations of phonemes in our models, we project input embeddings to two dimensions using UMAP [28], as shown in Figure 3. We encourage somewhat more local structure in our projections by reducing the default number of neighbouring points considered in the reference implementation of UMAP from 15 to 5, based on the intuition that individual phonemes are typically more closely related to a small subset of other sounds in any particular phoneme inventory in which they may be found. For clarity in Figures 3(a) *EN P-840* and 3(c) *DE P-240*, we exclude the randomly-initialised embeddings of phonemes from the other language (which are never updated during training for these systems) when projecting the embedding spaces. Although UMAP is a stochastic algorithm, we found the projections of our learned embeddings to be quite consistent across multiple runs.

There is some apparent structure for both phoneme and PF representations, with vowels and consonants grouped separately, distinct consonant classes grouped together (plosives, fricatives and nasals especially) and voiced and unvoiced consonants at the same place of articulation lying close together. Some higher-level relationships appear important for PF projections, for example with vowels seemingly arrayed primarily along an axis of rounding and within those [$\pm$*round*] clusters by frontness and height. For consonants, the *back* feature also appears to be significant above manner of articulation, with [$+$*back*] plosives /k/ and /g/, fricatives /ç/ and /x/ and the nasal /ŋ/ tending to be separated from their anterior counterparts.

Interesting differences may be seen in the behaviour of the two German-specific fricatives, velar /x/ and palatal /ç/, between the *P-240* model trained only on German data and *P-240-ft* which was fine-tuned from English phoneme representations. In *P-240*, these sounds are grouped closely together with other fricatives, and are quite apart from any plosive consonants. In the fine-tuned model, on the other hand, the separation between fricative and plosive is less clear, specifically with velar plosives /k/ and /g/ appearing close to /x/, while /ç/ is somewhat separated from the other fricatives along with /ʃ/. Notably, these two phonemes were initialised from the learned English embeddings for /k/ and /ʃ/, respectively. If we consider other German-specific phonemes and the corresponding English phonemes from which they were initialised, there is apparently very little movement from the English starting points in all cases. This

[1] https://dan-wells.github.io/pf-tts



(a) *EN P-840* (b) *DE P-240-ft* (c) *DE P-240*

(d) *EN F-840* (e) *DE F-240-ft* (f) *DE F-240*

Figure 3: *UMAP projections of input symbol embeddings for English and German models using phonemes (a–c) and PFs (d–f). German-specific phonemes are marked by circular outlines, and English phonemes used to initialise their representations in (b) by squares. Unseen German phonemes are included in (d) to show that novel combinations of PFs also produce sensible representations.*

could be a result of the high-dimensional (512) phoneme embedding space used: in such a large representational space, it may be possible to adapt a plosive /k/ to sound adequately like its corresponding fricative /x/ by making small perturbations in many dimensions. This high-dimensional perturbation might not then be preserved during low-dimensional projection as we have done here. By comparison, these phonemes pattern consistently across both *DE* models trained from scratch and through fine-tuning when using PFs, as well as in *EN F-840*, where they were completely unseen during training. This supports the notion that PFs should be a stable representation cross-linguistically, backing up observed improvements in multilingual training contexts in [7, 8].

5. Conclusion

In this work, we experimented with phonological feature vector inputs to TTS models in a transfer learning context. We confirmed previous results which showed that cross-lingual fine-tuning is a viable method for training synthetic voices with limited amounts of target language data, with source models trained on 14 hours of English being adapted using 15 minutes of German data matching the subjective naturalness ratings of models trained from scratch using 4 hours of German data only. We found this result to hold for PFs as well as phonemes, but consider PFs to bring practical benefits with regard to ease of parameter sharing in this transfer learning context. We also found a small but statistically significant improvement in naturalness ratings when training a voice from scratch on 4 hours of German data using PFs over phonemes.

While the models trained here may be called 'low-resource' in terms of annotated speech data available in the target language, we still rely on considerable lexical resources for grapheme-to-phoneme conversion of input text before we can expand IPA symbols to PFs. Future work may consider the application of recent approaches to multilingual g2p systems [29] as part of this low-resource pipeline, or make use of additional pre-existing linguistic resources such as the PHOIBLE phonological inventory database [30]. Following our analysis of learned input embeddings, we would also like to investigate more constrained embedding spaces to encourage more efficient parameter sharing, especially for phonemes which remain a common choice of input representation for TTS.

6. Acknowledgements

We would like to thank Gustav Eje Henter for helpful discussion on analysing MUSHRA results and Alexander Schotthöfer for translating experimental materials into German. This work was supported in part by the UKRI Centre for Doctoral Training in Natural Language Processing, funded by the UKRI (grant EP/S022481/1) and the University of Edinburgh, School of Informatics and School of Philosophy, Psychology & Language Sciences.

7. References

[1] K. Kastner, J. F. Santos, Y. Bengio, and A. Courville, "Representation Mixing for TTS Synthesis," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2019, pp. 5906–5910.

[2] J. Taylor and K. Richmond, "Analysis of Pronunciation Learning in End-to-End Speech Synthesis," in *Interspeech 2019*. ISCA, Sep. 2019, pp. 2070–2074.

[3] J. Fong, J. Taylor, K. Richmond, and S. King, "A Comparison of Letters and Phones as Input to Sequence-to-Sequence Models for Speech Synthesis," in *10th ISCA Speech Synthesis Workshop*. ISCA, Sep. 2019, pp. 223–227.

[4] Y.-J. Chen, T. Tu, C.-c. Yeh, and H.-Y. Lee, "End-to-End Text-to-Speech for Low-Resource Languages by Cross-Lingual Transfer Learning," in *Interspeech 2019*. ISCA, Sep. 2019, pp. 2075–2079.

[5] N. Chomsky and M. Halle, *The Sound Pattern of English*. New York: Harper & Row, 1968.

[6] International Phonetic Association, *Handbook of the International Phonetic Association: A Guide to the Use of the International Phonetic Alphabet*. Cambridge University Press, 1999.

[7] A. Gutkin, "Uniform Multilingual Multi-Speaker Acoustic Model for Statistical Parametric Speech Synthesis of Low-Resourced Languages," in *Interspeech 2017*. ISCA, Aug. 2017, pp. 2183–2187.

[8] A. Gutkin, M. Jansche, and T. Merkulova, "FonBund: A Library for Combining Cross-lingual Phonological Segment Data," in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. Miyazaki, Japan: European Language Resources Association (ELRA), May 2018.

[9] I. Himawan, S. Aryal, I. Ouyang, S. Kang, P. Lanchantin, and S. King, "Speaker Adaptation of a Multilingual Acoustic Model for Cross-Language Synthesis," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2020, pp. 7629–7633.

[10] M. Staib, T. H. Teh, A. Torresquintero, D. S. R. Mohan, L. Foglianti, R. Lenain, and J. Gao, "Phonological Features for 0-Shot Multilingual Speech Synthesis," in *Interspeech 2020*. ISCA, 2020, pp. 2942–2946.

[11] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerry-Ryan, R. A. Saurous, Y. Agiomvrgiannakis, and Y. Wu, "Natural TTS Synthesis by Conditioning Wavenet on Mel Spectrogram Predictions," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 2018, pp. 4779–4783.

[12] D. R. Mortensen, P. Littell, A. Bharadwaj, K. Goyal, C. Dyer, and L. Levin, "PanPhon: A Resource for Mapping IPA Segments to Articulatory Feature Vectors," in *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. Osaka, Japan: The COLING 2016 Organizing Committee, Dec. 2016, pp. 3475–3484.

[13] S. King and P. Taylor, "Detection of phonological features in continuous speech using neural networks," *Computer Speech & Language*, vol. 14, no. 4, pp. 333–353, Oct. 2000.

[14] Munich Artificial Intelligence Laboratories GmbH, "The M-AILABS Speech Dataset," 2019. [Online]. Available: https://www.caito.de/2019/01/the-m-ailabs-speech-dataset/

[15] K. Park and T. Mulc, "CSS10: A Collection of Single Speaker Speech Datasets for 10 Languages," in *Interspeech 2019*. ISCA, 2019, pp. 1566–1570.

[16] LibriVox, "LibriVox – Free public domain audiobooks." [Online]. Available: https://librivox.org/

[17] K. Kann, K. Cho, and S. R. Bowman, "Towards Realistic Practices In Low-Resource Natural Language Processing: The Development Set," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, 2019, pp. 3340–3347.

[18] K. Richmond, R. A. J. Clark, and S. Fitt, "Robust LTS Rules with the Combilex Speech Technology Lexicon," in *Interspeech 2009*, 2009, pp. 1295–1298.

[19] M. Schröder and J. Trouvain, "The German Text-to-Speech Synthesis System MARY: A Tool for Research, Development and Teaching," in *4th ISCA Speech Synthesis Workshop*, 2001.

[20] J. R. Novak, N. Minematsu, and K. Hirose, "Failure Transitions for Joint n-Gram Models and G2P Conversion," in *Interspeech 2013*, 2013, pp. 1821–1825.

[21] "Mozilla/TTS," Mozilla, Apr. 2021. [Online]. Available: https://github.com/mozilla/TTS

[22] Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio, Q. Le, Y. Agiomyrgiannakis, R. Clark, and R. A. Saurous, "Tacotron: Towards End-to-End Speech Synthesis," in *Interspeech 2017*. ISCA, Aug. 2017, pp. 4006–4010.

[23] L. Liu, H. Jiang, P. He, W. Chen, X. Liu, J. Gao, and J. Han, "On the Variance of the Adaptive Learning Rate and Beyond," in *Eighth International Conference on Learning Representations (ICLR 2020)*, Apr. 2020.

[24] R. Yamamoto, E. Song, and J. Kim, "Parallel WaveGAN: A Fast Waveform Generation Model Based on Generative Adversarial Networks with Multi-Resolution Spectrogram," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2020, pp. 6199–6203.

[25] T. Hayashi, "Kan-bayashi/ParallelWaveGAN," 2020. [Online]. Available: https://github.com/kan-bayashi/ParallelWaveGAN

[26] A. Corral, I. Leturia, A. Séguier, M. Barret, B. Dazéas, P. Boula de Mareüil, and N. Quint, "Neural Text-to-Speech Synthesis for an Under-Resourced Language in a Diglossic Environment: The Case of Gascon Occitan," in *Proceedings of the 1st Joint SLTU and CCURL Workshop (SLTU-CCURL 2020)*. Marseille, France: European Language Resources Association, 2020, pp. 53–60.

[27] ITU-R, "Recommendation ITU-R BS.1534-3: Method for the subjective assessment of intermediate quality level of audio systems," International Telecommunication Union, Tech. Rep. ITU-R BS.1534-3, 2015.

[28] L. McInnes, J. Healy, and J. Melville, "UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction," *arXiv:1802.03426*, 2018. [Online]. Available: https://github.com/lmcinnes/umap

[29] K. Gorman, L. F. E. Ashby, A. Goyzueta, A. D. McCarthy, S. Wu, and D. You, "The SIGMORPHON 2020 Shared Task on Multilingual Grapheme-to-Phoneme Conversion," in *Proceedings of the 17th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, Jul. 2020, pp. 40–50.

[30] S. Moran and D. McCloy, Eds., *PHOIBLE 2.0*. Jena: Max Planck Institute for the Science of Human History, 2019. [Online]. Available: https://phoible.org/

11th ISCA Speech Synthesis Workshop (SSW 11)
26-28 August 2021, Budapest, Hungary

Mind your p's and k's – Comparing obstruents across TTS voices of the Blizzard Challenge 2013

Ayushi Pandey[1], *Sebastien Le Maguer*[1], *Julie Carson-Berndsen*[2], *Naomi Harte*[1]

[1]Sigmedia Lab, ADAPT Centre, School of Engineering, Trinity College Dublin, Ireland
[2]ADAPT Centre, School of Computer Science, University College Dublin, Ireland
pandeya@tcd.ie, lemagues@tcd.ie, julie.berndsen@ucd.ie, nharte@tcd.ie

Abstract

Obstruent consonants have been investigated in speech quality assessment studies of natural speech, where enhancing their perception has improved overall speech quality. This paper presents a comparative analysis of acoustic-phonetic features of obstruent consonants in synthetic speech. Features for obstruent consonants are identified where TTS systems differ significantly from a natural human voice, as a function of quality.

The synthetic speech voices from the Blizzard Challenge of 2013 are used for this investigation. TTS systems were first assigned groups based on their MOS rating (quality) and shared TTS technique (family). Then, acoustic-phonetic features characteristic of contrastive properties in obstruents, were extracted from all systems. While quality differences between low-rated systems and high-rated systems were observed in a large number of features, we report those where statistically significant differences (p-val < 0.001) were observed between the systems. Where quality effects were not found, we investigated whether systems of the same family exhibit similar behaviour. Finally, individual systems within a group were examined for their differing influence on the acoustic-phonetic feature set of obstruents. Here, we found that HMM systems with similar MOS ratings do not differ in their acoustic realization of obstruents, while Unit Selection systems showed stronger individual system variability.

A comparative analysis of obstruent consonants across TTS systems applies techniques from the domain of corpus-phonetics to the task of speech synthesis evaluation. Identifying phonologically relevant acoustic features, may indicate the underlying articulatory process compromised in those systems, that correlates with the distorted acoustics.

1. Introduction

Methods in speech synthesis evaluation have looked at a variety of tools and techniques to analyze synthetic speech in recent years. Some techniques assess the efficacy and scalability in real-world scenarios, like interactive avatar-based settings [1] and long-form, paragraph-level sentences [2]. Objective measurement-based techniques use comparative features like mel cepstral distortion, and the PESQ family of ITU-T standards to predict speech quality compared to a natural voice as reference. Such tests reduce the dependence on expensive human-based listening tests. Machine-learning based techniques such as AutoMOS [3] go even further in modelling human responses and reduce the dependence on parallel natural speech as reference [4, 5]. Electroencephalography (EEG) [6, 7] and pupillometry [8] based measurements have explored the relationship between behavioural/neuronal responses of human participants and synthetic speech quality. To compare the perceived qualities of different TTS techniques, comparative MOS and MUSHRA based perceptual judgements have been conducted [9, 10].

Each of these techniques has advantages - ranging from practical environments, to cost-effective techniques, to contributing to knowledge of quality in speech perception. However, a feature-based comparison of systems using acoustic-phonetic attributes of the signal is largely missing from the discussion.

A central question in the domain of acoustic-phonetics is to identify those features in the signal that can contribute to the perception of contrast between speech sounds. For example, the low-frequency energy region before the release of the consonant allows us to perceive the difference between the utterances "take a pull" and "take a bull". While contrast may not necessarily be the target percept in studies of speech naturalness or quality, contrastive *features* encode rich information about the characteristics of speech sounds. Comparing TTS systems using these features can provide us with insights into system weaknesses, such as poor reproduction characteristics for specific types of consonants.

This paper is the first work we know of that applies techniques from the domain of corpus-phonetics to the task of speech synthesis evaluation. The dataset used for this analysis is the Blizzard Challenge 2013 (BC-2013), which is a single-speaker, parallel database, covering a variety of TTS techniques. Systems of BC-2013, have been grouped on the basis of their shared TTS technique (family) and MOS (quality). Comparative analysis between these groups has been conducted across each obstruent feature, with the original human voice as the reference. The method used is fully automatic, inexpensive and easily reproducible, even at a large scale. We envisage that such an approach can give speech synthesis researchers much greater insights into how the synthetic speech their system produces may be perceived, before conducting subjective evaluation. Features identified in this analysis can be used for comparison between different TTS techniques, system qualities and individual differences between systems.

The paper is organized as follows: Section 2 discusses the properties of obstruent consonants, and the motivation for their choice in this study. Section 3 gives a detailed description of the experimental procedure, entailing the dataset, the feature extraction, and the statistical model. Section 4 presents the results and Section 5 the discussion. Section 6 concludes the paper.

2. Why study obstruents?

Obstruent consonants are a major phonological class of consonants, accounting for 6 distinct phoneme types for stops, [p, t, k, b, d, g], 9 for fricatives, [f, v, θ, ð, s, z, ʃ, ʒ, h], and 2 for affricates [ʧ, ʤ] in English. Obstruents cover a large portion of the consonantal region in any language or dataset. Cross-

10.21437/SSW.2021-29

	Bilabial	Labiodental	Dental	Alveolar	Postalveolar	Velar	Glottal
Stop	p b 122 130			t d 519 402		k ɡ 191 78	
Affricate					ʧ ʤ 35 32		
Fricative		f v 130 122	θ ð 53 219	s z 314 172	ʃ ʒ 96 1		h 218

Table 1: *Frequency distribution of obstruent consonants in the 100 sentences of BC-2013 corpus. Each system has an identical distribution. The rows represent the manners of articulation, while the columns represent the places of articulation.*

linguistic evidence [11] suggests that obstruents cover between two-thirds and three-quarters of the frequency in phoneme inventories across different language groups. In the BC-2013 dataset, obstruents cover 63.9% of the total consonantal population. Their statistical dominance in the dataset makes a compelling case for their analysis.

In addition to their widespread coverage, obstruent consonants have also been evaluated for their contribution to improved speech quality, and poor recognition in noise. In a sequence of studies, Li and Loizou [12–14] report that improved access to obstruents improves intelligibility of speech in noise. Additionally, obstruent recognition has also been found to be more impaired in degraded listening conditions [15, 16], compared to sonorants and vowels, whereas the manipulation of their target cues [17] results in improved recognition. Each of these studies underscore the critical role that preserving obstruents can play in speech perception in non-ideal listening conditions. In this paper, we postulate that synthetic speech may be considered as another such non-ideal scenario. Finally, obstruents contain many acoustic properties of the speech signal, which are not found in sonorants. For instance, stops are characterized by complete obstruction of airflow, which results in a region of silence, followed by a short, high-energy transient region known as burst. Analyzing stops gives us insights into how rapid changes of energy within the acoustic spectrum are handled across different systems. Fricatives do not obstruct the air completely, but force the air through a narrow constriction. This results in air flowing out at high volume velocity, resulting in aperiodic signal with amplitude in high frequencies.

Synthetic speech in BC-2013 contains a range of speech-qualities, and a large proportion of obstruents. Thus we can compare systems in terms of their influence on obstruent properties and explore whether we can uncover relationships with quality that have been established in natural speech. The next section describes the details of BC-2013, our feature extraction procedure and explains the statistical model used for this analysis.

3. Experimental setup

3.1. Dataset

The Blizzard Challenge (BC) is an international task designed to compare state of the art corpus-based speech synthesis systems[1]. All participating teams are given the same training dataset. To participate in the challenge, all teams submit the same prescribed sentences as outputted by a TTS system of their own design. A subset of these sentences are then evaluated with subjective listener tests using MOS.

[1] https://www.synsig.org/index.php/Blizzard_Challenge

In this study, we use data from Blizzard Challenge 2013 (BC-2013). To generate the test sentences, 5 teams used parametric HMM-based techniques (systems C, F, H, I, P), 3 used Unit-Selection (systems B, L, N), and 2 used Hybrid method (systems K,M) for synthesis. Each team submitted the same 100 test sentences, which made BC-2013 a rich source for parallel synthetic speech, with controlled variability.

For the subjective listener test, 11 sentences were evaluated by 426 listeners. While many attributes of speech quality were evaluated, in this work, we focus on the perceived naturalness of the systems. Overall, system M was rated as the most natural and most similar to human speech, with a median MOS of 4 on a 5-point scale. Systems K (Hybrid), I, C (HMM) and L, N (Unit Selection) were the next most highly ranked. System P (HMM) was considered the least natural, and received a MOS of 1.2. In our analysis, the full 100 sentences submitted by each system were used for comparative analysis.

3.2. Feature extraction

This section discusses the feature extraction procedure. First, we discuss the phoneme and sub-phonemic boundary identification in the time domain. Then, we detail the signal processing specifications required for extraction of features from the noisy region of obstruents.

*3.2.1. **Temporal boundary identification***

For phoneme boundary estimation, all systems were forced-aligned using the Montreal Forced Aligner (MFA) [18]. Regions marked for obstruents could now be extracted from the resultant phoneme boundaries. The most important acoustic correlates of obstruent consonants are features extracted from the noisy region of the consonants. While noise continues in fricatives through the length of the consonant, in affricates and stops, it follows a region of silence. Therefore, a sub-phonemic demarcation of the noise region, separated from the silent region needed to be identified.

While most studies on obstruent contrasts depend on careful, hand-corrected methods for the analysis, it would have rendered our corpus-based approaches quite unscalable. Similarly, toolkits such as AutoVOT [19] require a sample of hand-annotated training data, and did not provide the best results for pre-vocalic and intervocalic consonants. However, visually examining the spectrographic properties of stops and affricates, we found a sharp increase in amplitude, representing the burst. To extract this location automatically, we first converted the consonantal signal to its frequency domain. Then, all amplitude values $<$1.5 kHz were removed, because energy from the low-frequency voicing-bar interfered with the estimation of the energy of the burst. Finally, the remaining frequency-domain signal was passed through a moving-average filter. Where en-

ergy of the signal exceeded a threshold of 50-55 dB, and the point of the highest amplitude in that interval was marked as the beginning of the noise region. The threshold was decided upon after examining 20% of the sentences manually.

3.2.2. *Feature-set*

Acoustic-phonetic properties of obstruents across durational [20–22] , amplitudinal, spectral [23–25] and transitional cues [26–28] are well-established in the literature. The feature extraction procedure closely follows the methodologies presented in Jongman et al.'s seminal work on fricatives [24], and their recent, and more comprehensive extension into all manners of obstruents [29]. The present discussion omits transitional cues and limits the analyses only to the consonantal portion of obstruents. The RMS amplitude has also been calculated in the frequency domain. Also, those cues which cannot be compared across all manners of articulation (for example, closure duration is only relevant for stops and affricates) are excluded.

To extract the spectral parameters, all instances of obstruents were first passed through a high-pass filter, so that the analysis spectrum remains between 550 Hz and 10,000 Hz, to separate source and filter characteristics [30, 31]. For fricatives, a full Hamming window was placed at the center of the frication noise. For stops and affricates, a half Hamming window was placed at the start of the burst, such that the silence region was not included. Then, spectral properties were computed using an 512-point FFT taken over these windowed signals. A brief description is provided below:-

- *Consonant duration:-* The duration of the consonantal region, as returned by the MFA. In the pre-vocalic position, this region starts with the beginning of the closure, and ends with the onset of the vowel. Conversely in the post-vocalic position, it begins at the offset of the vowel, and follows to the end of the consonant. The unit of measurement was milliseconds (ms).
- *Noise duration:-* For stops and affricates, as described above. For fricatives, since noise persists through the length of consonant, the entire region was included. The unit of measurement was milliseconds (ms).
- *RMS amplitude:-* The root-mean-squared amplitude of the power spectrum.
- *Peak amplitude:-* The value of the highest amplitude in the spectrum. The unit of measurement is dB.
- *Peak frequency:-* This is the spectral frequency at which peak amplitude was identified. Its value was measured in Hz.
- *Dynamic amplitude:-* The difference between the peak amplitude, and the minimum amplitude below 2 kHz. The unit of measurement was dB.
- *Spectral tilt:-* The frequency domain of the spectrum was log-transformed, and then a least-squares regression line was fitted through it. The slope of this line returned the spectral tilt.

These features were extracted for obstruent consonants across all the systems, as well as the natural voice, independently. The purpose of such an extraction was to compare these features across all the systems, and to identify those features, where the system (or groups of systems, See Section 3.3) showed significant differences from the natural voice.

R	Group	Sys.	Description
R1	Hybrid-R1	M K	Hybrid systems with MOS 3-4
R2	HMM-R2	I C	HMM systems with MOS 2-3
	UnS-R2	L N	UnS systems with MOS 2-3
R3	HMM-R3	H F	HMM systems with MOS 1-2
	UnS-R3	B	UnS systems with MOS 1-2
R4	HMM-R4	P	HMM systems with MOS 1

Table 2: *Grouping strategy. Rank(R) of the system is decided by MOS for naturalness. The groups correspond to the intersection of the rank and the system family (Hybrid, HMM, Unit Selection (UnS).*

3.3. Grouping strategy

As mentioned in the previous section, the BC-2013 provides a variety of synthetic speech systems, which differ both in family and quality. To achieve this comparative analysis, a grouping strategy between systems was created. The explanation for each of the schemes is described below, and a concise description is displayed in Table 2. Systems were first divided into 4 groups: R1, R2, R3 and R4. R denotes "rank", which was decided simply by the obtained naturalness MOS for a given system. Systems that received MOS in the same interval, i.e, shared the system quality attribute, were assigned the same rank. A comparison based only on rank would not have yielded any family specific insights. Therefore, these groups were further subdivided, so that all systems of the same rank and same family were grouped together. Therefore, the resultant groups were: Hybrid-R1, HMM-R2, UnS-R2, HMM-R3, UnS-R3 and HMM-R4, where UnS means Unit Selection. This strategy allowed us to compare high-rated systems with low-rated systems from the same family. HMM-R4 received poor ratings, and has not been discussed in this paper.

3.4. Statistical model

A linear regression analysis models the relationship between two variables. A linear regression analysis with feature value as the dependent variable, and system group as the predictor variable was conducted for each of the features described in Section 3.2. Separate models were created for each feature, such that the dependent variable changed with every feature in the model, while the independent variable remained system groups each time.

It must be carefully noted here, that the feature value calculated for the natural voice was considered the reference point (the intercept) in each case. The **deviation** from this voice was the comparative metric across which different behaviours of groups were recorded. A univariate analysis of this type allowed for a descriptive model of system group against features, where effect of system groups on each feature could be independently analyzed, and comparative results could be reported.

4. Results

4.1. Experiment I : Comparing the same families of different ranks

The purpose of this experiment is to explore quality differences between groups of the same family. The groups under comparison are HMM-R2 vs HMM-R3, and UnS-R2 vs UnS-R3. Features which showed the most statistically significant differences

between groups have been identified. Comparative influences of groups on such features is presented in the subsequent sections.

4.1.1. Comparison between HMM-R2 and HMM-R3

The most informative features for observing quality differences between HMM-R2 and HMM-R3 were RMS amplitude, peak amplitude and spectral tilt.

On the basis of RMS Amplitude, we see differences between HMM-R2 and HMM-R3 across each manner of articulation. In affricates and fricatives, the HMM-R3 systems were observed to lower the RMS Amplitude. HMM-R2, on the other hand, did not differ significantly from the natural voice in any manner of articulation. RMS Amplitude dropped in affricates by 1.8 dB, and in fricatives by 1.5 dB, with strongly significant effects (p-val < 0.001). In stops, HMM-R3 systems were found to increase the amplitude by 0.51 dB, with a moderately significant effect (p-val < 0.05). Therefore, through these results we can conclude that poor-quality HMM-R3 systems show lower amplitude in affricates and fricatives, and marginally higher amplitude compared to natural voice. In each case, HMM-R2 was not found significantly different from natural voice.

The second feature under consideration is the peak amplitude. Similarly as above, HMM-R3 systems are found to lower the peak amplitude in the context of affricates and in fricatives. The peak amplitude dropped in affricates by 2.4 dB, and in fricatives by 1.4 dB, with significant effects (p-val < 0.01). HMM-R2 systems, on the other hand, do not differ from the natural voice in affricates. On the contrary, they are seen to increase the amplitude for fricatives. The behaviour of the two groups was not different in stops. Therefore, we can learn that fricatives in HMM-R2 systems exhibit louder maxima of amplitude, and HMM-R3 have softer peak amplitudes in affricates and fricatives alike.

The third feature considered important is the spectral tilt. In all the manners of articulation, low-quality HMM-R3 systems increase the spectral tilt with strongly significant effects. The magnitude of this increase is 1.93 dB in affricates, 4.14 dB in fricatives, and 3.14 dB in stops (p-val < 0.001). In affricates and fricatives, HMM-R2 systems do not differ significantly from the natural voice. But in stops, HMM-R2 also increase the spectral tilt. However, groups can still be separable within this context, because the magnitude of this increase is much lesser (0.95 dB) than in HMM-R3. Therefore, we observe that fricatives and affricates have steeper slopes in low-quality HMM systems across all manners of articulation. But in the context of stops, HMM-R2 also contribute to this effect.

4.1.2. Comparison between UnS-R2 and UnS-R3

The most important features for comparison between UnS groups are consonant duration, noise duration and spectral tilt.

Both UnS-R2 and UnS-R3 systems shorten the consonant duration in the context of fricatives and stops, while affricates do not show differences in groups for consonant duration. However, the shortening in high-quality UnS-R2 systems is seen with a stronger effect (p-val < 0.001), compared to UnS-R3 systems. In UnS-R2, fricatives are shortened by 7.5 ms and stops by 5.8 ms. In UnS-R3, on the other hand, fricatives and stops are shortened by 4.4 ms and 2.6 ms, respectively (p-val < 0.01). Therefore, we observe here that high-quality UnS-R2 systems shorten fricatives and stops more than low-quality UnS-R3.

The second feature considered important for UnS quality comparison is noise duration. Similar to observations for noise duration, a decrease of noise duration is found in both UnS-R2 and UnS-R3 groups for all manners of articulation. However, there are two differences. Firstly, stops show comparable decrease of noise duration between UnS-R2 and UnS-R3, and therefore are not deemed a reliable context for group differentiation. Secondly, although both fricatives and affricates have different influences of groups, they do so in different directions. UnS-R2 systems reduce the duration of fricatives with stronger significance, but affricates are shortened in UnS-R3 more strongly. Fricatives in UnS-R2 are shortened by 7.5 ms (p-val < 0.001), compared to 4.4 ms in UnS-R3 (p-val < 0.01). On the other hand, affricates are shorter by 7.4 ms in UnS-R2 (p-val < 0.05), and 9.8 ms (p-val < 0.01) in UnS-R3. So here, we can learn that noise duration is reduced in both UnS-R2 and UnS-R3 groups, across all manners of articulation. Group differences can be seen within fricatives and affricates. But the direction of influence is not consistent across manners.

The third feature under consideration is the spectral tilt. Here we see, that UnS systems on the whole lower the spectral tilt, instead of the increasing effect found in HMM systems. While the effect of lowering is strong and significant in all manners of articulation alike (p-val < 0.001), affricates and fricatives show greater separation between UnS-R2 and UnS-R3. In affricates, UnS-R2 decrease the tilt by 3.3 dB, and UnS-R3 by 7.3 dB. Similarly for fricatives, UnS-R2 decrease the tilt by 5.43 dB, and UnS-R3 by 8.7 dB. Stops, on the other hand, show comparable lowering in both UnS-R2 and UnS-R3 groups. Therefore, this result indicates that low-quality UnS-R3 systems flatten the spectral tilt more than UnS-R2 system, especially for fricatives and affricates.

4.2. Experiment II : Comparing individual differences between systems of a group

The purpose of this experiment is to explore individual differences between systems of the same group. Comparison will be made under Hybrid-R1 between M and K, under HMM-R2 between I and C, and under UnS-R2 between L and N.

4.2.1. Comparison between individual systems of Hybrid-R1

It is important to note that although M and K are in the same group, with obtained MOS of 3.9 and 3.4 respectively, that difference was statistically significant in the BC-2013 evaluations. The three most important features identified for systemic differences are RMS amplitude, peak frequency and spectral tilt.

Regarding **RMS Amplitude**, in the context of affricates, M was found to lower the RMS Amplitude by 1.7 dB (p-val $<$ 0.001), but K was not found to be significantly different from the natural voice. However, this trend completely reversed in the context of fricatives and stops. K was observed to influence a strongly significant increase the amplitude of 1.72 dB (p-val < 0.001). But in both of these contexts, M was not found different from the natural voice. Therefore, affricates are softer than natural voice in M, and fricatives and stops are louder in K. So we can see that, although each manner of articulation shows systemic differences between Hybrid systems, affricates oppose the trend exhibited by fricatives and stops.

The second feature considered reliable for systemic differences within Hybrid-R1 is **peak frequency**. K shows a statistically significant raising of peak frequency in all affricates, fricatives and stops context. In affricates, the increase is by 946.23 Hz, while in fricatives, we see an increase of 337.46 Hz. Finally in stops, although the increase is smallest, of 201.8 Hz compared to other places, the effect is still strongly significant. In no context does M differ from the natural voice. Therefore,

K exhibits maximum amplitude at higher frequencies, while M remains closer to natural.

Finally, K shows a statistically significant raising of spectral tilt in each context. The increase was of 1.2 dB in affricates, 5.4 dB in fricatives, and 3.5 dB in stops. M does not differ significantly from the natural voice in fricatives and stops. However, greater separation in systems can be seen in affricates, where M shows a moderately significant lowering of the spectral tilt (p-val < 0.05). Therefore, K shows a steeper slope in the spectrum, while M does not differ significantly from the natural voice.

4.2.2. Comparison between individual systems of HMM-R2

Differences between I and C were **not found** in any feature, across any manner of articulation. This indicates that systems I and C have consistent patterns of influence on all the features across manners of articulation.

4.2.3. Comparison between individual systems of UnS-R2

The first feature to compare differences between L and N is **RMS Amplitude**. Differences on the basis of RMS Amplitude can be seen in all three classes of Manner - i.e., in affricates, fricatives and stops. In affricates and fricatives, N shows a strongly significant lowering of RMS Amplitude. The magnitude of this lowering is 3.0 dB and 2.9 dB in affricates and fricatives respectively (p-val < 0.001). L, on the other hand, does not differ significantly from the natural voice. Among stops, the difference is less distinct, because N brings about only a modest lowering of 0.56 dB (p-val < 0.05).

The second feature under consideration is **peak frequency**. Systemic differences can be seen predominantly in affricates, and modestly in Stops. In affricates, L shows a moderately significant lowering of 211.86 Hz (p-val < 0.05), while N does not differ much from the natural voice. Among stops, although the systems differ individually, the pattern of affricates is not replicated. Here, both L and N show a lowering of the frequency. The effect although, is stronger in N, with a lowering of 173.14 Hz (p-val < 0.001), compared to L which lowers by 142.76 Hz (p-val < 0.01).

Finally, differences based on **spectral tilt** can be seen in all three classes of Manner. In affricates and stops, N shows a strongly significant lowering of 5.55 dB (p-val < 0.001) and 3.15 db (p-val < 0.001) respectively, and L does not differ from the natural voice. In fricatives, the difference between systems is less clearer, because both N and L show lowering. However, a greater magnitude of lowering can be observed in N, of 8.7 dB with a strongly significant effect.

5. Discussion

In the previous section, we saw a detailed description of results gathered from the two experiments. Spectral tilt can clearly be seen to show important differences for each of the phenomena under consideration. From Experiment I, it can be seen that HMM-R3 show increased spectral tilts, while HMM-R2 do not differ significantly. Similarly, in Experiment II, comparatively lower-rated K, and N showed increased spectral tilts, in the Hybrid-R1 and UnS-R2 groups, respectively. This is consistent with previous findings on flatter spectral tilt contributing to improved intelligibility [32]. Although there is little agreement on the relationship between naturalness and intelligibility, we find that spectral tilt appears to differentiate system-groups based on naturalness as well.

System-family specific results can also be observed on the basis of spectral tilt, and on consonantal duration. In HMM-R3 systems, spectral tilt increases from the natural voice. However, in low-quality UnS-R3 systems, it is seen to decrease more steeply. Therefore, spectral tilt exhibits quality-specific differences, but the influence is family-dependent. In terms of perceived speech quality, this indicates a preference for preserving the spectral tilt, and that deviation in either direction compromises quality.

Another important result can be seen is that UnS systems show differences based on quality in *durational* cues, while HMM systems on the other hand, impact spectral features more. It may be speculated here that statistical averaging practised in HMM systems, compromises the necessary variation required to retain spectral features. From these results, we can also speculate that the cost function of the unit selection systems favors shorter units over longer ones. A deeper investigation about which units have been selected would bring a better insight about the reason of this trend.

Finally, from Experiment II, we see important individual variation between UnS-R2 systems, and none whatsoever between HMM-R2 systems. While systems of HMM-R2 are more closely rated in naturalness and intelligibility, UnS-R2 have also received quite similar ratings [33]. Therefore, good-quality HMM systems rigidly approach statistical averaging and filter out variation between systems.

6. Conclusion

In this study, we have presented a comparative analysis of TTS systems from the BC-2013, using acoustic-phonetic measurements extracted from obstruent consonants. 10 systems from BC-2013 were grouped on the basis of their quality and family. A linear regression analysis was conducted to establish a relationship between system groups and acoustic measurements, with the natural voice as reference. Spectral tilt emerged as the most informative feature, where several different phenomena of quality, family and individual system differences could be observed. In general, better-rated systems were found to be associated with flatter spectral tilts, and higher RMS amplitude values for obstruents. These results were consistent with previous studies on improved intelligibility.

Avoiding the use of expensive behavioural equipment, we have been able to connect the domains of phonetics and speech technology. We have shown that the use of phonetic measurements is useful for a variety of comparison tasks, and the results are meaningful from a speech production and perception standpoint. For future work, we will incorporate transitional cues from adjacent vowels to gain deeper insights into the obstruent behaviour across different systems, especially for analyzing their concatenative ability. The dataset from BC-2013 will be extended to include neural voices built using systems such as Tacotron [34] and FastPitch [35]. A long-term goal of this approach is to identify more acoustic-phonetic features across different phonetic segments, including non-obstruent consonants, vowels and dipthongs.

A complete description of segmental properties of parallel synthetic speech can give speech synthesis researchers immediate feedback about the expectation of naturalness in their systems. These studies can precede subjective evaluation tests, by informing speech technologists about signal distortion at a segment and co-articulation level. Finally, from an acoustic-phonetic point of view, these studies allow us to understand phonemic properties that remain intact in the signal, despite a loss in naturalness.

7. Acknowledgements

This research has the financial support of Science Foundation Ireland under Grant number 18/CRT/6224. The ADAPT Centre for Digital Content Technology is funded under the SFI Research Centres Programme (Grant 13/RC/2106) and is co-funded under the European Regional Development Fund.

8. References

[1] J. Mendelson and M. P. Aylett, "Beyond the listening test: An interactive approach to TTS evaluation." in *International Conference on Speech Communication and Technology (Interspeech)*, 2017, pp. 249–253.

[2] R. Clark, H. Silen, T. Kenter, and R. Leith, "Evaluating long-form text-to-speech: Comparing the ratings of sentences and paragraphs," *arXiv preprint arXiv:1909.03965*, 2019.

[3] B. Patton, Y. Agiomyrgiannakis, M. Terry, K. W. Wilson, R. A. Saurous, and D. Sculley, "AutoMOS: Learning a non-intrusive assessor of naturalness-of-speech," *ArXiv*, vol. abs/1611.09207, 2016.

[4] F. Hinterleitner, *Quality of Synthetic Speech: Perceptual Dimensions, Influencing Factors, and Instrumental Assessment*. Springer, 2017.

[5] S. wei Fu, Y. Tsao, H.-T. Hwang, H.-M. Wang *et al.*, "Quality-net: An end-to-end non-intrusive speech quality assessment model based on BLSTM," in *International Conference on Speech Communication and Technology (Interspeech)*, 2018.

[6] H. Maki, S. Sakti, H. Tanaka, and S. Nakamura, "Quality prediction of synthesized speech based on tensor structured EEG signals," *PloS one*, vol. 13, no. 6, 2018.

[7] I. H. Parmonangan, H. Tanaka, S. Sakti, S. Takamichi, and S. Nakamura, "Speech quality evaluation of synthesized japanese speech using EEG," *International Conference on Speech Communication and Technology (Interspeech)*, pp. 1228–1232, 2019.

[8] A. Govender and S. King, "Using pupillometry to measure the cognitive load of synthetic speech," in *International Conference on Speech Communication and Technology (Interspeech)*, 2018, pp. 2838–2842.

[9] M. Cohn and G. Zellou, "Perception of concatenative vs. neural text-to-speech (tts): Differences in intelligibility in noise and language attitudes," in *International Conference on Speech Communication and Technology (Interspeech)*, 2020, pp. 1733–1737. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2020-1336

[10] T. Merritt, B. Putrycz, A. Nadolski, T. Ye, D. Korzekwa, W. Dolecki, T. Drugman, V. Klimkov, A. Moinet, A. Breen *et al.*, "Comprehensive evaluation of statistical speech waveform synthesis," in *2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2018, pp. 325–331.

[11] B. Lindblom and I. Maddieson, "Phonetic universals in consonant systems," *Language, speech and mind*, vol. 6278, 1988.

[12] N. Li and P. C. Loizou, "The contribution of obstruent consonants and acoustic landmarks to speech recognition in noise," *The Journal of the Acoustical Society of America*, vol. 124, no. 6, pp. 3947–3958, 2008.

[13] ——, "Factors affecting masking release in cochlear-implant vocoded speech," *The Journal of the Acoustical Society of America*, vol. 126, no. 1, pp. 338–346, 2009.

[14] ——, "Masking release and the contribution of obstruent consonants on speech recognition in noise by cochlear implant users," *The Journal of the Acoustical Society of America*, vol. 128, no. 3, pp. 1262–1271, 2010.

[15] S. A. Phatak, A. Lovitt, and J. B. Allen, "Consonant confusions in white noise," *The Journal of the Acoustical Society of America*, vol. 124, no. 2, pp. 1220–1233, 2008.

[16] J. Meyer, L. Dentel, and F. Meunier, "Speech recognition in natural background noise," *PloS one*, vol. 8, no. 11, p. e79279, 2013.

[17] F. Li and J. B. Allen, "Manipulation of consonants in natural speech," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 3, pp. 496–504, 2011.

[18] M. McAuliffe, M. Socolof, S. Mihuc, M. Wagner, and M. Sonderegger, "Montreal forced aligner: Trainable text-speech alignment using kaldi." in *International Conference on Speech Communication and Technology (Interspeech)*, 2017, pp. 498–502.

[19] M. Sonderegger and J. Keshet, "Automatic measurement of voice onset time using discriminative structured prediction," *The Journal of the Acoustical Society of America*, vol. 132, no. 6, pp. 3965–3979, 2012.

[20] T. Cho and P. Ladefoged, "Variation and universals in vot: evidence from 18 languages," *Journal of phonetics*, vol. 27, no. 2, pp. 207–229, 1999.

[21] B. H. Repp, "Closure duration and release burst amplitude cues to stop consonant manner and place of articulation," *Language and speech*, vol. 27, no. 3, pp. 245–254, 1984.

[22] A. Jongman, "Duration of frication noise required for identification of english fricatives," *The Journal of the Acoustical Society of America*, vol. 85, no. 4, pp. 1718–1725, 1989.

[23] E. Chodroff and C. Wilson, "Burst spectrum as a cue for the stop voicing contrast in american english," *The Journal of the Acoustical Society of America*, vol. 136, no. 5, pp. 2762–2772, 2014.

[24] A. Jongman, R. Wayland, and S. Wong, "Acoustic characteristics of english fricatives," *The Journal of the Acoustical Society of America*, vol. 108, no. 3, pp. 1252–1263, 2000.

[25] K. N. Stevens and S. E. Blumstein, "Invariant cues for place of articulation in stop consonants," *The Journal of the Acoustical Society of America*, vol. 64, no. 5, pp. 1358–1368, 1978.

[26] H. M. Sussman, H. A. McCaffrey, and S. A. Matthews, "An investigation of locus equations as a source of relational invariance for stop place categorization," *The Journal of the Acoustical Society of America*, vol. 90, no. 3, pp. 1309–1325, 1991.

[27] H. M. Sussman, D. Fruchter, and A. Cable, "Locus equations derived from compensatory articulation," *The Journal of the Acoustical Society of America*, vol. 97, no. 5, pp. 3112–3124, 1995.

[28] D. T. P. D. McCarthy, "The acoustics of place of articulation in english plosives," Ph.D. dissertation, Newcastle University, 2019.

[29] C. Redmon, "Lexical acoustics: Linking phonetic systems to the higher-order units they encode," *PhD dissertation, University of Kansas, Lawrence*, 2020.

[30] C. H. Shadle and S. J. Mair, "Quantifying spectral characteristics of fricatives," in *Proceeding of Fourth International Conference on Spoken Language Processing. ICSLP'96*, vol. 3. IEEE, 1996, pp. 1521–1524.

[31] L. L. Koenig, C. H. Shadle, J. L. Preston, and C. R. Mooshammer, "Toward improved spectral measures of/s: Results from adolescents," 2013.

[32] Y. Lu and M. Cooke, "The contribution of changes in f0 and spectral tilt to increased intelligibility of speech produced in noise," *Speech Communication*, vol. 51, no. 12, pp. 1253–1262, 2009.

[33] S. King and V. Karaiskos, "The blizzard challenge 2013," in *The Blizzard Challenge Workshop*, 2013, http://festvox.org/blizzard/bc2013/summary_Blizzard2013.pdf.

[34] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan *et al.*, "Natural TTS Synthesis by Conditioning Wavenet on MEL Spectrogram Predictions," in *international Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4779–4783.

[35] A. Łańcucki, "Fastpitch: Parallel text-to-speech with pitch prediction," *arXiv preprint arXiv:2006.06873*, 2020.

11th ISCA Speech Synthesis Workshop (SSW 11)
26-28 August 2021, Budapest, Hungary

Improving Polyglot Speech Synthesis through Multi-task and Adversarial Learning

Jason Fong[1*], *Jilong Wu*[2], *Prabhav Agrawal*[2], *Andrew Gibiansky*[2], *Thilo Koehler*[2], *Qing He*[2†]

[1]The University of Edinburgh
[2]Facebook AI

jason.fong@ed.ac.uk, {jilwu, prabhavag, gibiansky, tkoehler, qinghe}@fb.com

Abstract

It is still quite challenging for polyglot speech synthesis systems to synthesise speech with the same pronunciations and accent as a native speaker, especially when there are fewer speakers per language. In this work, we target an extreme version of the polyglot synthesis problem, where we have only one speaker per language, and the system has to learn to disentangle speaker from language features from just one speaker-language pair. To tackle this problem, we propose a novel approach based on a combination of multi-task learning and adversarial learning to help the model produce more realistic acoustic features for speaker-language combinations for which we have no data. Our proposed system improves the overall naturalness of synthesised speech achieving upto 4.2% higher naturalness over a multispeaker baseline. Our qualitative listening tests also demonstrate that system produces speech which sounds less accented and more natural to a native speaker.

Index Terms: TTS, speech synthesis, multilingual, multi-task learning, generative adversarial networks

1. Introduction

The holy grail of Multilingual TTS is to build a truly 'polyglot' system, which can synthesise native-sounding speech in multiple languages using *any* of its voices. This polyglot capability would enable simple sharing of voices from high resourced languages to low resourced ones, resulting in an overall improvement of synthesis quality for low-resourced languages due to transfer learning. However, existing systems are far from this goal, since existing systems either require using a parallel multilingual corpora, which is expensive to record, or fail to fully disentangle speaker from language in synthesised speech if trained on a dataset with only monolingual speakers. In this paper we pursue model-based improvements to multilingual TTS in the extreme scenario where only one speaker per language is available.

It is important to tackle the limitations of existing systems since doing so would enable applications previously not possible that are both inclusive and key to connecting people across the globe. For example, polyglot TTS systems can allow the creation of personal voices, friends & family voices, and even celebrity voices in languages not spoken by each respective person. This is very exciting in the case of voice assistants, where it allows users to receive the same voice experience while maintaining speaker identity across multiple languages. In the scenarios above, we are usually familiar with the speaker, which makes us skilled at recognising a speaker's identity, subsequently this makes the problem of maintaining speaker similarity even more challenging[1].

Both unit-selection [2] and deep neural network [3] based Multilingual TTS approaches have shown good results leveraging large parallel corpora, consisting of 1000s of utterances per language per speaker. Parallel corpora improve polyglot synthesis by providing a wide coverage of how a speaker identity would pronounce phones in each target language. However, such parallel corpora are costly or sometimes impossible to procure since voice talents speaking multiple languages are rare and almost non-existent if we go beyond the most-spoken languages. Furthermore, even if multilingual voice talents are available, their proficiencies in their languages are unlikely to all be at a native level as the authors of [3] found.

Subsequently newer approaches to polyglot TTS have focused on lessening the need for native-level parallel corpora. Some approaches have tried using cross-lingual voice cloning to augment monolingual recordings thereby creating artificial parallel datasets [4] but these approaches require explicit voice cloning models, which have faced issues with producing good quality cross-lingual output.

More recent approaches have sought to train using only monolingual corpora. The difficulty of training on only monolingual corpora however is that of speaker and language factor entanglement. Since speakers only speak one language, there is perfect correlation between speaker identity and language in the data, making factorisation difficult, and potentially resulting in the model ignoring the language conditioning feature. This is problematic however as an acoustic factorisation [5] of speaker identity and language must be obtained in order for a model to be able to then generate arbitrary combinations of speaker and language. The approaches of [6, 7, 8] attempt to achieve factorisation by representing speaker and language factors as distinct transformations that are then applied sequentially to input linguistic features. [9, 10] alternatively use speaker and language features to condition the decoder of seq2seq TTS systems and then attempt to achieve a speaker-language factorisation by training with multiple speakers from multiple languages. The advantage of this approach is that it forgoes the need for adding separate modules or layers for different speakers and languages.

In this paper, we focus on the problem of building polyglot TTS systems using solely monolingual corpora. Our main contribution is to further improve cross-lingual voice quality through the use of additional training losses and tasks. Our approach, in a similar vein as [9], uses an adversarial loss to improve multilingual performance, however we apply it to predicted acoustics to improve the realisation of acoustics in general and phones in particular.

Our model architecture is novel but is slightly similar in concept to [11] that uses a loss term to preserve speaker identity,

*Work performed while interning at Facebook AI.
† Correspondence to Qing He

10.21437/SSW.2021-30

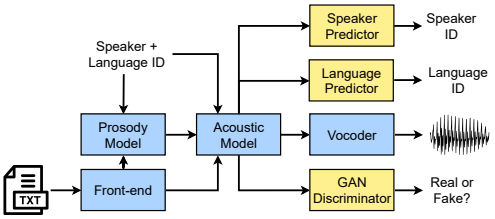


Figure 1: *Proposed model overview. Our baseline acoustic model is coloured blue, and proposed model additions are coloured in yellow.*

but this is performed over speaker embeddings only, whereas we do it using a multi-task speaker-language prediction task over predicted acoustics. They do this to avoid the problem of speaker embeddings also encoding language information. Their model doesn't use language-based conditioning features, and instead relies on using language specific text encoders. They train in a scheduled manner, first training the network to synthesise multilingual speech and then optimising the speaker embedding space for polyglot synthesis using unseen speaker-language combinations. In our model we do not perform such scheduling. [12] similarly tries to resolve language dependency in the speaker space by viewing cross-lingual TTS as a domain adaptation problem and attempt to learn a language independent speaker space.

The main contribution of this work is in improving the naturalness and quality of speech in a language foreign to the original voice talent. We demonstrate that multi-task learning over speaker and language features combined with a GAN inspired adversarial loss can be fruitful when little data is available but polyglot systems are required.

2. Proposed Multilingual Acoustic Model

At the core of our proposed model is a seq2seq acoustic model (AM) that predicts output vocoder features from input linguistic features concatenated with speaker/language one-hot vectors. To improve the AM's Speaker Language Factorisation (SLF), its core acoustic loss function is augmented with additional losses obtained from supplementary tasks. An overview of the losses and tasks in our proposed model is found in Figure 1 and the following subsections detail the AM and each of its augmentations. To keep the notation of our various losses clear we use the following notation: a loss L's subscript denotes which model it is used to help train, and its superscript denotes where it is obtained from.

Since the augmentations detailed in the subsequent subsections work with the AM's outputs and are not tied to our particular AM architecture they are subsequently likely usable with other AM architectures, such as transformers or feed forward networks for example.

2.1. Multilingual Acoustic model (AM)

The AM receives as input a series of T frame-wise linguistic features ($\mathbf{x}_{1:T}$) and is trained to output a corresponding series of frame-wise 'vocoder' features ($\hat{\mathbf{y}}_{1:T}$), such as MFCCs, f_0, and periodicity features that can be fed to a signal processing based vocoder [13, 14], or mel-spectrograms that can be used to condition a neural vocoder [15, 16, 17]. We define a forward pass through the acoustic model as follows: $\hat{\mathbf{y}}_{1:T} = AM(\mathbf{x}_{1:T})$. Further description of the linguistic and acoustic features used to train models for our experiments is deferred to Subsection 3.1.

The AM uses the encoder-decoder with multi-rate attention architecture of [18]. The encoder and decoder are both unidirectional single-layer LSTMs with 512 hidden dimensions. The decoder additionally uses a multi-rate attention mechanism to attend over the hidden states of three encoders, providing contextual information relevant to a particular decoder timestep by attending over frame, syllable, and word-level features.

The AM, is *primarily* trained using an acoustic L^2 loss between ground truth and predicted vocoder features. We denote this primary loss component as $L_{AM}^{acoustic} = \sum_{t=1}^{T} L^2(\mathbf{y}_t, \hat{\mathbf{y}}_t)$ where $\mathbf{y}_t$ is a particular frame of ground truth acoustics, and $\hat{\mathbf{y}}_t$ is its corresponding predicted frame. Additional loss components detailed in the following subsections are used with $L_{AM}^{acoustic}$ in order to obtain Equation 1 which is the final loss function used to update the AM's weights during training:

$$L_{AM} = L_{AM}^{acoustic} + \alpha L_{AM}^{MT} + \beta L_{AM}^{adv} \quad (1)$$

where $\alpha = 0.025$ and $\beta = 20.0$ are weights for each loss component discovered from hyperparameter search.

2.2. Speaker & Language Multi-task prediction heads

Along with the AM we train speaker and language multi-task (MT) prediction heads for one core reason: such prediction tasks serve as an inductive bias [19] that can encourage the AM to utilise speaker and language features. Both prediction heads use the *same* series of T frame-wise acoustic predictions from the AM to make a *downsampled* series of U categorical predictions over k classes (k_S speaker classes or k_L language classes).

The architecture of each prediction head consists of 5 1D convolutional layers each with 256 filters, a stride of 3, kernel size 5, and a padding of 2. Book-ending the 5 convolutional layers are two linear projection layers: an input layer projects features from Dim_{in} to Dim_{hid} dimensions, and an output layer projects features from Dim_{hid} to Dim_{out} dimensions. Additionally we apply a dropout of 0.2 to the input features before the first linear projection layer and each convolutional layer uses the Leaky ReLU [20] activation function with a leakiness of 0.2 and slope of -0.1.

We train the prediction heads' weights using a Cross-Entropy loss between their output logits and ground-truth one-hot targets. The loss for each prediction head is $L_{MT} = \sum_{u=1}^{U} CE(\mathbf{c}_u, \hat{\mathbf{c}}_u)$ where $\hat{\mathbf{c}}_{1:U} = MT(\hat{\mathbf{y}}_{1:T})$ represents speaker or language predictions obtained from passing predicted acoustics through the multi-task heads, $\mathbf{c}_{1:U}$ represents a corresponding series of one-hot ground truth classes, and $CE(.)$ is the Cross-Entropy loss function.

Note that we train the prediction heads using predicted acoustics $\hat{\mathbf{y}}_{1:T}$ rather than ground truth acoustics $\mathbf{y}_{1:T}$ in order to avoid train-test mismatch that can be caused by teacher-forcing.

By default we do *not* detach the acoustic predictions from the computation graph before feeding them to the multi-task heads so that L_{MT} also updates the AM's weights during training. Therefore we also refer to L_{MT} as L_{AM}^{MT}. We also experimented with detached multi-task losses, in which case L_{MT} does not update the AM, but found that in doing so our model does not improve over our baseline.

2.3. Adversarial training of AM

To complement the multi-task prediction heads we introduce a GAN discriminator that is trained to predict whether a series of acoustic features are either ground truth (*'real'*) or predictions generated by the AM (*'fake'*). We use the GAN discriminator to help ensure that the AM uses speaker/language inputs in a perceptual way rather than *cheating* by minimising L_{MT} in non-perceptual ways. That is, by encoding speaker and language information into the predicted acoustics in a acoustically non-perceivable way.

The architecture of the discriminator follows that of [21], consisting of 10 1D convolutional layers each with 128 filters, a stride of 1, kernel size 3, and a linearly increasing dilation rate (dilation increases by 1 per layer). Identical to the multi-task prediction heads detailed in Subsection 2.2 the discriminator's convolutional layers are each followed by LeakyRELU activation functions and are book-ended by linear projection layers. The final projection layer which projects from Dim_{hid} to Dim_{out}, where Dim_{out} is equal to 1, is followed by a Sigmoid activation function, collapsing the model's output to the range $[0, 1]$ and as such its output can be interpreted as the probability that the discriminator's input is real acoustic data.

To train the discriminator to differentiate between real and fake acoustics we adopt a two component loss $L_D = L_D^{real} + L_D^{fake}$. We train the discriminator to output 1 when it recognises real acoustics with $L_D^{real} = \sum_{t=1}^{T} L^2(\mathbf{r}_t, 1)$, and train it to output 0 when it recognises fake acoustics with $L_D^{fake} = \sum_{t=1}^{T} L^2(\mathbf{f}_t, 0)$ where $\mathbf{r}_{1:T} = D(\mathbf{y}_{1:T})$ and $\mathbf{f}_{1:T} = D(\hat{\mathbf{y}}_{1:T})$ are generated from the discriminator by feeding it ground truth and predicted acoustics respectively.

Finally we obtain from the discriminator an adversarial loss $L_{AM}^{adv} = \sum_{t=1}^{T} L^2(\mathbf{f}_t, 1)$ that is incorporated into the AM's loss function to help ensure its predicted acoustics are high quality and perceptually synthesise speaker and language. This loss is minimised when the AM successfully generates acoustics that fool the discriminator into believing that they are real.

2.4. Training loop

In this Subsection we define one iteration of the training loop for our proposed model.

1. Use inputs $\hat{\mathbf{x}}_{1:T}$ to get AM predictions $\hat{\mathbf{y}}_{1:T}$.
2. Use $\hat{\mathbf{y}}_{1:T}$ to a) get the acoustic loss $L_{AM}^{acoustic}$, b) get the GAN discriminator adversarial loss L_{AM}^{adv}, and c) calculate speaker and prediction losses through the multi-task heads to obtain L_{MT} and use this loss to train the heads.
3. Combine all of the AM's losses to get Equation 1 and use it to update the AM.
4. Use the inputs $\hat{\mathbf{x}}_{1:T}$ again to get a *new* set of AM predictions and use them to obtain L_D and train the GAN discriminator.

3. Experimental setup

To evaluate the efficacy of our proposed model, we perform a subjective listening test to compare its performance against two baseline models. A monospeaker baseline and a multispeaker baseline. This section describes the details of these experiments.

3.1. Input representations

The framewise input features used by our AM are obtained by up-scaling the output of our linguistic front-end. This up-scaling is performed using durations obtained from a prosodic model that predicts both the duration and f_0 of each phone aligned frame of contextual linguistic features.

In order to improve multilingual TTS performance by encouraging the model to share language-independent acoustic knowledge across languages, our front-end produces a *shared* phone representation common to all our languages. Previous work has approached this by using a phone set that is common across all languages [11]. Recent work [22] however uses 'phonological features' (PFs) as input to a neural TTS system. These PFs features have been shown to enable zero-shot multilingual TTS to unseen languages, and [23] also show that using PFs improves intelligibility and naturalness for low-resourced languages due to pooling of data, and pervasive sharing of encoder parameters across languages. Our model similarly uses multidimensional PFs to represent each phone. We start with a phone-set, which represents phonetic identity using the various dimensions for speech production such as place of articulation, and manner of articulation. This ensures that our baseline system can produce multilingual output of reasonable quality , without requiring an explicit mapping between phone-sets.

3.2. Modelling

Both our baseline and proposed acoustic models share the same core multi-rate attention architecture [18]. Acoustic or prosodic features are predicted for every frame by a recurrent LSTM module. Additionally contextual information at different levels relevant to producing a particular timestamp of acoustics is summarised from the entire input sequence by the multi-rate attention module. Previous experiments have found that the usage of multiple attention alignments overall improve prosody realisations from input linguistic features.

The acoustic models predict spectrum features, which is a 19-dim feature vector consisting of 1-dim f_0 vector, a 13-dim MFCC vector along with a 5-dim periodicity vector. Our conditional neural vocoder is a WaveRNN [16] model, with hidden dimension 1024. It takes in the 19-dim spectrum features and generates the audio waveform at 24kHz.

Our AMs and vocoder are additionally made multilingual via the use of speaker and language one-hot conditioning features. In this work we use one-hot features rather than speaker embeddings as in this study we focus on improving polyglot synthesis, rather than enabling multilingual synthesis for new unseen speakers, which we leave as potential future work.

3.3. Training setup & Data

Our acoustic models are trained with the Adam optimizer with a learning rate of 1e-4. We implemented them using Pytorch and conduct the training with distributed GPU clusters. After some fine-tuning, we decide to train at 500K steps with a training time of approximately 2 days using batch size of 32.

The TTS datasets were recorded in a voice production studio by contracted professional voice talents. Our multilingual dataset `5lang-5speaker` contains five voices each speaking a different language: English (30 hours), Spanish (23 hours), Italian (9 hours), German (8 hours) and French (10 hours) and the data was collected at a 24kHz sampling rate. `5lang-5speaker` is used to train both the baseline and proposed multilingual AMs, and our multilingual multi-

speaker WaveRNN. We additionally use each individual voice in `5lang-5speaker` to train monospeaker baseline AMs that can still perform some level of multilingual TTS due to our use of language-independent phonological features.

3.4. Evaluation

We have designed our listening tests to answer one question regarding our proposed AM vs baseline AMs: does adding speaker and language prediction tasks along with adversarial training improve the overall naturalness of speech when synthesising polyglot 'non-native' speech.

We synthesised each language's test set conditioning using a *non-native* speaker, that is a monolingual speaker whom has no data in that particular language. In other words in our experiment we examine how well *each* of our dataset's speakers perform at 'non-native' polyglot synthesis. We use the following speaker and language combinations for generating our *non-native* test sets: S_{ES}-T_{EN}, S_{DE}-T_{ES}, S_{IT}-T_{FR}, S_{FR}-T_{DE}, S_{EN}-T_{IT}. For clarification EN is English, ES is Spanish, FR is French, IT is Italian, and DE is German. Also S_{ES} refers to our Spanish speaker and T_{EN} refers to our English test set.

Using a crowdsourcing platform we recruited the following number of participants for each test set language: 349 English, 214 Spanish, 39 French, 300 Italian, and 61 German. Participants are all native speakers of the language that they are rating. Each participant is shown 50 stimuli from that language and are asked to rate them from 1 to 5 in terms of naturalness as a voice assistant. We use these ratings to obtain an averaged naturalness MOS for each system.

3.5. Voice training and inference

We trained a total of 7 AMs for submission to listening tests: 5 monospeaker baselines, 1 multispeaker baseline, and 1 proposed multispeaker model. They are each trained with the following data and hyper-parameters:

- $\texttt{B}_{mono}$: We train 5 monospeaker baselines each one trained using a single native dataset as described in Subsection 3.3.
- $\texttt{B}_{multi}$: We train a single multispeaker baseline using the `5lang-5speaker` dataset.
- $\texttt{P}_{multi}$: We train a single multispeaker proposed model using the `5lang-5speaker` dataset. It differs from $\texttt{B}_{multi}$ with its use of speaker and language prediction tasks with adversarial loss during training.

To generate listening test stimuli for our subjective evaluations we use the *non-native* speaker and language combinations discussed in Subsection 3.4 to condition each multispeaker model in order to generate the test set that matches the language. The monospeaker models generate a non-native language for which it never saw any training data. For example $\texttt{B}_{EN}$ is used to generate the Italian test set, even though it used only English data during training. Again this is made possible by our model's use of phonological features rather than language specific phone-sets. A selection of samples used in our listening test can be found on our webpage for this paper[1].

4. Results

A summary of our MOS listening test results can be found in Figure 2. We observe several clear trends across the three types of systems: First of all $\texttt{B}_{multi}$ consistently out-performs $\texttt{B}_{mono}$, suggesting that training acoustic models with data from multiple speakers and languages is beneficial even given we only have one speaker per language. Secondly, except from the S_{EN}-T_{IT} stimuli, $\texttt{P}_{multi}$ consistently out-performs $\texttt{B}_{multi}$, suggesting that our proposed model modifications make an improvement in both quality and naturalness. The largest gains from using our proposed model are seen with S_{FR}-T_{DE} where naturalness is improved by 4.2 % over the multispeaker baseline. When listening to the test set stimuli we discovered that our proposed model also makes improvements in how native each utterance sounds and in phone intelligibility. We include examples on our samples page reflecting these findings.

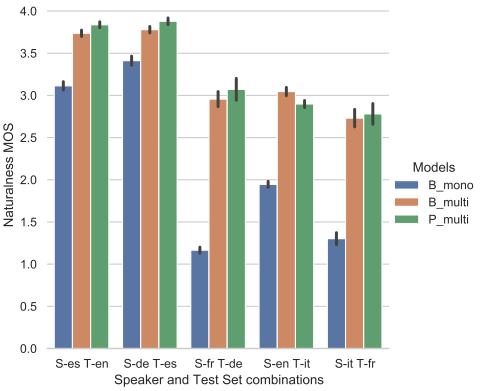


Figure 2: *Mean opinion scores obtained from our subjective listening test described in Subsection 3.4. 95% confidence intervals are depicted as black lines. Colours of the bars refer to one of three model types: monospeaker baseline, multispeaker baseline, and multispeaker proposed model. Further details regarding these models can be found in 3.5.*

5. Conclusions

In this work, we have proposed a novel way to improve polyglot speech synthesis across five languages through adversarial learning and multi-task training. According to a MOS study using on average 200 native raters per language, our proposed model achieved better overall quality compared with a multispeaker and multilingual baseline. As for future work, we plan to extend the proposed idea to prosodic modelling and combine it together with our proposed acoustic model. Another direction we also would like to pursue is using data augmentation methods to further improve the overall quality using synthetic polyglot data.

[1] https://multilingual-tts.github.io/samples/

6. References

[1] J. Kreiman and G. Papcun, "Comparing discrimination and recognition of unfamiliar voices," *Speech Communication*, vol. 10, no. 3, pp. 265–275, 1991.

[2] M. Chu, H. Peng, Y. Zhao, Z. Niu, and E. Chang, "Microsoft mulan-a bilingual tts system," in *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03).*, vol. 1. IEEE, 2003, pp. I–I.

[3] H. Ming, Y. Lu, Z. Zhang, and M. Dong, "A light-weight method of building an lstm-rnn-based bilingual tts system," in *2017 International Conference on Asian Language Processing (IALP)*. IEEE, 2017, pp. 201–205.

[4] S. Zhao, T. H. Nguyen, H. Wang, and B. Ma, "Towards natural bilingual and code-switched speech synthesis based on mix of monolingual recordings and cross-lingual voice conversion," *arXiv preprint arXiv:2010.08136*, 2020.

[5] M. Gales, "Acoustic factorisation," in *IEEE Workshop on Automatic Speech Recognition and Understanding, 2001. ASRU'01.* IEEE, 2001, pp. 77–80.

[6] H. Zen, N. Braunschweiler, S. Buchholz, M. J. Gales, K. Knill, S. Krstulovic, and J. Latorre, "Statistical parametric speech synthesis based on speaker and language factorization," *IEEE transactions on audio, speech, and language processing*, vol. 20, no. 6, pp. 1713–1724, 2012.

[7] B. Li and H. Zen, "Multi-language multi-speaker acoustic modeling for lstm-rnn based statistical parametric speech synthesis," 2016.

[8] Y. Fan, Y. Qian, F. K. Soong, and L. He, "Speaker and language factorization in dnn-based tts synthesis," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 5540–5544.

[9] Y. Zhang, R. J. Weiss, H. Zen, Y. Wu, Z. Chen, R. Skerry-Ryan, Y. Jia, A. Rosenberg, and B. Ramabhadran, "Learning to speak fluently in a foreign language: Multilingual speech synthesis and cross-language voice cloning," *arXiv preprint arXiv:1907.04448*, 2019.

[10] Amazon, "English-language Alexa voice learns to speak Spanish," 2021. [Online]. Available: https://www.amazon.science/blog/english-language-alexa-voice-learns-to-speak-spanish

[11] E. Nachmani and L. Wolf, "Unsupervised polyglot text-to-speech," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 7055–7059.

[12] D. Xin, Y. Saito, S. Takamichi, T. Koriyama, and H. Saruwatari, "Cross-lingual text-to-speech synthesis via domain adaptation and perceptual similarity regression in speaker space," *Proc. Interspeech 2020*, pp. 2947–2951, 2020.

[13] H. Kawahara, "Straight, exploitation of the other aspect of vocoder: Perceptually isomorphic decomposition of speech sounds," *Acoustical science and technology*, vol. 27, no. 6, pp. 349–353, 2006.

[14] M. Morise, F. Yokomori, and K. Ozawa, "World: a vocoder-based high-quality speech synthesis system for real-time applications," *IEICE TRANSACTIONS on Information and Systems*, vol. 99, no. 7, pp. 1877–1884, 2016.

[15] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[16] N. Kalchbrenner, E. Elsen, K. Simonyan, S. Noury, N. Casagrande, E. Lockhart, F. Stimberg, A. Oord, S. Dieleman, and K. Kavukcuoglu, "Efficient neural audio synthesis," in *International Conference on Machine Learning*. PMLR, 2018, pp. 2410–2419.

[17] K. Kumar, R. Kumar, T. de Boissiere, L. Gestin, W. Z. Teoh, J. Sotelo, A. de Brébisson, Y. Bengio, and A. Courville, "Melgan: Generative adversarial networks for conditional waveform synthesis," *arXiv preprint arXiv:1910.06711*, 2019.

[18] Q. He, Z. Xiu, T. Koehler, and J. Wu, "Multi-rate attention architecture for fast streamable text-to-speech spectrum modeling," *arXiv preprint arXiv:2104.00705*, 2021.

[19] R. Caruana, "Multitask learning," *Machine learning*, vol. 28, no. 1, pp. 41–75, 1997.

[20] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. icml*, vol. 30, no. 1. Citeseer, 2013, p. 3.

[21] A. Polyak, L. Wolf, Y. Adi, and Y. Taigman, "Unsupervised cross-domain singing voice conversion," *arXiv preprint arXiv:2008.02830*, 2020.

[22] M. Staib, T. H. Teh, A. Torresquintero, D. S. R. Mohan, L. Foglianti, R. Lenain, and J. Gao, "Phonological features for 0-shot multilingual speech synthesis," *arXiv preprint arXiv:2008.04107*, 2020.

[23] A. Gutkin, "Uniform multilingual multi-speaker acoustic model for statistical parametric speech synthesis of low-resourced languages," 2017.

11th ISCA Speech Synthesis Workshop (SSW 11)
26-28 August 2021, Budapest, Hungary

Multi-Scale Spectrogram Modelling for Neural Text-to-Speech

Ammar Abbas, Bajibabu Bollepalli, Alexis Moinet, Arnaud Joly, Penny Karanasou, Peter Makarov, Simon Slangens, Sri Karlapati, Thomas Drugman

Alexa AI, Cambridge, United Kingdom
{syeabbs, bajibabb, drugman}@amazon.com

Abstract

We propose a novel Multi-Scale Spectrogram (MSS) modelling approach to synthesise speech with an improved coarse and fine-grained prosody. We present a generic multi-scale spectrogram prediction mechanism where the system first predicts coarser scale mel-spectrograms that capture the suprasegmental information in speech, and later uses these coarser scale mel-spectrograms to predict finer scale mel-spectrograms capturing fine-grained prosody. We present details for two specific versions of MSS called *Word-level MSS* and *Sentence-level MSS* where the scales in our system are motivated by the linguistic units. The Word-level MSS models word, phoneme, and frame-level spectrograms while Sentence-level MSS models sentence-level spectrogram in addition. Subjective evaluations show that Word-level MSS performs statistically significantly better compared to the baseline on two voices.

Index Terms: neural text-to-speech, multi-scale spectrogram, word-level, sentence-level

1. Introduction

Over the last few years, the progress in Text-To-Speech (TTS) technology has been astounding. Specifically, neural models such as Wavenet [1] and Tacotron [2] have revamped all the components of a modern TTS system [3]. Due to this, the Neural Text-To-Speech (NTTS) has become a standard paradigm where a neural sequence-to-sequence (seq2seq) model is employed to map the input text into acoustic features, and a neural vocoder model is employed to convert the acoustic features into a corresponding waveform. These NTTS systems are capable of generating high-quality speech that is often indistinguishable from human speech [4].

However, NTTS systems still struggle to produce speech with appropriate prosody compared to human speech [5]. The perceived prosody in the synthesized speech may sound inappropriate given the textual context, and includes problems such as wrong type of intonation, pausing, or emphasis. The lack of appropriate prosody stems from multiple reasons ranging from the model design to the way data is processed. Although the prosody is a suprasegmental phenomenon, the existing NTTS systems are designed in a way that they take fine textual representations such as phonemes as an input and predict finer-level acoustic representations such as mel-spectrograms as an output. Thus, the speech produced by the NTTS system can sound flat and require more cognitive effort to process it [6].

Numerous studies have been proposed in the literature to address the aforementioned issues [7, 8, 9, 10]. Most of these studies focus on modelling a latent representation space of prosody using a separate encoder called reference encoder [7]. The reference encoder guides the prosody of the output speech signal to generate expressive speech. The reference encoder can be designed in a variational [11] or non-variational [7] style. Moreover, the latent embedding vectors encoded by reference encoder can be represented either at a coarser level e.g. sentence [8] or at a finer level e.g. word or phonemes [9, 10].

Along with these latent representation based methods, another set of studies focus on modelling prosody in a hierarchical manner along with a reference encoder [12, 13, 14]. Here the input text is represented at various levels that are spanning from coarser (e.g., sentences) to finer (e.g., phonemes) levels. Kenter et al. [13] proposed such kind of hierarchical approach to model prosodic features such as F0, energy, and duration, and these prosodic features along with linguistic features are utilized by a neural vocoder to render the final speech waveform. However, one of the shortcomings of the prosody modelling studies based on latent representations is that they use reference mel-spectrograms to learn prosody embeddings during training, whereas during the inference time they either rely on textual based features to sample [8] or select [15] a prosody embedding from a set of pre-existing latent embeddings. This results in a mismatch between training and inference which could lead to an inappropriate prosody in the output speech signal.

Instead of modelling the prosodic latent space, few studies predict the conventional prosodic features (e.g., F0 and energy) in a multi-task manner and utilize those features to control the prosody of the synthesized speech [16, 17]. The performance of these methods depends on the accuracy and robustness of prosodic feature extraction and modelling. However, F0 extraction and modelling is generally prone to a number of errors [18].

Complementing to the aforementioned studies, in this paper, we propose a Multi-Scale Spectrogram (MSS) modelling technique to capture short and long-range dependencies observed in the speech signal. In MSS, the mel-spectrograms are predicted sequentially from a coarser scale capturing higher-level representation of speech to a finer scale capturing fine-grained prosodic details. Each subsequent finer scale is conditioned by the previous scale's predicted mel-spectrograms. This allows the MSS modelling to produce prosody appropriate at different linguistic units such as sentence, word and phonemes, thereby improving the overall naturalness of the NTTS systems.

Similar to our proposed approach, Vasquez et al. [19] predict the mel-spectrograms in a multi-scale manner. They initially predict lower resolution mel-spectrograms and progressively increase their resolution by 2 times at each scale irrespective of the semantic units in language or acoustic units in speech. Contrary to that, we provide a generic multi-scale mechanism to represent mel-spectrograms and later develop two specific MSS systems where the scales correspond to linguistic units which are sentences, words, and phonemes. Moreover, each scale in MSS has its own objective to learn and all scales are learned together by multi-task learning [20].

Another major difference between [19] and our approach is the use of explicit duration modelling instead of an attention mechanism to find alignment between text and speech.

10.21437/SSW.2021-31

Due to the stability issues of the attention mechanism of neural seq2seq models in the NTTS systems, the synthesized speech signals could have unpleasant artifacts such as mumbling, repetitions, or skipping [21]. To mitigate the stability issues, non-attentive neural seq2seq models have recently become popular, where the attention mechanism is replaced by an explicit duration model [8, 22, 17]. Hence, the non-attentive neural seq2seq model based NTTS system is employed in this paper.

Our main contributions are as follows: i) We propose a novel multi-scale mel-spectrogram modelling technique to improve the overall quality and naturalness of NTTS systems by appropriately capturing the coarse and fine-grained prosody; ii) We conduct and present an ablation study on two specific versions of MSS, called Sentence-level MSS and Word-level MSS. The Word-level MSS models word, phoneme, and frame-level spectrograms while Sentence-level MSS models sentence-level spectrogram additionally; iii) We evaluate Word-level MSS against a baseline system that is based on external duration model and show that it is significantly better than the baseline on two voices.

2. The Baseline

We use the same baseline system as in [8], which is a modified version of DurIAN [23]. Figure 1 illustrates the block diagram of our baseline system. It is composed of two major components: a seq2seq model and a duration model. First, the input text containing W words $\boldsymbol{w} = [w_0, w_1, \ldots, w_{W-1}]$ is passed through the front-end to extract P phonemes $\boldsymbol{p} = [p_0, p_1, \ldots, p_{P-1}]$ as an output. Next, the P phonemes are passed through an encoder, which captures the relations between phonemes, and produces P phoneme embeddings as an output. The encoder is composed of 1D convolutions followed by a bidirectional LSTM. Finally, the decoder takes these P phoneme embeddings and P phoneme durations in frames $\boldsymbol{d} = [d_0, d_1, \ldots, d_{P-1}]$ where $\sum_{d \in \boldsymbol{d}} = T$ as an input, and predicts T mel-spectrogram frames $\mathbf{Y} = [\boldsymbol{y}_0, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_{T-1}]$ auto-regressively as an output. There is no post-net after decoder as it resulted in instabilities when training with a reduction factor of 1.

During training, the phoneme durations are obtained from a forced-alignment algorithm, whereas during inference, the phoneme durations are predicted by a duration model trained separately. The duration model takes P phonemes as an input and predicts P durations. Both the acoustic and the duration model are optimized using an L2 loss function.

3. Multi-Scale Spectrogram (MSS) modelling

This section introduces the proposed MSS modelling technique. As shown in Figure 1, the decoder of the baseline system predicts the mel-spectrogram frames directly from phoneme embeddings using phoneme durations. Contrary to this, the decoder based on MSS modelling technique predicts the mel-spectrogram frames after conditioning them on all the higher-level mel-spectrograms as illustrated in Figure 2. Specifically, the MSS modelling technique first predicts the mel-spectrogram vectors representing speech on a coarser scale which are later used for the prediction of mel-spectrogram vectors representing speech at a finer scale. The coarser scale representation captures most of the suprasegmental aspects of the prosody resulting in a more appropriate prosody for the given text. In principle, these scales can be defined in both time and frequency axes of the

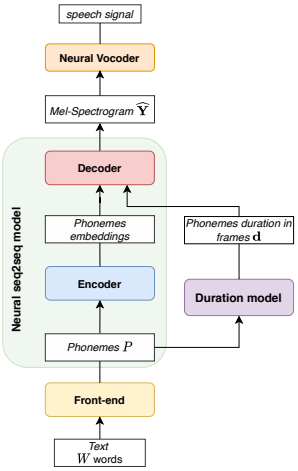


Figure 1: *Block diagram showing the architecture of our baseline. The decoder module colored in red is substituted by a multi-scale decoder in the proposed MSS modelling technique.*

mel-spectrogram. However, in this paper, the scales are defined only along the time axis while keeping the number of mel-bins constant ($= 80$) along the frequency axis. Extending the scales to the frequency axis is left as future work.

3.1. Generic multi-scale representations

Before moving to modelling, we first discuss how to construct targets for learning a generic multi-scale model. Let us assume that there are in total $L+1$ scales in the MSS modelling. At each scale l where $0 \leq l \leq L$, we compute a mel-spectrogram $\mathbf{S}^l = [\boldsymbol{s}_0^l, \boldsymbol{s}_1^l, \ldots, \boldsymbol{s}_{N_l-1}^l]$ of dimension $N_l \times 80$ from the ground-truth mel-spectrogram $\mathbf{Y} = [\boldsymbol{y}_0, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_{T-1}]$ of dimension $T \times 80$. Here, the L^{th} scale (the highest level representation of speech) has the least number of mel-spectrogram vectors, and their number progressively increases on each subsequent scale

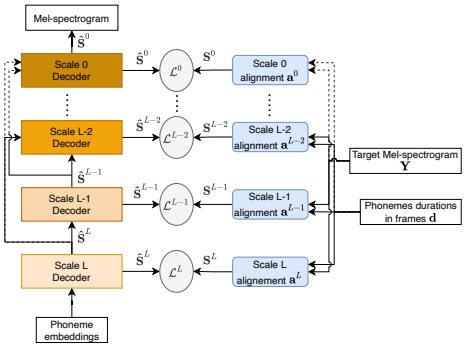


Figure 2: *Block diagram of the generic MSS decoder with $L+1$ scales. The scale 0 decoder depends upon the outputs of all the previous scale decoders.*

$l < L$ until the last scale 0 such that $N_L < N_{L-1} < \cdots < N_0 = T$. For example in Sentence-level MSS, $N_L = 1$, which is the number of sentences in an utterance.

We compute the mel-spectrogram $\mathbf{S}^l$ at scales $l > 0$ using the following equation:

$$\mathbf{S}^l = [\boldsymbol{s}_0^l, \boldsymbol{s}_1^l, \ldots, \boldsymbol{s}_{N_l-1}^l],$$

where each

$$\boldsymbol{s}_i^l = \begin{cases} \dfrac{1}{a_i^l} \displaystyle\sum_{j=c_{i-1}^l}^{c_i^l} \boldsymbol{y}_j, & 1 \leq i < N_l, \\ \dfrac{1}{a_i^l} \displaystyle\sum_{j=0}^{c_i^l} \boldsymbol{y}_j, & i = 0, \end{cases} \tag{1}$$

$$c_k^l = \sum_{i=0}^{k} a_i^l \tag{2}$$

In Eq. 1, $\boldsymbol{a}^l = [a_0^l, a_1^l, \ldots, a_{N_l-1}^l]$ is an alignment vector that denotes the alignments at scale $l > 0$ where each a_i^l represents the number of mel-spectrogram frames of $\mathbf{Y}$ corresponding to the spectrogram $\boldsymbol{s}_i^l$ at scale l. Thus the total sum of $\sum \boldsymbol{a}^l = T$. The alignment vectors a_i^l can be computed based on the definition of each scale (cf. Section 3.2). The vector $\boldsymbol{c}^l = [c_0^l, c_1^l, \ldots, c_{N_l-1}^l]$ is the cumulative sum of alignment vector $\boldsymbol{a}^l$, and each element c_k^l corresponds to the starting frame for token $k-1$ and ending frame for token k. So each target vector $\boldsymbol{s}_i^l$ at $l > 0$ is computed by taking the mean of mel-spectrogram frames $\mathbf{Y}$ from index c_{i-1}^l to c_i^l. The mel-spectrogram $\mathbf{S}^0$ at 0th scale of MSS modelling technique is equal to the target mel-spectrogram $\mathbf{Y}$ i.e. $\mathbf{S}^0 = \mathbf{Y}$, thus $N_0 = T$. This paper considers two specific cases of MSS modelling technique for validating our proposed approach: 1) *Sentence-level MSS* and 2) *Word-level MSS*, named after the coarser used linguistic unit.

3.2. Word-level MSS and Sentence-level MSS

The Word-level MSS has a total number of three scales ($L = 2$) where the 1st and 2nd scales correspond to the linguistic unit of phonemes and words respectively. The 0th scale corresponds to frame-level mel-spectrogram as discussed above. The number of mel-spectrogram vectors at each scale are given as such: $N_2 =$ number of words in a given utterance, $N_1 =$ number of phonemes present in an utterance, and $N_0 = T$. In Sentence-level MSS, $L = 3$ and there is an additional 3rd scale that corresponds to sentences. The $N_3 =$ number of sentences in a given utterance, which is equal to 1 in our case. In both these specific systems, the alignment vectors $\boldsymbol{a}^l$ are obtained from the phoneme durations and relations between phoneme, word, and sentences, which are obtained by the front-end. More specifically, in Sentence-level MSS, the alignment vector $\boldsymbol{a}^1$ is equal to phoneme durations $\boldsymbol{d}$ in frames. The alignment vector $\boldsymbol{a}^2$ is equal to the word durations in frames and $\boldsymbol{a}^3$ is equal to the sentence duration in frames.

The multi-scale representations that need to be modelled in Sentence-level MSS are $\mathbf{S} = [\mathbf{S}^0, \mathbf{S}^1, \mathbf{S}^2, \mathbf{S}^3]$. To model 3rd scale (sentence-level) mel-spectrogram $\hat{\mathbf{S}}^3$, we first project P phoneme embeddings into a sentence-level vector by taking the last hidden state of the LSTM encoder. Later, the sentence-level vector is passed through a sequence of 1D convolutions to obtain sentence-level mel-spectrogram $\hat{\mathbf{S}}^3$. The loss function at the 3rd scale (sentence-level) is defined as:

$$\mathcal{L}^3 = \left\| \hat{\mathbf{S}}^3 - \mathbf{S}^3 \right\|_2 \tag{3}$$

The $\hat{\mathbf{S}}^3$ mel-spectrogram is assumed to capture the sentence-level acoustic properties such as speaker-identity, recording environment, or speaking style of the sentence.

Similarly, to model the 2nd scale (word-level) mel-spectrogram $\hat{\mathbf{S}}^2$, we first project phonemes of each word into a word-level vector. Later, the word-level vectors are concatenated with the upsampled sentence-level mel-spectrogram $\hat{\mathbf{S}}_{\uparrow}^3$. $\hat{\mathbf{S}}_{\uparrow}^3$ is computed by upsampling the predicted sentence-level mel-spectrogram $\mathbf{S}^3$ to have the same dimension as $\mathbf{S}^2$ using an alignment defined between 2nd and 3rd scale, i.e. between words and the sentence they belong to. The loss function on 2nd scale is defined as:

$$\mathcal{L}^2 = \left\| \hat{\mathbf{S}}^2 - \mathbf{S}^2 \right\|_2 \tag{4}$$

$\hat{\mathbf{S}}^2$ is assumed to capture the word level acoustic properties such as word prominence, rise and fall of pitch and energy at word level.

We follow the same procedure to predict the phoneme-level mel-spectrogram $\hat{\mathbf{S}}^1$ and final frame-level mel-spectrogram $\hat{\mathbf{S}}^0$ or $\hat{\mathbf{Y}}$. In each of the scales l, the predicted mel-spectrograms are also conditioned on all the coarser scale predicted mel-spectrograms. The loss function at each scale l is defined as $\mathcal{L}^l = \left\| \hat{\mathbf{S}}^l - \mathbf{S}^l \right\|_2$. The sentence-level MSS is trained by minimizing the loss at all scales. So the training loss for the whole system is defined as:

$$\mathcal{L} = \mathcal{L}^0 + \mathcal{L}^1 + \mathcal{L}^2 + \mathcal{L}^3, \tag{5}$$

which can be interpreted as maximizing the likelihood of mel-spectrograms at all scales:

$$\begin{aligned} p(\hat{\mathbf{S}}^3, \hat{\mathbf{S}}^2, \hat{\mathbf{S}}^1, \hat{\mathbf{S}}^0) =& p(\hat{\mathbf{S}}^3).p(\hat{\mathbf{S}}^2|\hat{\mathbf{S}}^3).p(\hat{\mathbf{S}}^1|\hat{\mathbf{S}}^3, \hat{\mathbf{S}}^2) \\ & .p(\hat{\mathbf{S}}^0|\hat{\mathbf{S}}^3, \hat{\mathbf{S}}^2, \hat{\mathbf{S}}^1) \end{aligned} \tag{6}$$

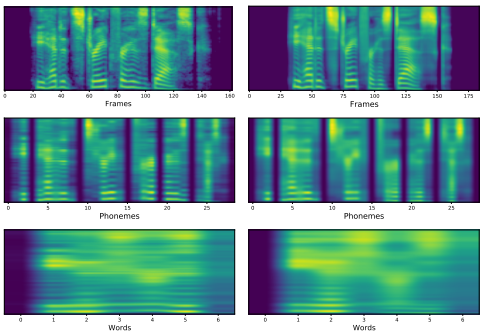
4. Experiments

4.1. Data

The experiments were carried out on an internal voice dataset that was recorded by two native US-English female voice talents. We refer to them as speaker-A and speaker-B. The training and test sets for speaker-A are 33 and 5.5 hours respectively, while they are 32 and 3 hours for speaker-B respectively. The test set is reserved for testing in this and future research studies. The sampling rate of the recorded audio is 24kHz. We extracted 80 band mel-spectrograms with a frame shift of 12.5ms.

4.2. Training and inference

As mentioned in Section 3.2, we have developed two systems based on MSS modelling technique: Sentence-level MSS and Word-level MSS. In both systems: 1) we train the seq2seq model according to the loss defined in Equation 5 where oracle alignments $\boldsymbol{a}^l$ are provided to the model at each scale l ; 2) we train the duration model using L2 loss as shown in Section 2. We optimize the loss in both the models using Adam optimizer [24] with different learning rates. We use a learning rate of 10^{-3} and 10^{-4} for the acoustic and the duration model respectively.



(a) *Oracle spectrogram* (b) *Predicted spectrogram*

Figure 3: *Visualisation of mel-spectrograms at different scales in Word-level MSS given the text: "He headed straight for his desk.". Left panel: oracle spectrograms, right panel: predicted spectrograms. The mel-spectrograms in bottom, middle, and top row correspond to 2nd (word), 1st (phoneme), and 0th (frame) scale respectively.*

During inference, we follow these 2 steps in the following order: I) we predict durations $\hat{\boldsymbol{d}}$ from the duration model trained in step 2; II) we generate mel-spectrograms $\hat{\mathbf{S}}^0$ from the seq2seq model trained in step 1 using the predicted durations $\hat{\boldsymbol{d}}$ from step I. We use Wavenet vocoder [1] to synthesize speech from $\hat{\mathbf{S}}^0$.

4.3. Evaluations

4.3.1. Qualitative evaluation of predicted and target spectrograms on different scales

Figure 3 shows the target (left column) and predicted (right column) mel-spectrograms at different scales l in the Word-level MSS for speaker-B as an example. The bottom row shows the 2^{nd} scale (word-level) mel-spectrogram $\mathbf{S}^2$. There are 7 mel-spectrogram vectors $[\boldsymbol{s}_0^2, \boldsymbol{s}_1^2, \ldots, \boldsymbol{s}_6^2]$ which represent coarse-grained prosodic features such as prominence at word-level. We can observe the harmonics and energy, and how they vary after each word. The middle row shows the phoneme-level mel-spectrogram $\mathbf{S}^1$. On this scale, there are 29 mel-spectrogram vectors $[\boldsymbol{s}_0^1, \boldsymbol{s}_1^1, \ldots, \boldsymbol{s}_{28}^1]$ which represent fine-grained prosodic features at phoneme-level. At this scale, we can see how the prosody varies around phonemes and can identify the stress on phonemes based on their acoustic representations. The top row shows frame-level mel-spectrogram $\mathbf{S}^0$, which is equal to the target spectrogram $\mathbf{Y}$. When comparing the target (left column) to the predicted (right column) mel-spectrograms, the Word-level MSS is able to capture the high-level prosodic features at 1^{st} and 2^{nd} scale, albeit with a smoother representation due to the nature of the L2 loss used in eq. 5.

4.3.2. Ablation study

A MUSHRA [25] evaluation was conducted on speaker-A to evaluate how the number/definition of scales affects the performance of MSS modelling. For the MUSHRA evaluation, we selected the following three systems: the baseline from Section 2, Word-level MSS, and Sentence-level MSS. A total of 50 utterances were selected randomly from the test set, and the duration of each utterance was approximately 15 seconds. Each utterance was rated by 24 native US-English professional listeners. Figure 4 presents the results of the MUSHRA evaluation. We have used a pairwise two-sided Wilcoxon signed-rank test corrected for multiple comparisons to measure statistical significance between the systems. The Word-level MSS system performed statistically significantly better than both the other systems (p-value $< 10^{-6}$) while there was no statistically significant difference between the baseline and sentence-level MSS systems (p-value $= 0.27$). We believe that during training, the Sentence-level MSS system is overfitting on the sentence-level spectrogram $\hat{\mathbf{S}}^3$ prediction. Due to this, it fails to capture coarse-grained prosodic features observed in the target $\mathbf{S}^3$, thus adversely affecting the prediction of spectrograms in lower scales $l < L$ which are conditioned on $\hat{\mathbf{S}}^3$. Moreover, an utterance in our data corresponds to a sentence which makes the prediction of $\hat{\mathbf{S}}^{\mathbf{3}}$ even more difficult because it does not have the surrounding sentences as an input to the system unlike scales $l < L$. These reasons could suggest why there is no improvement in the Sentence-level MSS system compared to the baseline.

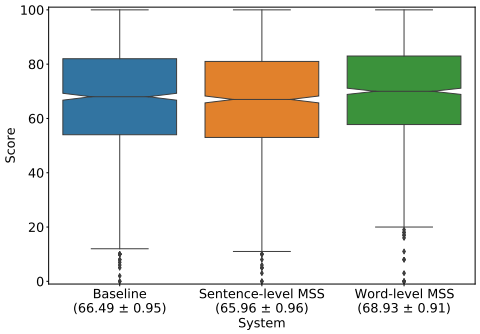
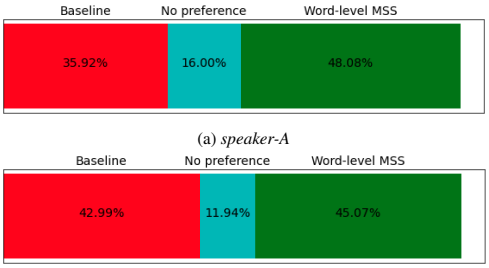


Figure 4: *Results of the MUSHRA evaluation of ablation study on speaker-A voice. Mean rating and 95% confidence intervals are reported below system names.*

4.3.3. Preference tests

As the Word-level MSS system showed a significant improvement in the ablation study, we have selected it for further comparisons with the baseline system. A preference test was conducted on speaker-A and speaker-B voices. For speaker-A, 50 utterances were selected randomly from the test set and each utterance had a duration of approximately 15 seconds. Similar to the MUSHRA evaluation, we used a third-party vendor to complete the test, and a total 24 listeners participated. The results are shown in Figure 5a. We use a binomial significance test to measure the statistical significance. The Word-level MSS is found to be statistically significantly better than the baseline (p-value $< 10^{-4}$)

For speaker-B, 100 utterances were selected randomly from the test set and the duration of each utterance was approximately 15 seconds. A total of 48 subjects participated in the preference test. However, this evaluation was conducted via Clickworker platform - a crowdsourced evaluation, unlike earlier evaluations. The results of the preference test are shown in Figure 5b. There was not a statistically significant preference for the Word-level MSS system (p-value=0.068). However, upon removing



(a) *speaker-A*

(b) *speaker-B*

Figure 5: *The results of preference tests between the Word-level MSS and the baseline system on two voices.*

listeners that had low reliability because they did not listen completely to both samples, we found that the difference becomes significant, i.e. Word-level MSS is preferred statistically significantly (p-value=0.007). The results of both MUSHRA and preference tests suggest that the Word-level MSS system is able to produce more natural speech than the baseline system. We observe that the Word-level MSS has more contextually appropriate emphasis on the words and generally better intonation without impacting the segmental quality. Although the Word-level MSS system is preferred over the baseline system, we do note that on certain utterances it is unable to produce the right kind of intonation, specifically when a question does not start with an interrogative word.

5. Conclusion

In this paper, we presented a novel method for multi-scale modelling of mel-spectrograms to improve the quality of NTTS systems. We presented a generic MSS modelling approach and later provided details for its two specific versions called Sentence-level MSS and Word-level MSS where the scales correspond to the linguistic units. The ablation study showed that the Word-level MSS system performed statistically significantly better than Sentence-level MSS. In the preference evaluations on 2 voices, the Word-level MSS system showed statistically significantly better results than the baseline system. In the future, we want to introduce scales in MSS along the frequency axis as well which could result in an even improved segmental quality. We also want to extend the sentence-level MSS to broader linguistic units for a better modelling of the coarse-grained prosody of speech. Furthermore, we want to introduce another scale that corresponds to syllables as they are strongly linked to prosodic events like stress and intonation [26].

6. References

[1] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[2] Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio *et al.*, "Tacotron: Towards end-to-end speech synthesis," *arXiv preprint arXiv:1703.10135*, 2017.

[3] H. Zen, K. Tokuda, and A. W. Black, "Statistical parametric speech synthesis," *Speech Communication*, vol. 51, no. 11, pp. 1039–1064, 2009.

[4] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan *et al.*, "Natural TTS synthesis by conditioning wavenet on mel spectrogram predictions," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4779–4783.

[5] Y. Ning, S. He, Z. Wu, C. Xing, and L.-J. Zhang, "A review of deep learning based speech synthesis," *Applied Sciences*, vol. 9, no. 19, p. 4050, 2019.

[6] A. Curtin and H. Ayaz, "Cognitive considerations in auditory user interfaces: Neuroergonomic evaluation of synthetic speech comprehension," in *International Conference on Engineering Psychology and Cognitive Ergonomics*. Springer, 2017, pp. 106–116.

[7] R. Skerry-Ryan, E. Battenberg, Y. Xiao, Y. Wang, D. Stanton, J. Shor, R. Weiss, R. Clark, and R. A. Saurous, "Towards end-to-end prosody transfer for expressive speech synthesis with tacotron," in *international conference on machine learning*. PMLR, 2018, pp. 4693–4702.

[8] S. Karlapati, A. Abbas, Z. Hodari, A. Moinet, A. Joly, P. Karanasou, and T. Drugman, "Prosodic Representation Learning and Contextual Sampling for Neural Text-to-Speech," *arXiv preprint arXiv:2011.02252*, 2020.

[9] Z. Hodari, A. Moinet, S. Karlapati, J. Lorenzo-Trueba, T. Merritt, A. Joly, A. Abbas, P. Karanasou, and T. Drugman, "CAMP: a Two-Stage Approach to Modelling Prosody in Context," *arXiv preprint arXiv:2011.01175*, 2020.

[10] G. Sun, Y. Zhang, R. J. Weiss, Y. Cao, H. Zen, A. Rosenberg, B. Ramabhadran, and Y. Wu, "Generating diverse and natural text-to-speech samples using a quantized fine-grained VAE and autoregressive prosody prior," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6699–6703.

[11] K. Akuzawa, Y. Iwasawa, and Y. Matsuo, "Expressive Speech Synthesis via Modeling Expressions with Variational Autoencoder," *Proc. Interspeech 2018*, pp. 3067–3071, 2018.

[12] W.-N. Hsu, Y. Zhang, R. J. Weiss, H. Zen, Y. Wu, Y. Wang, Y. Cao, Y. Jia, Z. Chen, J. Shen *et al.*, "Hierarchical generative modeling for controllable speech synthesis," *arXiv preprint arXiv:1810.07217*, 2018.

[13] T. Kenter, V. Wan, C.-A. Chan, R. Clark, and J. Vit, "CHiVE: Varying prosody in speech synthesis with a linguistically driven dynamic hierarchical conditional variational network," in *International Conference on Machine Learning*. PMLR, 2019, pp. 3331–3340.

[14] C.-M. Chien and H.-y. Lee, "Hierarchical Prosody Modeling for Non-Autoregressive Speech Synthesis," *arXiv preprint arXiv:2011.06465*, 2020.

[15] S. Tyagi, M. Nicolis, J. Rohnke, T. Drugman, and J. Lorenzo-Trueba, "Dynamic prosody generation for speech synthesis using linguistics-driven acoustic embedding selection," *arXiv preprint arXiv:1912.00955*, 2019.

[16] T. Raitio, R. Rasipuram, and D. Castellani, "Controllable neural text-to-speech synthesis using intuitive prosodic features," *arXiv preprint arXiv:2009.06775*, 2020.

[17] Y. Ren, C. Hu, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, "FastSpeech 2: Fast and High-Quality End-to-End Text to Speech," in *International Conference on Learning Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=piLPYqxtWuA

[18] T. Drugman, G. Huybrechts, V. Klimkov, and A. Moinet, "Traditional machine learning for pitch detection," *IEEE Signal Processing Letters*, vol. 25, no. 11, pp. 1745–1749, 2018.

[19] S. Vasquez and M. Lewis, "Melnet: A generative model for audio in the frequency domain," *arXiv preprint arXiv:1906.01083*, 2019.

[20] S. Ruder, "An overview of multi-task learning in deep neural networks," *arXiv preprint arXiv:1706.05098*, 2017.

[21] M. He, Y. Deng, and L. He, "Robust Sequence-to-Sequence Acoustic Modeling with Stepwise Monotonic Attention for Neural TTS," in *Proc. Interspeech 2019*, 2019, pp. 1293–1297. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2019-1972

[22] Z. Zeng, J. Wang, N. Cheng, T. Xia, and J. Xiao, "AlignTTS: Efficient feed-forward text-to-speech system without explicit alignment," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6714–6718.

[23] C. Yu, H. Lu, N. Hu, M. Yu, C. Weng, K. Xu, P. Liu, D. Tuo, S. Kang, G. Lei *et al.*, "Durian: Duration informed attention network for multimodal synthesis," *arXiv preprint arXiv:1909.01700*, 2019.

[24] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[25] B. Series, "Method for the subjective assessment of intermediate quality level of audio systems," *International Telecommunication Union Radiocommunication Assembly*, 2014.

[26] J. Itô, *Syllable theory in prosodic phonology*. Routledge, 2018, vol. 10.

11th ISCA Speech Synthesis Workshop (SSW 11)
26-28 August 2021, Budapest, Hungary

How do Voices from Past Speech Synthesis Challenges Compare Today?

Erica Cooper, Junichi Yamagishi

National Institute of Informatics, Japan
ecooper@nii.ac.jp, jyamagis@nii.ac.jp

Abstract

Shared challenges provide a venue for comparing systems trained on common data using a standardized evaluation, and they also provide an invaluable resource for researchers when the data and evaluation results are publicly released. The Blizzard Challenge and Voice Conversion Challenge are two such challenges for text-to-speech synthesis and for speaker conversion, respectively, and their publicly-available system samples and listening test results comprise a historical record of state-of-the-art synthesis methods over the years. In this paper, we revisit these past challenges and conduct a large-scale listening test with samples from many challenges combined. Our aims are to analyze and compare opinions of a large number of systems together, to determine whether and how opinions change over time, and to collect a large-scale dataset of a diverse variety of synthetic samples and their ratings for further research. We found strong correlations challenge by challenge at the system level between the original results and our new listening test. We also observed the importance of the choice of speaker on synthesis quality.

Index Terms: speech synthesis, mean opinion score, listening test, Blizzard Challenge, Voice Conversion Challenge

1. Introduction

Since 2005, the annual Blizzard Challenge (BC) has provided researchers with a venue to compare their methods using common datasets and standardized evaluations. Likewise, since 2016, the biennial Voice Conversion Challenge (VCC) has done the same for the task of speaker conversion. Since the inception of the Blizzard Challenge, speech synthesis technology has transformed immensely, progressing through a diverse range of methods from unit selection synthesis, hidden Markov model based synthesis, and hybrid models to present-day state-of-the-art approaches such as end-to-end neural network based synthesis. In recent years, speech synthesis technology has also reached an overall level of acceptability to the general public where it is now very commonly used in various everyday consumer technologies.

In addition to providing shared data and evaluations for researchers to compare their approaches, the Blizzard and Voice Conversion Challenges also make the synthesized samples and raw listening test results publicly available, which is an invaluable resource for studying different models and approaches over time. Nevertheless, it is well-known that results from different listening tests cannot be meaningfully compared to each other [1] because the setting and conditions of the tests are not identical – the set of systems is different, and in particular the differing best and worst systems each year provide listeners with a completely different context for their evaluations. For this reason, we have gathered samples from past Blizzard and Voice Conversion Challenges into one new large-scale listening test which enables us to compare many past text-to-speech and voice conversion systems together. This allows us to answer the following research questions:

- How reproducible are MOS test results?
- How do past listening test results compare to ratings gathered in the present day?
- Will results still correlate even though the listening test context has changed?
- Can we observe the improvement of speech synthesis technology over the years in this data?
- How does quality of text-to-speech synthesis and voice conversion compare?
- What is the effect of the target speaker data on perceived synthesis quality?

Furthermore, a dataset of many years of synthesized samples along with their ratings from a single listening test will be a useful resource for training automatic evaluation metrics such as MOSNet [2]. In this paper, we will describe the design of a large-scale listening test that aims to compare quality of a diverse range of synthesis methods from past years' Blizzard and Voice Conversion Challenges. We will then present what we learned from this test in terms of synthesized speech and listener preferences of natural speech. To the best of our knowledge, this is the first time that samples from different years' challenges, as well as a combination of both text-to-speech synthesis and voice conversion samples, have been compared in one listening test together.

2. Related Work

In a 2014 overview of a decade of past Blizzard challenges [3], it was observed that unit selection based systems consistently had the best naturalness ratings over the years, whereas statistical parametric methods such as hidden Markov model based synthesis produced the most intelligible speech. Hybrid systems were beginning to show signs of incorporating the best of both worlds. While it was noted that "naturalness" as a basis for rating speech audio is inherently poorly-defined, the consistency of listener judgments shows that listeners are nevertheless able to understand and complete the task. In another meta-study [4], nine different past studies of human ratings of synthetic speech revealed five common aspects (naturalness, prosodic quality, intelligibility, disturbances, and calmness) that were consistently salient to listeners' judgements. There have also been a number of studies that re-visit or reproduce listening tests in order to study the reliability of MOS tests. For instance, [5] ran the same listening test both in lab and as an online crowdsourced task and found strong correlations between ratings in both settings, and furthermore ran the crowdsourced test five times on five different days with five different sets of listeners and also found good reliability between the sets of results. [6] also found good agreement and strong correlations between an in-lab listening

10.21437/SSW.2021-32

test and a crowdsourced one. In a 2015 re-visitation of the 2013 Blizzard Challenge results, [7] studied the stability of the significant differences between systems, finding that the results stabilize and have good reliability and discriminative power when at least 30 different listeners are included in the test.

Due to the expense and time-consuming nature of conducting subjective listening tests, there has long been interest in the development and use of objective measures for evaluating synthesis quality, and in particular, with the recent advances in neural network based modeling approaches, past listening test results can be used to train models for this purpose. For example, MOSNet [2] trained an end-to-end model for naturalness assessment on the VCC 2018 listening test results to predict human ratings of voice-converted speech. They further extended their model to predict speaker similarity in addition to MOS. While they found high correlations at the system level but only fair correlations at the utterance level due to large variances between listeners, [8] extended MOSNet to learn from this listener variation by incorporating a listener bias network that takes the listener label into consideration. In addition to improving utterance-level correlations when the appropriate listener label is given, overall system-level correlations were also improved. Another extension of MOSNet was conducted in [9], in which the models trained for VC were found not to generalize well to TTS, so MOSNet models were trained on the ASVspoof 2019 Logical Access dataset [10], which contains synthesized speech from 13 different speech synthesizers and voice conversion systems trained on the same set of speakers. Eight different feature representations were studied to determine which one is best for this type of evaluation task. While [11] cautions that even an objective measure depends on its context (i.e., its training data) much in the same way that human listening tests do, it is our hope that very large-scale listening test data such as that collected in our study will provide sufficient context to train objective measures that have good generalization capability in the future.

3. Listening Test Design

We gathered samples and ratings from past Blizzard and Voice Conversion Challenges[1]. We focused on English-language synthesis and the main Hub tasks for each year. The Blizzard Challenge years that we included were 2008 [12], 2009 [13], 2010 [14], 2011 [15], 2013 [16], and 2016 [17], as well as all Voice Conversion Challenge years (2016 [18, 19], 2018 [20], and 2020 [21, 22]). We also included samples from a number of systems from ESPnet [23], which is a popular open-source toolkit for end-to-end speech and language technologies, since samples for a number of implemented text-to-speech architectures have been released along with their listening test results [24]. Our total number of systems, including natural speech, was 187.

We chose 38 samples for each of the 187 systems, balancing where relevant over genre (e.g. news, audiobook, conversational). We excluded semantically-unpredictable sentences, which were used in past Blizzard challenges mainly for intelligibility evaluation, as well as any other genres which were not included in the original naturalness evaluations, and genres for which there were no corresponding natural speech samples. For voice conversion systems, we balanced over all combinations of source and target speakers. Even though VCC 2020 had both intra-lingual and cross-lingual tasks, we only included samples from the intra-lingual task. Some challenges did not have 38 unique test utterances, so in those cases we included repeat samples. To avoid differing sampling rates as a confounding factor, we downsampled all audio to 16kHz, and conducted amplitude normalization using sv56 [25].

Each listening test set consisted of one sample from each of the 187 different systems. Listeners could listen to each sample as many times as they liked, but were required to play the entire sample at least once and choose a rating for it before proceeding to the next one. Listeners were asked to rate each sample on a 5-grade Mean Opinion Score (MOS) scale from 1 (very bad) to 5 (very good). In order to get ratings from as many different listeners as possible, each listener was only permitted to evaluate one set. Each set was rated by eight different listeners, and overall, 304 different listeners participated in our test. Due to the constraints of our location, we recruited Japanese native listeners to participate in our test, but we also note the very strong correlations with native English listeners reported in [21]. Listener gender demographics were 141 male, 159 female, and 4 other. Listener age demographics were 48 listeners between 18 and 29 years old, 118 listeners in their 30s, 90 listeners in their 40s, 35 listeners in their 50s, 12 listeners in their 60s, and one listener age 70 or older. We measured significant differences between systems using the Mann-Whitney U test, following [26], at a level of $p<0.05$, with Bonferroni correction for multiple comparisons.

4. Results and Analysis

A histogram of the ratings for all 187 systems, arranged from lowest to highest MOS, can be seen in Figure 1. We found moderate listener agreement, with both Krippendorff's alpha and intra-class correlation equal to 0.50.

Looking at the standard deviations of each system, we noticed that some systems were less agreed-upon than others. ESPnet-Merlin, a DNN-based parametric model trained using the Merlin toolkit [27], had the highest standard deviation, with an almost equal number of 5 and 1 ratings. The systems with the lowest standard deviations tended to be natural speech (very highly rated) or the lowest-rated systems. Violin plots of the rating distributions of the most- and least-agreed-upon systems are in Figure 2.

4.1. Best and Worst Systems

Systems are named according to the challenge that they came from, followed by the team letter name or other system identifier. The five best synthesized systems, which were not significantly different from one another, are the following:

- ESPnet-transformerv3
- BC2010-M
- ESPnet-transformerv1
- ESPnet-tacotron2v3
- ESPnet-nvidia

It is notable that four out of the five best-rated systems are from ESPnet. One drawback of our listening test as compared to the standard Blizzard evaluations is that we are mixing systems that were trained on a variety of different databases, so it becomes more difficult to determine whether a model is inherently better or if listeners simply prefer the sound of the voice data on which it was trained. We will discuss this more in Section 5.

[1] https://www.cstr.ed.ac.uk/projects/blizzard/data.html

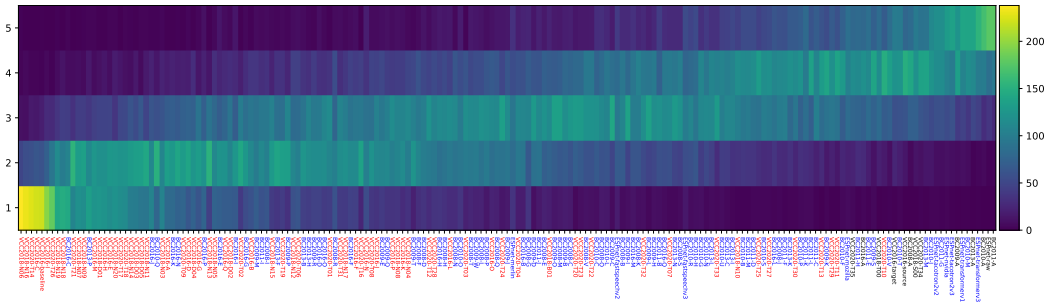


Figure 1: *Histogram of MOS ratings for 187 systems. Natural speech system names are indicated in black text, TTS systems are blue, and voice conversion systems are red.*

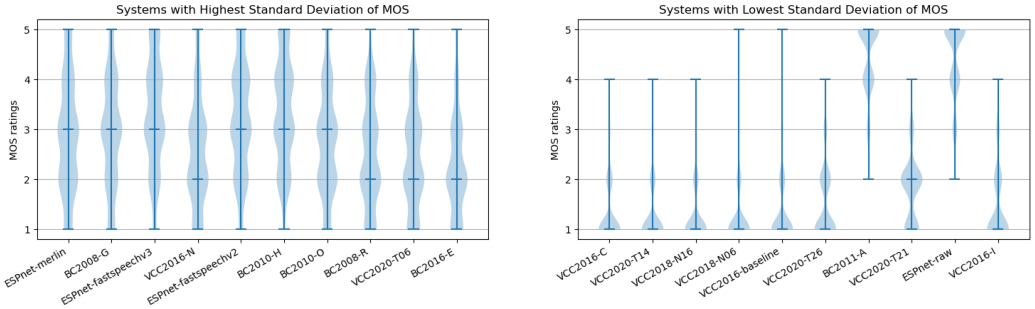


Figure 2: *Violin plots of the systems with the highest and lowest standard deviations.*

The group of worst systems which are not significantly different from one another are as follows:

- VCC2018-N06
- VCC2018-N16
- VCC2020-T14
- VCC2016-C
- VCC2016-baseline

Text-to-speech and voice conversion systems are rarely compared together in the same listening test, but this large-scale test gave us the opportunity to do so. It is notable that the worst-rated systems are all voice conversion ones. Is the state of the art of text-to-speech synthesis better overall (in terms of naturalness) than that of voice conversion? One consideration is that voice conversion from a source speaker to a target speaker of a different gender may produce worse speech signal quality than the same-gender condition, since the distance between source and target speaker is farther. So, we tried excluding samples where the source and target speakers were different genders and re-computed MOS. Although the ordering changes slightly, we find that the worst systems are still voice conversion ones. Furthermore, although the MOS values tend to improve slightly by only considering same-gender conversion, we find that it is generally not statistically significant – only four out of the 73 voice conversion systems show any significant improvement. Since we have both TTS and VC systems from 2016, we can also compare the best systems from both challenges in the same year: BC2016-L was rated as significantly better than VCC2016-O.

Another consideration is that the Voice Conversion Challenges provide teams with much less data per speaker, often as few as around 80 utterances, whereas Blizzard Challenge data is typically on the order of a few hours or thousands of utterances. These kinds of low-resource data conditions make it more challenging to achieve a high level of naturalness, which is apparent from the listener ratings.

4.2. Correlations with past challenge results

At the system level, challenge by challenge, we found very strong correlations, using both the Pearson correlation coefficient (PCC) and the Spearman rank correlation coefficient (SRCC), between the original listening test results and the new ones. We report these values and also root mean squared error (RMSE) in Table 1. At the utterance level, we find lower but still moderately positive correlations. Individual utterance-level scores were not available for BC2013 and BC2016.

The large RMSE values show the effects of context – even though year-by-year correlations are strong, the overall values of the ratings themselves compared to the original ones do vary.

4.3. Improvements of speech synthesis over time

Year by year, is the best system in each challenge better than the previous year's best system? At what point in time did synthesized speech reach the quality where it was not rated as significantly different from natural speech? Table 2 shows the MOS of the best system for each challenge, whether its MOS has

Table 1: *System-level and utterance-level PCC, SRCC, and RMSE between original and new listening test results by challenge or set of systems*

	System-level			Utterance-level		
Challenge	**PCC**	**SRCC**	**RMSE**	**PCC**	**SRCC**	**RMSE**
BC2008	0.93	0.89	0.33	0.70	0.67	0.62
BC2009	0.97	0.95	0.48	0.76	0.72	0.64
BC2010	0.93	0.98	0.66	0.74	0.73	0.85
BC2011	0.91	0.90	0.76	0.76	0.67	0.87
BC2013	0.97	0.98	0.49	-	-	-
BC2016	0.97	0.93	0.40	-	-	-
VCC2016	0.97	0.92	0.42	0.56	0.53	1.12
VCC2018	0.96	0.91	0.77	0.55	0.53	1.10
VCC2020	0.98	0.96	0.23	0.87	0.87	0.48
ESPnet	0.99	0.98	0.09	0.73	0.61	0.59

Table 2: *Best system in each challenge compared to the previous challenge's best system and to natural speech – whether MOS has improved since the last challenge (Impr.?), whether the difference is significant (Sig.?), and whether the difference in MOS to that year's natural speech is significant (Sig. (Nat)).*

Year : Best system	**MOS**	**Impr.?**	**Sig.?**	**Sig. (Nat)**
BC2008 : J	3.63			✓
BC2009 : S	3.87	✓	x	✓
BC2010 : M	4.27	✓	✓	x
BC2011 : G	4.12	x	x	✓
BC2013 : M	4.01	x	x	x
BC2016 : L	3.63	x	✓	x
VCC2016 : O	2.86			✓
VCC2018 : N10	3.55	✓	✓	x
VCC2020 : T10	3.88	✓	x	x
ESPnet : transformerv3	4.33			x

improved over the previous challenge's best system, whether this difference is significant, and whether this challenge's best system is significantly different from that same year's natural speech. We can observe that while VCC best systems do improve challenge by challenge, the best Blizzard Challenge system from 2016 was rated as significantly worse than the best system from BC2013. This is likely due to the effects of the different training corpora. We can also observe that in 2010 and onwards (excepting 2011) for TTS, and from 2018 onwards for voice conversion, the best systems' MOS ratings were not significantly different from natural speech.

Since some listeners may have strong preferences about the speaker voice chosen for a given year's challenge, and since these preferences will therefore skew that listener's ratings for all systems trained on that dataset, adjusting for these preferences may allow us to see more clearly an overall trend of how TTS systems perform relative to the quality of natural speech over time. Z-score normalization was conducted based on statistics of all of a listener's ratings for systems in a single challenge, normalized average scores were computed for each system from the normalized individual ratings, and differences were computed between the normalized score of a given year's natural speech and of each system from that year. Results are plotted in Figure 3. We can see that the gap between natural speech and the best system becomes smaller year by year (with the exception of a very good best system in 2010), and also that a larger number of systems tend to approach natural speech over time.

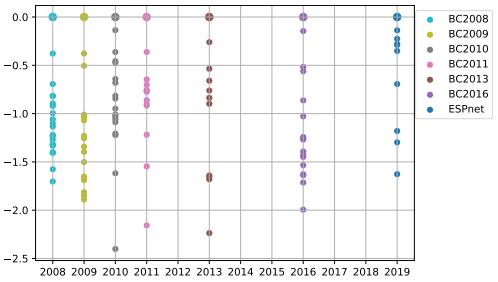


Figure 3: *Difference of each system from natural speech, computed from averaged z-score-normalized ratings by listener for each challenge. The top row of larger dots are natural speech.*

4.4. Objective measures

We objectively measured all of the samples included in our listening test using a number of metrics: word error rate (WER) using the IBM Watson speech-to-text API[2], signal-to-noise ratio (SNR) using the WADA SNR algorithm [28], the ITU-T P.563 method for objective speech quality assessment [29], and a pretrained MOSNet model from [9]. Although there are many pretrained MOSNet models to choose from, we chose this one because the fact that it was trained on a variety of TTS and VC systems from ASVspoof makes it a good match for our domain, and furthermore, other models which were trained on VCC data would not be valid to use since we would be testing on those models' training data. Surprisingly we found that WER had the strongest (negative, as expected) Pearson's correlation with MOS at r=-0.52. SNR had a weak negative corrlation of r=-0.17. The p563 measure had a very weak correlation of r=0.05, and surprisingly, MOSNet had the weakest correlation of all at r=0.03. There is clearly room for improvement in terms of generalizable objective measures for synthesized speech.

5. Natural speech preferences and effects of the corpus on TTS

In [30], the voice of the speech corpus was found to have a significant effect on the ratings of the synthesized speech. They caution that the selection of the speaker for the training corpus is crucial due to the large effects that the speaker can have on the perceived quality of the synthesized speech. [9] similarly found that the speaker has a large effect on synthesis quality, and that systems trained on data from certain speakers reached a consistent quality, regardless of the type of synthesis model used. We observe this in our listening test data as well. This is a confounding factor that prohibits meaningful direct comparisons between systems from different challenges; however, for training a system such as MOSNet, it is important to be able to replicate these human preferences even if they are simply based on characteristics of the speaker data.

5.1. Metadata

We have useful metadata about various speaker characteristics, such as gender, dialect (American vs. British), and whether or not the speaker is a professional voice talent (speakers who were not specifically stated to be professional speakers in the data

[2]https://www.ibm.com/cloud/watson-speech-to-text

descriptions were assumed not to be). We found a significant preference for professional speech over non-professional speakers, a marginally significant preference for female speakers over male speakers at p=0.05, and no significant preference between British and American speakers. The preference for professional speech may account for some of the difference between voice conversion and text-to-speech systems, since the voice conversion challenges rely on non-professional speech.

According to [31], listeners tend to rate spontaneous speech as more natural, even if not explicitly instructed to pay attention to style. So, we consider whether the genre or style of the natural speech has an effect on perceived naturalness. The three main genres that we included from the Blizzard Challenges are news, book sentences, and a "conversational" genre which is not spontaneous conversational speech, but rather meant to be speech from a virtual conversational agent whose purpose is to help the user search for restaurants and navigate the results. We find that news sentences are overall rated the most natural with a MOS of 4.36 and the conversational genre had a MOS of 4.14. The book sentences were rated as significantly less natural than the news sentences, with a MOS of 4.09. It is surprising that the book speech was rated as less natural, but the highly expressive style of many of the book sentences may come across as unnatural out of context.

One interesting observation we made during these analyses is that although the speech data came from the same speaker in both Blizzards 2008 and 2009, there was a significant preference for the Blizzard 2009 natural speech. In fact, even controlling for genre by considering only news utterances, we still found a significant difference. Listening to samples from these sets, we observed that the audio quality was much better for the 2009 samples. From this we can conclude that listeners are able to consistently pick up on such differences in recording quality.

5.2. Speaker characteristics

We next consider whether there are certain speaker characteristics that listeners tend to favor when rating naturalness. For each speaker, using Praat [32], we measure the minimum, maximum, mean, and standard deviation of f0 and energy, as well as noise-to-harmonic ratio (NHR), jitter, and shimmer. We found moderate negative correlations with MOS for shimmer (r=-0.46), NHR (r=-0.41), and mean energy (r=-0.37), and a moderate positive correlation for standard deviation of energy (r=0.41). A study of vocal attractiveness [33] also found that harmonic-to-noise ratio (the inverse of NHR) was significantly correlated to ratings of vocal attractiveness, suggesting that perceptions of naturalness and vocal attractiveness may be related. Furthermore, [34] observed that selecting speakers with low mean energy for training statistical parametric speech synthesis models resulted in more intelligible synthetic speech, which we have also observed correlates with better naturalness ratings.

5.3. Effect of corpus on benchmark systems

Every Blizzard evaluation contains samples from two benchmark systems: Festival [35] and HTS [36]. The Festival benchmark system is the same every year, and the HTS benchmark is that year's HTS version. Festival and HTS are denoted as systems B and C respectively in each Blizzard challenge. The inclusion of these benchmarks allows us to study the effect of the speaker data on TTS systems that are mostly consistent. Comparing Pearson and Spearman corrrelations between the scores for these benchmark systems each year and the corresponding natural audio, we find moderate correlations for Festival (Pearson r=0.33, Spearman r=0.54) and strong correlations for HTS (Pearson r=0.87, Spearman r=0.90). This indicates that while both systems' synthesis output reflects preferences about the chosen speaker used for training, HTS is more sensitive to the choice of data. Groupings of the MOS of benchmark systems with their respective natural speech can be seen in Figure 4.

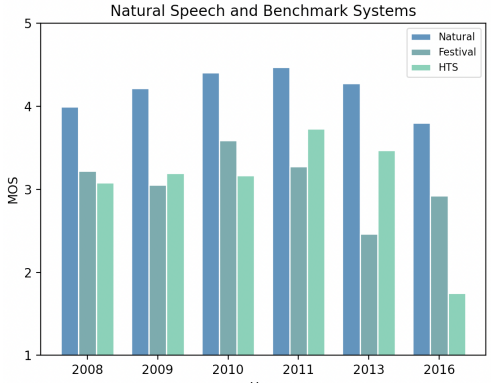


Figure 4: *MOS for natural speech and benchmarks for each Blizzard year*

6. Discussion and Future Work

In a large-scale listening test combining samples from various Blizzard Challenges, Voice Conversion Challenges, and ESPnet models, we showed the reliability of MOS tests through their strong correlations with MOS results from past tests. In doing so, we also produced a very large dataset of synthesized samples from 187 different systems, each with eight human ratings for naturalness, and all in the same listening test context, with both text-to-speech and voice conversion systems rated together, which can be used for further analysis and for training MOSNet-type systems for automatic objective evaluation. We also observed the importance of the choice of speaker for the training data on synthesis quality, and identified some speaker characteristics for which listeners had preferences. By adjusting for individual listener preferences and measuring distance to natural speech, we can observe the trend of improvement in TTS over time as more systems approach the quality of natural speech.

We have observed that some systems have clear agreement, whereas others, such as ESPnet-Merlin, have a wider distribution of scores. For these such less-agreed-upon systems, it would be interesting to know the source of these disagreements and what makes them so "controversial" – e.g., if certain types of artifacts or unnaturalness are very salient to some listeners but not others, or if the variation comes from large differences in quality of synthesis by utterance.

In future work, we will conduct a similar listening test with native English listeners, and also collect ratings for speaker similarity. These large datasets will allow us to to train or fine-tune MOSNet models for this test context. Having similar listening test data for both English and Japanese listeners will also enable us to study cross-cultural aspects of preferences for speaker characteristics and speaking styles.

7. Acknowledgements

We would like to thank the organizers of the Blizzard Challenges and Voice Conversion Challenges for making the samples and listening test data freely available. We would also like to thank Simon King and Alan Black for answering our questions about the data, Tomoki Hayashi for kindly providing utterance-level MOS ratings for the ESPnet systems, and Sébastien Le Maguer for helpful discussions. This study is supported by JST CREST Grants (JPMJCR18A6 and JPMJCR20D3) and MEXT KAKENHI Grants (21K11951), Japan.

8. References

[1] T. Hoßfeld, P. E. Heegaard, M. Varela, and S. Möller, "QoE beyond the MOS: an in-depth look at QoE via better metrics and their relation to MOS," *Quality and User Experience*, vol. 1, no. 1, pp. 1–23, 2016.

[2] C.-C. Lo, S.-W. Fu, W.-C. Huang, X. Wang, J. Yamagishi, Y. Tsao, and H.-M. Wang, "MOSNet: Deep Learning-Based Objective Assessment for Voice Conversion," in *Proc. Interspeech 2019*, 2019, pp. 1541–1545. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2019-2003

[3] S. King, "Measuring a decade of progress in text-to-speech," *Loquens*, vol. 1, no. 1, pp. e006–e006, 2014.

[4] F. Hinterleitner, C. Norrenbrock, and S. Möller, "Is intelligibility still the main problem? a review of perceptual quality dimensions of synthetic speech," in *Eighth ISCA Workshop on Speech Synthesis*, 2013.

[5] B. Naderi and R. Cutler, "An open source implementation of ITU-T Recommendation P. 808 with validation," 2020.

[6] R. Zequeira Jiménez, A. Llagostera, B. Naderi, S. Möller, and J. Berger, "Intra-and inter-rater agreement in a subjective speech quality assessment task in crowdsourcing," in *Companion Proceedings of The 2019 World Wide Web Conference*, 2019, pp. 1138–1143.

[7] M. Wester, C. Valentini-Botinhao, and G. E. Henter, "Are we using enough listeners? no!—an empirically-supported critique of Interspeech 2014 TTS evaluations," in *Interspeech*, 2015.

[8] Y. Leng, X. Tan, S. Zhao, F. Soong, X.-Y. Li, and T. Qin, "MBNet: MOS prediction for synthesized speech with mean-bias network," *arXiv preprint arXiv:2103.00110*, 2021.

[9] J. Williams, J. Rownicka, P. Oplustil, and S. King, "Comparison of speech representations for automatic quality estimation in multi-speaker text-to-speech synthesis," 2020.

[10] M. Todisco, X. Wang, V. Vestman, M. Sahidullah, H. Delgado, A. Nautsch, J. Yamagishi, N. Evans, T. H. Kinnunen, and K. A. Lee, "ASVspoof 2019: future horizons in spoofed and fake audio detection," in *Proc. Interspeech*, 2019, pp. 1008–1012.

[11] "ITU-T Recommendation P.800.2, mean opinion score interpretation and reporting," 2016.

[12] V. Karaiskos, S. King, R. A. Clark, and C. Mayo, "The Blizzard Challenge 2008," in *Proc. Blizzard Challenge Workshop*. Citeseer, 2008.

[13] A. W. Black, S. King, and K. Tokuda, "The Blizzard Challenge 2009," in *Proc. Blizzard Challenge*, 2009, pp. 1–24.

[14] S. King and V. Karaiskos, "The Blizzard Challenge 2010," 2010.

[15] ——, "The Blizzard Challenge 2011," 2011.

[16] ——, "The Blizzard Challenge 2013," 2013.

[17] ——, "The Blizzard Challenge 2016," 2016.

[18] T. Toda, L.-H. Chen, D. Saito, F. Villavicencio, M. Wester, Z. Wu, and J. Yamagishi, "The Voice Conversion Challenge 2016," in *Interspeech*, 2016.

[19] M. Wester, Z. Wu, and J. Yamagishi, "Analysis of the Voice Conversion Challenge 2016 evaluation results." in *Interspeech*, 2016, pp. 1637–1641.

[20] J. Lorenzo-Trueba, J. Yamagishi, T. Toda, D. Saito, F. Villavicencio, T. Kinnunen, and Z. Ling, "The Voice Conversion Challenge 2018: Promoting development of parallel and nonparallel methods."

[21] Z. Yi, W.-C. Huang, X. Tian, J. Yamagishi, R. K. Das, T. Kinnunen, Z. Ling, and T. Toda, "Voice Conversion Challenge 2020 — intra-lingual semi-parallel and cross-lingual voice conversion —," in *Proc. Joint Workshop for the Blizzard Challenge and Voice Conversion Challenge 2020*, 2020, pp. 80–98.

[22] R. K. Das, T. Kinnunen, W.-C. Huang, Z. Ling, J. Yamagishi, Y. Zhao, X. Tian, and T. Toda, "Predictions of subjective ratings and spoofing assessments of Voice Conversion Challenge 2020 submissions," in *Proc. Joint Workshop for the Blizzard Challenge and Voice Conversion Challenge 2020*, 2020, pp. 99–120. [Online]. Available: http://dx.doi.org/10.21437/VCC_BC.2020-15

[23] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. Enrique Yalta Soplin, J. Heymann, M. Wiesner, N. Chen, A. Renduchintala, and T. Ochiai, "ESPnet: End-to-end speech processing toolkit," in *Proceedings of Interspeech*, 2018, pp. 2207–2211. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2018-1456

[24] T. Hayashi, R. Yamamoto, K. Inoue, T. Yoshimura, S. Watanabe, T. Toda, K. Takeda, Y. Zhang, and X. Tan, "ESPnet-TTS: Unified, reproducible, and integratable open source end-to-end text-to-speech toolkit," 2020.

[25] International Telecommunication Union, Recommendation G.191: Software Tools and Audio Coding Standardization, Nov 11 2005.

[26] A. Rosenberg and B. Ramabhadran, "Bias and statistical significance in evaluating speech synthesis with mean opinion scores," in *Proc. Interspeech 2017*, 2017, pp. 3976–3980. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2017-479

[27] Z. Wu, O. Watts, and S. King, "Merlin: An open source neural network speech synthesis system," *9th ISCA Speech Synthesis Workshop*, 2016.

[28] C. Kim and R. M. Stern, "Robust signal-to-noise ratio estimation based on waveform amplitude distribution analysis," in *Ninth Annual Conference of the International Speech Communication Association*, 2008.

[29] L. Malfait, J. Berger, and M. Kastner, "P.563—The ITU-T standard for single-ended speech quality assessment," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 6, pp. 1924–1934, 2006.

[30] F. Hinterleitner, C. Manolaina, and S. Möller, "Influence of a voice on the quality of synthesized speech," in *2014 Sixth International Workshop on Quality of Multimedia Experience (QoMEX)*. IEEE, 2014, pp. 99–104.

[31] R. Dall, J. Yamagishi, and S. King, "Rating naturalness in speech synthesis: The effect of style and expectation," in *Proceedings of Speech Prosody*, 2014.

[32] P. Boersma, "Praat, a system for doing phonetics by computer," *Glot International*, vol. 5, no. 9–10, pp. 341–345, 2001.

[33] L. Bruckert, P. Bestelmeyer, M. Latinus, J. Rouger, I. Charest, G. A. Rousselet, H. Kawahara, and P. Belin, "Vocal attractiveness increases by averaging," *Current biology*, vol. 20, no. 2, pp. 116–120, 2010.

[34] K.-Z. Lee and E. Cooper, "A comparison of speaker-based and utterance-based data selection for text-to-speech synthesis," *Interspeech 2018*, vol. 12873, 2018.

[35] A. Black, P. Taylor, and R. Caley, "The Festival speech synthesis system, system documentation, edition 2.4, for Festival version 2.4.0," http://www.festvox.org/docs/manual-2.4.0/festival_toc.html, 2014.

[36] H. Zen, T. Nose, J. Yamagishi, S. Sako, T. Masuko, A. W. Black, and K. Tokuda, "The HMM-based speech synthesis system (HTS) version 2.0," *6th ISCA Workshop on Speech Synthesis*, 2007.

11th ISCA Speech Synthesis Workshop (SSW 11)
26-28 August 2021, Budapest, Hungary

Accent Modeling of Low-Resourced Dialect in Pitch Accent Language Using Variational Autoencoder

Kazuya Yufune[1], *Tomoki Koriyama*[1], *Shinnosuke Takamichi*[1], *Hiroshi Saruwatari*[1]

[1]Graduate School of Information Science and Technology, The University of Tokyo, Japan.

kazuya_yufune@ipc.i.u-tokyo.ac.jp, t.koriyama@ieee.org

Abstract

Realizing text-to-speech (TTS) system of dialects is useful for personalizing TTS systems. However, TTS for many dialects of pitch accent languages is not realized because of low-resourced problem. Among many dialects of pitch accent languages, this paper focuses on Osaka dialect of Japanese, one of the most challenging pitch accent languages. For Japanese TTS system, accent labels are known to be necessary as input to synthesize natural speech. In rich-resourced dialect, human-resourced approaches and dictionary-based approaches are often used to annotate accent labels for training and inference, but such approaches are unfeasible and time-consuming for low-resourced dialects. In this paper, we propose accent extraction model that utilizes vector quantized variational autoencoder (VQ-VAE) to prepare accent information from speech, and accent prediction models that utilize decision tree and deep learning techniques to predict accent information from the input text. The models were examined with corpus of Osaka dialect, whose accent labels do not exist. The results showed that accent extraction model succeeded in extracting accent information of Osaka dialect from speech utterances as latent variable. It also showed that the accent of synthesized speech by accent prediction models were not better than baseline, but it had advantages such as interpretability.

Index Terms: pitch accent, speech synthesis, Japanese dialect, VQ-VAE, accent label, latent variable

1. Introduction

Text-to-speech (TTS) systems with dialects makes speech applications diverse. For example, personalizing TTS with the speaker's dialect can be an alternative form of voice output for patients who have progressive dysarthria and want to speak in their dialects [1]. For another example, dialect TTS systems could be adopted for local characters to speak in the local dialects.

For pitch accent languages such as Japanese, it is known that accent information of input texts has an important role for TTS to synthesize natural-sounding speech [2, 3]. For example, in Japanese, a change in pitch makes a difference between words. Changing the pitch of "chopsticks" (/ha'shi/) differentiates the meaning into "bridge" (/hashi'/) or "edge" (/hashi'/). Though these words have the same phonemes /hashi/, Japanese speakers distinguish them by the pitch accent. In Japanese TTS system, without inputting the accent information as accent labels, an acoustic model cannot capture the pitch fluctuations appropriately, resulting in unnatural (sometimes even wrong) synthetic speech. Hence, accent labels need to be correctly given from text in pre-processing for Japanese TTS systems. For TTS of the Tokyo dialect, accent labels are annotated typically by professional annotators or dictionary-based approaches such as OpenJTalk [4]. Since the Tokyo dialect is the standard dialect of Japanese, TTS of this dialect can utilize rich resources such as professional annotators and an accent dictionary.

However, TTS systems for many dialects of pitch accent languages have been suffering from low resource problems. Specifically, recorded speech data set is not sufficient for modeling of fundamental frequency (F0) curves of accents even if we use an end-to-end TTS framework [2]. Although it is true that accent labels improve the synthetic F0 curves, annotating pitch accent labels requires professional annotators familiar with both the target dialect and the pitch accent system. Moreover, the accents of dialects are rarely summarized as an accent dictionary. Therefore, we should investigate the dialect TTS system under the condition that accent labels are not sufficiently provided.

In this paper, we focus on the Osaka dialect, which is among the dialects of Japanese and significantly different from the Tokyo dialect. To overcome the low resource problems, we propose two frameworks: accent extraction models for accent modeling in training, and accent prediction models for accent modeling in inference. The accent extraction models are used to extract latent representations of accent from speech. As the accent extraction models, we use not only variational autoencoder (VAE) [5], which was successful in extracting sentence-level prosody representations [6], but also vector quantized VAE (VQ-VAE) [7] to express discrete characteristics of Japanese accent. Mora-level latent variable representation of accent using VAE and VQ-VAE enables an acoustic model to be trained without annotated accent labels. The accent prediction models are used to infer the latent variable representations. We examine the effectiveness of two accent prediction models using recurrent neural networks (RNNs) and decision trees. We also investigate the use of the accent dictionary of the Tokyo dialect as the input of the accent prediction models.

2. Japanese pitch accent of Tokyo and Osaka dialects

The label of the accent system of Japanese is defined as high or low for each mora, which fundamentally corresponds to a Japanese Hiragana/Katakana character [8]. In the Tokyo dialect, Japanese words have an accent nucleus position, where the label changes from high to low. In the case of two-mora nouns, the nucleus position is among "no-nucleus (0)," "1," or "2." An example of accent labels of two-mora nouns for the Tokyo dialect is shown in Figure 1. The last mora "wa" is a postpositional particle in Japanese. In this example,"ha-shi (edge)" has no accent nucleus, "ha-shi (chopsticks)" has nucleus position of "1," and "ha-shi (bridge)" has that of "2." Since these words have different accents, their corresponding accent labels are different. This accent information is in the accent dictionary for the Tokyo dialect.

10.21437/SSW.2021-33

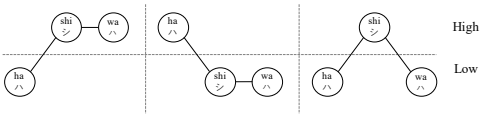


Figure 1: *Example of accent labels. left: "ha-shi (edge)"+"wa", center: "ha-shi (chopsticks)"+"wa", right: "ha-shi (ridge)"+"wa".*

Table 1: *Corresponding relationships of accent labels of two-mora nouns + postpositional particle "wa" between the Tokyo and Osaka dialects (H: High, L: Low)*

Tokyo dialect	Osaka dialect
L - H - H (no-nucleus)	H - H - H
H - L - L (nucleus position 1)	L - H - L
L - H - L (nucleus position 2)	H - L - L

The Osaka dialect is spoken in around Osaka prefecture[1]. When constructing a TTS system for this dialect, accent labels of the dialect are needed as the input, but the accent dictionary of the Osaka dialect does not exist. One of the available resources related to Japanese pitch accent is the accent dictionary of the Tokyo dialect. However, since the Osaka dialect has an accent system which is completely different from that of the Tokyo dialect, the accent dictionary of the Tokyo dialect is not suitable as it is for estimating the accent of the Osaka dialect. On the other hand, there are some corresponding relationships between the Tokyo and Osaka dialects [9]. For example, it is known that the accents of two-mora nouns of the dialects correspond to each other as Table 1 shows.

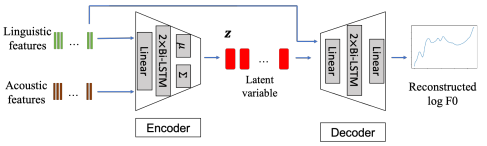
3. Related work

There have been many studies that focused on representation learning of prosody from acoustic features, including [10, 6, 11, 12]. Zhao et al. [11] proposed a model that reconstructs speech waveform with VQ-VAE [7] and down-sampled frame-level F0-related latent representation extracted from F0 curve. Hodari et al. [12] succeeded in improving prosody of synthesized speech by learning word-level prosody representations from referenced mel-spectrogram using VQ-VAE, and predicting them from the context in inference. Kenter et al. [6] proposed a hierarchical VAE [5] model that can synthesize a variety of prosodic features such as F0 by using sentence-level prosody embeddings. In this study, we examine VAE and VQ-VAE models for accent modeling of the Osaka dialect, with mora-level latent representation learning of pitch accent.

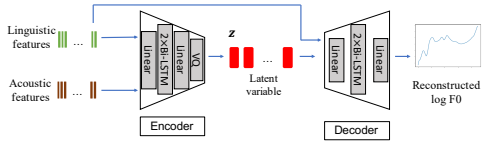
4. Latent-variable-based accent extraction models for Japanese using VAE models

We propose latent-variable-based accent extraction models for the expression of pitch accent of Osaka dialect. Specifically, we utilize VAE and VQ-VAE, which have been successful in prosody modeling described in the previous section. We adopted mora-level latent representation for accent modeling, as Japanese pitch accent of all dialects is defined for each mora as described in Sec. 1. We assume that we only have texts and speech utterances, and that accent labels are unavailable, which often happens, especially when targeting at low-resourced dialects.

[1]The second-largest metropolitan area in Japan.



(a) *VAE structure. Linear means linear layer, bi-LSTM means bi-directional long short term memory (LSTM) cells layer*



(b) *VQ-VAE structure. VQ means vector quantization layer which quantized the output of the previous linear layer.*

Figure 2: *Structure of accent extraction models*

4.1. Structure of accent extraction models

We propose accent extraction models that use VAE and VQ-VAE to extract the accent information from speech samples as latent variables. The model structures are shown in Figure 2. First, the encoder takes time-series frame-level linguistic $\boldsymbol{x}$ and acoustic features $\boldsymbol{y}$ as the input, and outputs latent variables $\boldsymbol{z}$ for each mora. In the second bi-LSTM layer of the encoder, the output of the last frame of each mora is propagated to the next layer, which results in transforming the frame-level features into mora-level features. The decoder takes frame-level linguistic features $\boldsymbol{x}$ and the mora-level latent variables $\boldsymbol{z}$ as the input, and predicts F0 curve for the speech $\hat{\boldsymbol{y}}_{F0}$. By providing linguistic features $\boldsymbol{x}$ that have no accent information of either the Tokyo nor Osaka dialect, we expect that the latent variables represent the accent information extracted from the acoustic features.

4.2. VAE model

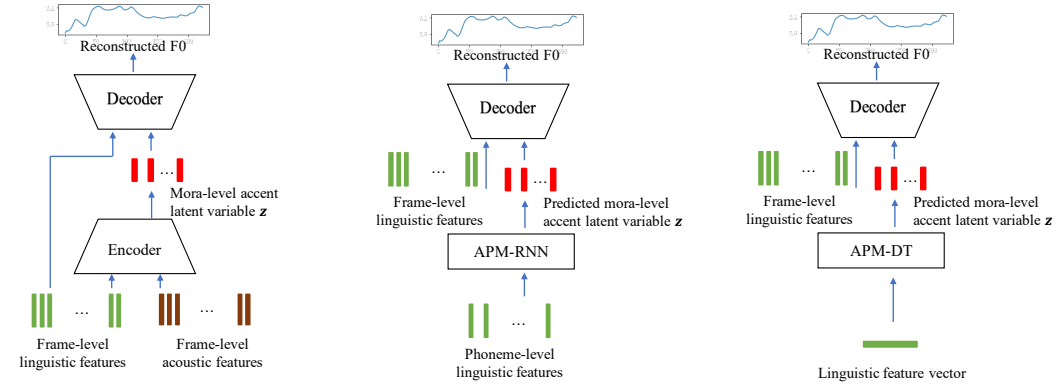
In this section, we propose an accent extraction model with VAE, which is often used in unsupervised learning of latent representations of speech [6]. Figure 3a shows the structure of the VAE model. The boxes of "$\boldsymbol{\mu}$" and "$\boldsymbol{\Sigma}$" in the figure mean linear layers that output mean vector $\hat{\boldsymbol{\mu}}$ and diagonal variance matrix $\hat{\boldsymbol{\Sigma}}$, respectively. The posterior distribution of latent variable $\boldsymbol{z}$ is defined as a Gaussian distribution with mean $\hat{\boldsymbol{\mu}}$ and variance $\hat{\boldsymbol{\Sigma}}$. Following [5], we define the loss function $\mathcal{L}_{\mathrm{VAE}}$ as follows:

$$\mathcal{L}_{\mathrm{VAE}} = \sum_{i=1}^{N_d} \left\{ \|\boldsymbol{y}_{\mathrm{F0}}^i - \hat{\boldsymbol{y}}_{\mathrm{F0}}^i\|^2 + D_{\mathrm{KL}}[\mathcal{N}(\hat{\boldsymbol{\mu}}^i, \hat{\boldsymbol{\Sigma}}^i) || \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})] \right\} \tag{1}$$

where N_d is the number of speech samples, $\mathcal{N}$ means Gaussian distribution and D_{KL} means the Kullback–Leibler divergence. $\boldsymbol{I}$ means an identity matrix.

4.3. VQ-VAE model

Since the Japanese accent information that people perceive is discrete as described in Sec. 2, we adopt VQ-VAE, which quantizes the latent space, to take advantage of this discrete characteristic of the Japanese pitch accent. Figure 3b shows the structure of the VQ-VAE model. The vector quantization layer quantizes the output of the previous linear layer. Following [7], we



(a) *Accent extraction model using VAE and VQ-VAE*

(b) *APM-RNN*

(c) *APM-DT*

Figure 3: *Summary of proposed models. (a) is accent extraction model and (b) and (c) are accent prediction models.*

define the loss function $\mathcal{L}_{\mathrm{VQ-VAE}}$ as follows:

$$\mathcal{L}_{\text{VQ-VAE}} = \sum_{i=1}^{N_d} \Big\{ \|\boldsymbol{y}^i_{\text{F0}} - \hat{\boldsymbol{y}}^i_{\text{F0}}\|^2 + \|sg(\boldsymbol{z}^i_{\text{uq}}) - \boldsymbol{z}^i\|^2 + \beta \|sg(\boldsymbol{z}^i) - \boldsymbol{z}^i_{\text{uq}}\|^2 \Big\} \quad (2)$$

where $\boldsymbol{z}_{\text{uq}}$ are the values of $\boldsymbol{z}$ before quantization by the VQ layer, and function $sg(\cdot)$ stops the gradient. β was set to 1 in the experiment.

5. Accent prediction models

In this section, we propose two accent prediction models that predict accent latent variables from linguistic features, for synthesizing speech of the Osaka dialect with only text. The relationship between the proposed accent extraction models and the accent prediction models is shown in Figure 3. The accent prediction models predict accent latent variables, and the F0 curve is synthesized by inputting the predicted accent latent variables into the decoder. One uses RNNs and the other uses a decision tree [13]. For both of these two models, there are two candidates that take different input features. One takes only the linguistic features of text as the input. The other takes the linguistic features of the text and accent information of the Tokyo dialect. Using accent information of the Tokyo dialect as the input is possibly useful for the models to learn the corresponding relationships between the accents of Tokyo and Osaka dialects, hypothesizing that there generally exist the correspondences as described in Sec. 2. We examine the impact of accent information of the Tokyo dialect on predicting accents of the Osaka dialect by comparing the results of these two candidates.

5.1. Accent prediction model using RNNs (APM-RNN)

This model uses RNNs to predict the accent latent variables. Since this model adopts deep learning techniques, it can capture more complex features than the decision tree model. It takes phoneme-level linguistic features and accent information of the Tokyo dialect, and outputs the accent latent variables. The structure is almost the same as the decoder of the accent extraction models. The differences are that this model does not take the latent variable as the input, and that the outputs of this model are accent latent variables, not F0 curve.

5.2. Accent prediction model using decision tree (APM-DT)

This model uses a decision tree to predict the accent latent variables. For this model, we expect robustness, because the correspondences as shown in Table 1 are so simple that RNN models may be too expressive. Since the decision tree can output only a scalar value, we define a decision tree model for each mora index. As the input, this model takes linguistic feature vector, and accent latent variables of preceding moras to consider time series feature of accent. When predicting the accent latent variable of a four-mora word, four decision trees are used.

6. Experiments

6.1. Experimental conditions

We used a subset of the JSUT corpus, BASIC5000 [14], which consists of 5000 utterances of sentences spoken in the Tokyo dialect by a female speaker, and OSAKA3696, which consists of 3696 utterances of phrases spoken in the Osaka dialect by a male speaker. The phrases were composed of 258 verbs, 156 adjectives and 930 nouns and each phrase consisted of one content word and a positional particle. Since Japanese verbs and adjectives are conjugated depending on the postpositional particle or auxiliary verb, all conjugated forms were recorded for each verb and adjective with postpositional parts. Nouns were recorded with the postpositional particle "wa" because the accent of a noun affects the accent of a postpositional particle. We used 3000 utterances of BASIC5000, and 3126 utterances of OSAKA3696 for training, 285 of OSAKA3696 for validation, and 285 of OSAKA3696 for testing. The reason we also used the Tokyo dialect corpus was to make training stable.

Based on the context label of Japanese HTS [15], the linguistic feature vector for the accent extraction model was defined as a 444-dimensional one, which consisted of phoneme information, parts of speech, and one-hot speaker embedding. The linguistic feature vector for the APM-RNN was defined as a 442-dimensional one, which consisted of phoneme information, parts of speech. The accent information vector of the Tokyo dialects (Tokyo accent vector) was defined as a 91-dimensional one. For the APM-DT model, we used a phrase-level 159-dimensional vector including parts of speech as the linguistic feature vector. As the accent information vector of the Tokyo

Table 2: *RMSEs of reconstructed F0 [cent] using extracted accent latent variable*

model	F0 RMSE [cent]
VAE	216
VQ-VAE	**172**
NO-ALV	247

dialect, we used the same Tokyo accent vector as the APM-RNN.

The sampling rate of all speech signals was 48 kHz, and the frame shift length was set to 5 ms. The acoustic features were defined as the 0–59th mel-cepstral coefficients, continuous log F0, five-band aperiodicity, first and second derivatives of all these parameters, and a voiced/unvoiced flag. WORLD [16] was used for parameter extraction and waveform synthesis. As pre-processing of F0, trajectory smoothing [17] with a 10 Hz cutoff frequency was used. The number of classes of VQ-VAE latent space was set to 2, on the basis of the accent system of Japanese as described in Sec. 2. The basic structure of the DNN models (encoder, decoder, APM-RNN) consisted of a linear layer, 2 $\times$ bi-directional LSTM layer with 734 cells, and a linear layer. For the VAE encoder, the last linear layer was replaced with two linear layers of $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ as shown in Figure 2a. For the VQ-VAE encoder, VQ layer was added as the last layer as shown in Figure 2b. The maximum depth of decision tree was set to 11.

6.2. Evaluations of accent extraction models

6.2.1. Objective evaluations of accent extraction models

To evaluate the performance of the accent extraction models, we calculated the root mean squared errors (RMSEs) of the F0 curves reconstructed by the accent extraction models. The results are shown in Table 2. "NO-ALV" means a model that directly predicted F0 curve without accent latent variable (ALV), and had the same structure as the decoder. The F0 RMSEs of both the VAE and VQ-VAE models were smaller than NO-ALV, which did not use the accent latent variable. This implies that the proposed accent extraction models succeeded in extracting accent information as latent variables. Moreover, the RMSE of the VQ-VAE model was 172 cent, which was smaller than that of the VAE model. This implies that the discrete representation of the two classes was more suitable for representing high/low Japanese accent.

6.2.2. Subjective evaluations of accent extraction models

To confirm the effectiveness of the accent extraction models also in a subjective evaluation, we conducted an XAB test on the accent reproducibility. The evaluation was done by 30 listeners on our crowdsourcing system with speech samples vocoded with reconstructed F0 curve, original mel-cepstrum, and original band aperiodicity. The listeners were asked to answer which of two accents of synthetic speech samples was closer to the original one. Table 3 shows the results. As shown in the Table, the F0 curves created by the VAE and VQ-VAE models were significantly closer to the original speech than that of the NO-ALV. Moreover, the F0 curve created by the VQ-VAE model was significantly closer to the original than that of the VAE model. The effectiveness of the proposed accent extraction models and quantization were confirmed also in the subjective evaluation.

Table 3: *XAB test results of accent extraction models*

model A				p-value	model B
VAE	**0.591**	vs.	0.401	$< 10^{-5}$	NO-ALV
VQ-VAE	**0.700**	vs.	0.300	$< 10^{-5}$	NO-ALV
VAE	0.375	vs.	**0.625**	$< 10^{-5}$	VQ-VAE

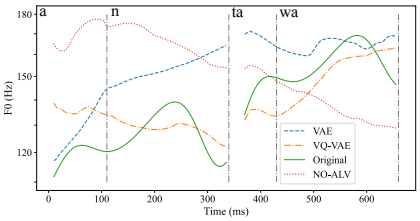


Figure 4: *F0 plot of "a-n-ta-wa" synthesized by accent extraction models. The vertical dotted lines mean the borders of mora.*

6.2.3. Synthesized F0 curves

The VQ-VAE model succeeded in extracting the accent than the VAE model. Here, we look into the the reconstructed F0 curves of the models. Figure 4 shows F0 curves for the phrase "a-n-ta-wa"(a noun "a-n-ta" and a postpositional particle "wa") synthesized by the accent extraction models. The accent of the original speech signal was low/low/low/high. Since only "wa" had high accent in the phrase, the original F0 curve had a higher value for "wa" than the others. The predicted F0 curve of the VQ-VAE model had the same tendency as the original one. However, the F0 curves of the other methods were different from the original one. The F0 curve of the VAE model had high values not only in "wa", but also in "n" and "ta". The F0 curve of the NO-ALV was a simple declination, which is far from the original one. The VQ-VAE model succeeded better in reconstructing the F0 curve of the original speech than the other models.

6.2.4. Examples of accent latent variable of VQ-VAE model

We confirmed that the VQ-VAE model succeeded in extracting accent information as latent variable better than the VAE model. In this section, we check how the extracted latent variables look like. Examples of the extracted latent variables extracted by the VQ-VAE model are shown in Figure 5. Compared with manual labels that we annotated to a part of the corpus, we found that, one of the classes (Class 1) of the latent space tended to correspond to high, and the other (Class 2) tended to correspond to low. Since the VQ-VAE had better results, we adopted the VQ-VAE model as our accent extraction model. The accent latent variables extracted by the VQ-VAE model were used as the teacher labels for the accent prediction models.

6.3. Results of accent prediction models

6.3.1. Objective evaluation of predicted F0

To measure the quality of the F0 curves predicted by the accent prediction models, we calculated the RMSEs of the F0 curves for three parts of speech (verbs, nouns, and adjectives) in OSAKA3696. Table 4 shows the objective evaluation results of the predicted F0 curves. "W/" means that the input included accent labels of the Tokyo dialect, and "W/O" means that the input did not include them. The F0 RMSE of the APM-RNN W/ was the smallest (256 cent), while that of the APM-DT W/O was

Phrase 1: スレバ (su-re-ba)

Mora	ス (su)	レ (re)	バ (ba)
Annotated accent label	High	Low	Low
Class of extracted latent variable	1	2	2

Phrase 2: ゼンブハ (ze-n-bu-wa)

Mora	ゼ (ze)	ン (n)	ブ (bu)	ハ (wa)
Annotated accent label	Low	High	Low	Low
Class of extracted latent variable	2	1	2	2

Phrase 3: オイシイ (o-i-shi-i)

Mora	オ (o)	イ (i)	シ (shi)	イ (i)
Annotated accent label	Low	Low	High	Low
Class of extracted latent variable	2	1	1	2

Figure 5: *Example of accent latent variables extracted by VQ-VAE model*

Table 4: *RMSE of reconstructed F0 [cent] for each part of speech*

model	all	verb	noun	adjective
APM-DT W/	313	289	351	323
APM-DT W/O	321	287	368	322
APM-RNN W/	**256**	**239**	**334**	**215**
APM-RNN W/O	272	241	365	222

the largest (323 cent). The RMSEs of F0 of the APM-DT were much larger than those of the APM-RNN, which implies that APM-DT was not expressive enough to predict the accent latent variables. All models with the Tokyo accent labels had better prediction results compared with those without the Tokyo accent labels. As for the difference among parts of speech, the effect of adding accent labels of the Tokyo dialect was relatively small in verbs and adjectives compared with nouns. One of the causes may be that the accents of verbs and adjectives of the Osaka dialect have a few fundamental patterns. For example, accent labels of an n-mora adjective are fundamentally defined as high/.../high/low/low. This may make the accent of them easy to predict without the accent information of the Tokyo dialect.

6.3.2. Subjective evaluation of predicted F0

In addition to the objective evaluation, We conducted XAB tests on the accent reproducibility of the predicted F0 curves to check the prediction performance of the APMs. This subjective evaluation was done by two groups with speech samples vocoded with predicted F0 curves, original mel-cepstrum, and original 5 band aperiodicity. One was done by 30 listeners on our crowdsourcing system. The other was done by 30 listeners who speaks Osaka dialect. The listeners were asked which of two accents was similar to the original one, in the same way as Sec. 6.2.2. Table 5 shows the results of the evaluation by our crowdsourcing system, and Table 6 shows those by Osaka citizens. The results of both evaluations were similar. As both of the tables show, the APM-RNN W/ had significantly better performance than the APM-DT W/. There was no significant difference between the W/ models and the W/O models.

Since the degradation of RMSE in nouns by adding the Tokyo accent labels were larger than other parts of speech, we additionally conducted a subjective evaluation experiment, by limiting the test utterances to nouns. The experiment was done only on our crowdsourcing system, as the results of Osaka citizens and our crowdsourcing system were similar. The results are shown in Table 7. The APM-RNN W/ was significantly better at reproducing the accents of the Osaka dialect nouns than the APM-RNN W/O. It is estimated that adding accent labels of the Tokyo dialect was useful in predicting the accents of nouns of the Osaka dialect.

Table 5: *Subjective evaluation of predicted F0 by crowdsourcing system*

model A				p-value	model B
APM-DT W/	0.375	vs.	**0.625**	$< 10^{-5}$	APM-RNN W/
APM-DT W/	0.519	vs.	0.481	0.35	APM-DT W/O
APM-RNN W/	0.498	vs.	0.502	0.96	APM-RNN W/O

Table 6: *Subjective evaluation of predicted F0 by Osaka citizens*

model A				p-value	model B
APM-DT W/	0.334	vs.	**0.666**	$< 10^{-5}$	APM-RNN W/
APM-DT W/	0.533	vs.	0.467	0.12	APM-DT W/O
APM-RNN W/	0.511	vs.	0.489	0.64	APM-RNN W/O

6.3.3. Predicted F0 curves

The APM-RNN succeeded better in reproducing the accents of the Osaka dialect than the APM-DT. Here, we look into the predicted F0 curves of an adjective. Figure 6 shows the predicted F0 curves for a five-mora adjective "a-ri-ga-ta-i", whose accent labels of the Osaka dialect are high/high/high/low/low. Since the last two moras of the term have low accent, the values of original F0 of them tend to be smaller than those of former three moras. The predicted F0 curve of the APM-RNN W/ had a similar tendency to the original one, which can be perceived as the same accent high/high/high/low/low. However, The F0 curve of the APM-DT W/ fell around third mora "ga", which can be perceived as a wrong accent, high/high/low/low/low.

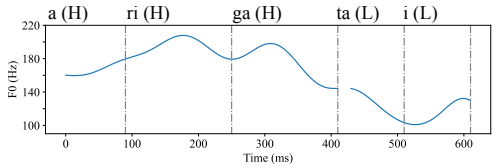
7. Conclusions

In this paper, we have proposed accent extraction models and accent prediction models for automatic accent modeling of the Osaka dialect. The result showed that the proposed accent extraction model succeeded in extracting accent information as latent variable using VQ-VAE. This model will make it possible to train an acoustic model that synthesizes natural F0 curve without annotated accent labels, which is one of the problems that TTS of Japanese non-Tokyo dialects is suffering from. For the accent prediction models, the result showed that the APM-RNN reproduced the accent of the Osaka dialect better than the APM-DT, and adding the accent labels of the Tokyo dialect is useful for predicting the accent of nouns of the Osaka dialect.

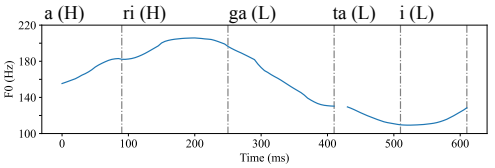
Combining the proposed accent extraction models and accent prediction models enables us to synthesize speech of texts without speech samples. Although the RMSEs of the proposed prediction models were still larger than that of the model without accent latent variables (NO-ALV), the proposed synthesis

Table 7: *Subjective evaluation of predicted F0 of nouns*

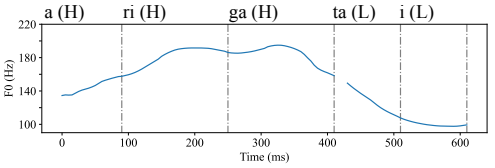
model A				p-value	model B
ALV-DT W/	0.526	vs.	0.474	0.35	ALV-DT W/O
ALV-RNN W/	**0.657**	vs.	0.343	$< 10^{-2}$	ALV-RNN W/O



(a) *Original*



(b) *APM-DT W/*



(c) *APM-RNN W/*

Figure 6: *Predicted F0 curve of an adjective, "a-ri-ga-ta-i". The horizontal and vertical axes mean Time and F0 respectively, and the dashed lines and the labels above mean the border of mora, the phoneme of mora and how people can perceive the accent of the mora.*

methods have some advantages such as:

- Interpretability:
  Looking into the input accent latent variables enables us to understand how the accents of speech utterances were synthesized.
- Controllability:
  Changing the input accent latent variables enables us to easily modify the accent of synthesized speech into more natural one.

Moreover, the proposed accent extraction models are possibly useful for an accent analysis of low resourced dialects, since they can easily visualize the accent information only with texts and speech utterances, even without professionals of the accent of the dialect.

Future work includes:

- Apply the proposed models to other dialects of pitch accent languages including Japanese
- Research model structures of the accent extraction models for better representation of the accent
- Incorporate modification systems or other input features into the proposed accent prediction models for better prediction
- Extend the proposed models to extract other features of speech signals such as emotion and dialog acts.

8. Acknowledgements

This work was supported by JSPS KAKENHI Grant Number 18K18100, 19K20292.

9. References

[1] M. Pucher, D. Schabus, J. Yamagishi, F. Neubarth, and V. Strom, "Modeling and interpolation of Austrian German and Viennese dialect in HMM-based speech synthesis," *Speech Communication*, vol. 52, no. 2, pp. 164–179, 2010.

[2] Y. Yasuda, X. Wang, S. Takaki, and J. Yamagishi, "Investigation of enhanced Tacotron text-to-speech synthesis systems with self-attention for pitch accent language," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 6905–6909.

[3] T. Koriyama and T. Kobayashi, "Semi-supervised Prosody Modeling Using Deep Gaussian Process Latent Variable Model." in *INTERSPEECH*, 2019, pp. 4450–4454.

[4] "openjtalk," http://open-jtalk.sp.nitech.ac.jp/.

[5] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," in *International Conference on Learning Representations*, 2014.

[6] T. Kenter, V. Wan, C.-A. Chan, R. Clark, and J. Vit, "CHiVE: Varying prosody in speech synthesis with a linguistically driven dynamic hierarchical conditional variational network," in *International Conference on Machine Learning*, 2019, pp. 3331–3340.

[7] A. van den Oord, O. Vinyals, and K. Kavukcuoglu, "Neural Discrete Representation Learning," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, p. 6309–6318.

[8] S. Kawahara, "The phonology of Japanese accent," *The handbook of Japanese phonetics and phonology*, pp. 445–492, 2015.

[9] H. Kindaichi, "Akusento no bunpu to hensen," *Iwanami kouza nihongo*, vol. 11, pp. 129–180, 1977, in Japanese.

[10] Y. Wang, D. Stanton, Y. Zhang, R.-S. Ryan, E. Battenberg, J. Shor, Y. Xiao, Y. Jia, F. Ren, and R. A. Saurous, "Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis," in *International Conference on Machine Learning*. PMLR, 2018, pp. 5180–5189.

[11] Y. Zhao, H. Li, C.-I. Lai, J. Williams, E. Cooper, and J. Yamagishi, "Improved prosody from learned f0 codebook representations for vq-vae speech waveform reconstruction," *arXiv preprint arXiv:2005.07884*, 2020.

[12] Z. Hodari, A. Moinet, S. Karlapati, J. Lorenzo-Trueba, T. Merritt, A. Joly, A. Abbas, P. Karanasou, and T. Drugman, "Camp: a two-stage approach to modelling prosody in context," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 6578–6582.

[13] S. R. Safavian and D. Landgrebe, "A survey of decision tree classifier methodology," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 21, no. 3, pp. 660–674, 1991.

[14] S. Takamichi, R. Sonobe, K. Mitsui, Y. Saito, T. Koriyama, N. Tanji, and H. Saruwatari, "JSUT and JVS: Free Japanese voice corpora for accelerating speech synthesis research," *Acoustical Science and Technology*, vol. 41, no. 5, pp. 761–768, 2020.

[15] "HTS," http://hts.sp.nitech.ac.jp/.

[16] M. Morise, F. Yokomori, and K. Ozawa, "World: a vocoder-based high-quality speech synthesis system for real-time applications," *IEICE Transactions on Information and Systems*, vol. 99, no. 7, pp. 1877–1884, 2016.

[17] S. Takamichi, K. Kobayashi, K. Tanaka, T. Toda, and S. Nakamura, "The NAIST text-to-speech system for the Blizzard Challenge 2015," in *Proc. Blizzard Challenge workshop*, 2015.

11th ISCA Speech Synthesis Workshop (SSW 11)
26-28 August 2021, Budapest, Hungary

Liaison and Pronunciation Learning in End-to-End Text-to-Speech in French

Jason Taylor[1], *Sébastien Le Maguer*[2], *Korin Richmond*[1]

[1] The Centre for Speech Technology Research, The University of Edinburgh.
[2]Sigmedia Lab, ADAPT Centre, School of Engineering, Trinity College Dublin, Ireland

jason.taylor@ed.ac.uk, lemagues@tcd.ie, korin@cstr.com

Abstract

Sequence-to-sequence (S2S) TTS models like Tacotron have grapheme-only inputs when trained fully end-to-end. Grapheme inputs map to phone sounds depending on context, which traditionally is handled by extensive preprocessing in the TTS front-end. However, French orthography does not provide a clear one-to-one mapping between graphemes and sounds, and in English, which similarly has rather non-phonetic orthography, pronunciations are a significant cause of error in S2S-TTS with grapheme-inputs. In this paper, we test implicit pronunciation knowledge where graphemes do not map directly to phones. Implicit pronunciation knowledge learnt in S2S-TTS is similar to a standalone grapheme-to-phoneme (G2P) model, which makes explicit phone predictions at the sequential level. We find grapheme-input S2S-TTS makes implicit pronunciation errors similar to explicit G2P models - notably for foreign names. In a traditional front-end pipeline, there are also post-lexical rules which modify G2P output at the sequential level. In French, post-lexical rules require a deep knowledge of linguistic structure in a process called *Liaison*. Without explicit rules, we find S2S-TTS with grapheme-inputs over-inserts *Liaison* sounds, leading to a significant preference for a phone-based equivalent. By testing with linguistically-motivated stimuli, we observe differences that would otherwise go undetected.
Index Terms: Text-to-Speech, Phoneme, *Liaison*, *Enchaînement*

1. Introduction

Neural text encoders enable text-to-speech synthesis from raw text-audio pairs without extensive text normalisation and/or linguistic preprocessing such as lexicon and G2P model lookups. Traditionally these initial steps, formulated in the front-end, ensured correct pronunciations and provided useful information for modules further down the text-to-speech pipeline. With the rise of end-to-end (E2E) TTS with Tacotron [1] and subsequent text encoders [2, 3], the extent and need for a front-end for TTS is in question.

In [4], implicit pronunciation knowledge learned in a grapheme-based Tacotron was framed as a G2P model trained on the text from training datasets in English such as LJ [5] and VCTK [6]. Implicit G2P models were poorer than lexicon-based G2P models, being unable to pronounce place names and foreign names - especially those with non-phonetic orthography in English.

French also has non-phonetic orthography. In [7], the use of graphemes and phones were analysed as input features. The authors visualised embedded grapheme-input with t-SNE, observing single graphemes in context can map to multiple phone sounds. The authors sampled 50 sentences from the SIWI dataset in a MUSHRA comparing graph and phone input. Listeners were also asked to rate the pronunciation of the samples on a scale from 1-5 in a MOS-style test. Grapheme and phone-input performed with no-significant difference in these tests. In addition, tongue twisters were tested to measure pronunciation learning abilities, also with no significant difference found. The authors noted that both grapheme and phone-input based systems produced errors in the pronunciation of *Liaison*, but did not formally test this difficulty.

In this paper, we compare grapheme and phone input for French E2E-TTS using linguistically motivated stimuli. We first target stimuli to test the implicit G2P model and disallowed cases of *Liaison*. Under *Liaison*, phones may be inserted between word boundaries (me**s** amis, mo**n** amour). The post-lexical rules governing *Liaison* derive from linguistic information such as part-of-speech (POS) tags and semantic roles (subject, object, etc). We think phonetic control of *Liaison* is handled more reliably when using phones as a representation.

We proceed to add syllable boundaries to input to test another supra-segmental process in French known as *Enchaînement*. In French, syllables span word-boundaries so that consonants are not left at the end of syllables (eg, mon cher ami - mon. che. **r**ami). We test using stimuli containing examples of *Enchaînement*.

Overall, we find there are definite differences in pronunciations between grapheme- and phone-inputs in French, and these differences are revealed when using linguistically targeted stimuli.

2. Previous Work

2.1. Linguistic Features in Tacotron

The TTS front-end consists of a pipeline of processes to normalise input text and generate a linguistic specification for use by neural encoders, duration/prosody models, and vocoders. E2E TTS is an approach that aims to simplify the traditional modular TTS pipeline. The first Tacotron paper demonstrated high quality E2E-TTS was possible with grapheme-input, although the authors noted pronunciation errors were common and performance was enhanced with a front-end [1].

Some pronunciation issues derive from text normalisation. For instance the string '3' may be ordinal or cardinal, or abbreviations such as stock-ticker symbols can have ambiguous pronunciations. Traditionally, such errors have been averted using rule-based verbalisers. While the general performance of RNN-based verbalisers is accurate, some errors are irrecoverable and unacceptable for deployed systems [8]. RNN-based errors require an FST filter, a core problem presented in the Kaggle-hosted Text Normalisation Challenge [9], where the hosts noted the high degree of manual rule-writing for the top performing systems [10]. There is a recent drive to verbalisation that shares a unified representation across ASR and TTS [11] enabling swift rollout of FST filters to low resource languages using a template-based questionnaire [12].

10.21437/SSW.2021-34

Relatedly, pronunciation errors may also derive from a lack of deeper linguistic knowledge learned implicitly from text-audio pairs in the dataset. Increasingly, research demonstrates augmenting E2E-TTS with linguistic features improves quality in English, such as with phones [13] or with morphemes [14]. Pronunciation correction is also possible when mixing input representations between graphemes, phones and syllables [15, 16]. For non-alphabetic languages such as Japanese and Chinese, phones are preferred to characters to avoid large character sets. In these languages, the implicit pronunciation model does not learn pitch or other prosodic information meaningfully. Contextual linguistic features such as the mora [17] and pitch accents [18] are helpful, although such contextual features must be compact to be beneficial [19]. Such features were used in [20] with simplified alignments.

Recently in English the field has also used linguistic features to improve prosody: using syllabic stress [21], semantic and syntactic features [22, 23] and pre-trained language model embeddings [24, 25]. Clockwork RNNs were also used to hierarchically encode linguistic features at varying levels in [26], a hierarchical encoder having previously helped in DNN-based TTS [27, 28].

2.2. French Pronunciation

Recently, grapheme and phone inputs were tested in a French Tacotron model [7], with the authors finding no significant difference between the two inputs in a MUSHRA listening test. They chose samples from a random test set, however, which can mask subtle but important differences between systems. It was proposed in [29] for instance that listening test samples should instead be chosen containing large differences in acoustic mismatch. Tongue twisters were also tested in [7], with no significant difference found between grapheme and phone inputs. While the rapid repetition of certain articulations are difficult for humans to pronounce, we posit the grapheme-to-sound relations contained in tongue twisters are usually unambiguous and thus not an appropriate way to test implicit pronunciation learning. Instead, we target test stimuli to evaluate particular G2P and post-lexical challenges for E2E-TTS in French: G2P error words, *Liaison* and *Enchaînement*.

With grapheme-input, the text-encoder learns pronunciations implicitly while learning acoustic features. In TTS, data driven G2P models are typically trained with more than 100,000 entries from a pronunciation lexicon. While G2P conversion is regular in French, the training data is restricted in vocabulary covering fewer words than in a lexicon and G2P relations of foreign words. Figure 1 shows that the full size of the SIWI and CSS10 French datasets have limited word coverage. In [4], the authors demonstrated explicit G2P models trained on words in TTS training data underperformed G2P models trained on a full lexicon in English. They also showed G2P error words were mispronounced by grapheme-input E2E-TTS. Likewise here, we test the pronunciation of grapheme- and phone-input models using stimuli containing words with inaccurate G2P conversion.

We also test *Liaison* which is a process where linking sounds are inserted between words. Traditionally, it occurs during the "post lexical" module of a TTS front-end, after an initial phone string has been obtained from a lexicon lookup or G2P model. The plural possessive '*mes*' before a following consonant has no pronunciation corresponding to the 's' grapheme: mes chats - [me . ʃa]. But before a following vowel, the 's' grapheme corresponds to the pronunciation [z]: mes amis - [me. **z**a. mi] The rules governing *Liaison* operate at a deep linguistic level which are difficult to model. For instance, *Liaison* is disallowed after a singular noun. While data modelling of *Liaison* has been tested with decision trees [30] and templates [31], the process is complicated further because its use is often stylistic and optional [32], consequently hand-written rules are often used for TTS. [7] notes that grapheme-input Tacotron does insert *Liaison* sounds but does not learn when to use it appropriately. Their phone-input model also made *Liaison* errors, but their front-end used a low-accuracy rule-based G2P system and did not use post-lexical *Liaison* rules. We re-evaluate grapheme and phone-based *Liaison* using a test set of disallowed *Liaisons*.

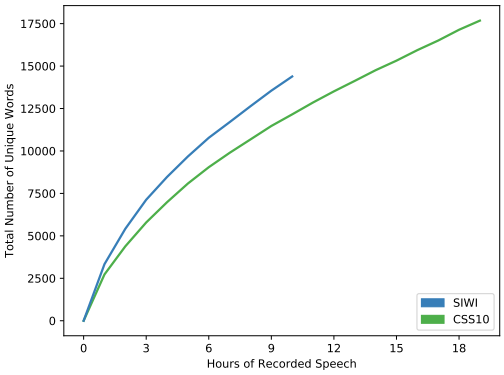


Figure 1: *Total unique words in SIWI and CSS10 French TTS Datasets. The datasets cover fewer unique words than standard pronunciation lexica which typcially contain more than 100,000 entries. Unusual G2P relations not covered in the training data may not be predicted accurately, such as in foreign names.*

Enchaînement occurs when the final sound of one word transfers to the first syllable of the next word. For instance, in *mon cher ami* the final rhotic of the word 'cher' is the onset to the syllable of the next word *ami* - [mõ . ʃɛ . **ʁ**a . mi] A multi-task G2P with syllabic boundaries included in output was shown to improve G2P performance in 14 languages [33], although French was not included in their reported results. As noted above, contextual phone information has been helpful in mora-based languages such as Japanese.

3. Methods

3.1. Tacotron Model

The Tacotron model we use for our experiments here [34], has a pre-net and CBHG module to encode a series of one-hot input characters into a single representation. Unlike previous DNN-based systems, a sequential text encoder and attention mechanism align input text to audio directly, enabling grapheme-based input. We used Location Sensitive Attention (LSA) to reduce instability in output speech as recommended in [2]. Each Tacotron was trained for 350k training steps. We use a WaveRNN vocoder based on [35], trained using Tacotron's predicted outputs up to 2000k steps, and synthesised samples in batch-mode. We used a sampling rate of 16kHz.

3.2. Data

3.3. Front-End

For our phone-based systems, we used the French front-end from MaryTTS [36], with its default lexicon and G2P model. The lexicon is based on the database Lexique [37] and each word has been phonetized as well as syllabified using *LIA PHON* [38] whose Phone-Error-Rate is 1.3%. However, in contrast to *LIA PHON*, MaryTTS doesn't provide post-lexical rule-based phonetization such as *Liaison*. Therefore, we manually wrote *Liaison* post-lexical rules based upon the guide available in [39]. Since POS tagging was a core input attribute we used the Stanford POS tagger [40] to ensure as high accuracy as possible.

3.4. Experiments

We ran AB preference tests on 10 sentences held-out from the CSS10 dataset between: i) graphemes (G) and phones (P) as input; and ii) phones (P) and phones enriched with syllable boundaries (S). The general AB tests complement the targeted AB test results.

To test the implicit knowledge of French pronunciation in the grapheme-based Tacotron, we applied the method used in [4] to test implicit pronunciation learning of grapheme-based Tacotron in English: train a G2P model using the TTS training data, identify and synthesise G2P error words with the Tacotron model. We used OpenNMT [41] for G2P modelling. We placed 10 problematic words in carrier sentences and synthesised them using the G and *Liaison* P systems.

To test *Liaison*, we hand-crafted a test set of 10 sentences, each containing disallowed *Liaisons*. As noted in [7], disallowed cases of *Liaison* are problematic for Tacotron - for example where an s is inserted before an aspirated-h as in *les_haricots*. We submitted the G and *Liaison* P systems to a forced choice test for preference.

To test *Enchaînement*, we hand-crafted a test set of 10 sentences, each containing cases where the word-final consonant becomes the onset of the following word-initial syllable. We did augment the G model here as syllable strings could only be derived from phone-based systems. We synthesised samples from the *Liaison* phone-input model (containing word boundaries) and the *Enchaînement* phone-input systems for an AB preference test.

We built the AB preference tests in Qualtrics. Due to social distancing policies, we held our listening test online using the Prolific platform. We used 30 participants. Participants were paid £5 per 30 minutes of their time. Participants were native French speakers and had no known hearing difficulties. We did not allow participants to take the test on their mobile phones - forcing them to use a desktop. For the general and targeted preference tests the accompanying question on each screen was: *Which clip has better pronunciation?/ (Quel clip a la meilleure prononciation?)*[1]

3.5. Results

3.6. CSS10 Test Stimuli

The results from the general AB listening test are shown in Figure 2. No significant differences were found between the G and P systems, nor between the P and S systems.

[1]We encourage the reader to listen to samples using this link: http://homepages.inf.ed.ac.uk/s1649890/fren/

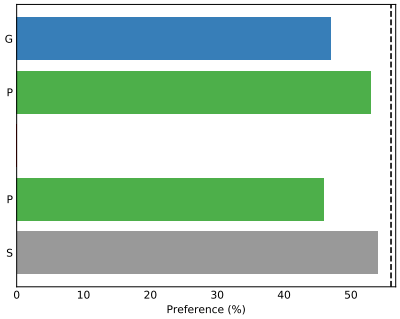


Figure 2: *Results from preference tests using CSS10 stimuli. No significant differences were observed between grapheme-input (G), phone-input (P) and phones enriched with syllable boundaries (S). The significance level at p = 0.05 is shown by the black dotted line at x=57.*

3.7. Targeted stimuli

The results from the targeted AB listening test are shown in Figure 4.

3.7.1. Words of inaccurate G2P

The phone-input models had accurate phone labels for this targeted preference test. Listeners significantly preferred the phone-based model over the grapheme-based model. Some incorrect pronunciations by system G are shown in Figure 3.

The words contain unusual G2P relations in French missing from the TTS training data. Representation mixing [16, 15] may correct pronunciations provided the reader has a large enough pronunciation lexicon to label a sufficient quantity of training data.

3.7.2. Liaison *stimuli*

Listeners significantly preferred the phone-based system. The French language has a highly active normative body called the Academy (l'Académie Française) who maintain a strict standard form of the language prohibiting insertion of *Liaison* sounds in certain contexts, such as before the aspirated h in combinations like *les **h**aricots* or *les **h**érissons*. While speakers do not strictly obey all rules, the prescribed norm of correct pronunciation remains, and incorrect *Liaison* insertion was perceived by listeners.

Word	G (Incorrect)	P (Correct)
Miguel de Cervantès	[**d**igɛl də sɛʁvā**tz**]	[migɛl də sɛʁvātɛz]
Les Coopers	[**te sk**opə]	[le kypɛ]
Monica Lewinsky	[**pw**anika lewẽ**s**i]	[monika lywinski]
Rio de Janeiro	[**tʁ**io də ʒanero]	[ʁio də ʒanero]
McLaren	[**klaʁno**]	[məklaʁɛn]

Figure 3: *IPA transcriptions of words of inaccurate G2P included in preference test. Mispronunciation of names by the G model are highlighted in bold.*

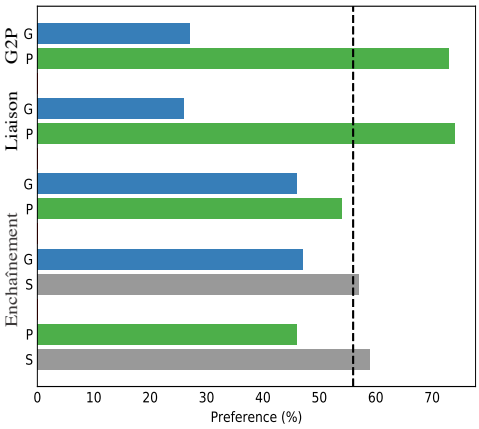


Figure 4: *Results from targeted preference Test. The first tier shows G2P results, the second tier shows* Liaison*. The last 3 tiers show results from the test with* Enchaînement *stimuli.*

3.7.3. Enchaînement *stimuli*

No significant differences were observed between G and P, but there was a preference for S over P. With syllable boundaries replacing word-boundaries, prosodic breaks occurred between syllables and less so at word boundaries.

4. Discussion

To compare grapheme and phone inputs, consider that phone inputs result from a pipeline of complex processes in the front-end. The final quality of phone labels depends on processes such as the pronunciation lexicon, the G2P model and post-lexical rules. Error propagation from these processes may contribute to phone-label inaccuracies, as was noted in [7] where *Liaison* errors were observed in the phone-based system. However, phones are preferred where graphemes do not offer the same level of control. Thus, we highlight the importance of linguistically motivated stimuli to observe the differences in pronunciation of G2P error words and *Liaison* for phones and graphemes.

5. Conclusion

We investigated pronunciation learning with a Tacotron model's text-encoder when using grapheme inputs in French. Grapheme inputs from raw or minimally normalised text reduce preprocessing required to build TTS voices. However, graphemes are not accurate phonetic labels so the text encoder learns an implicit, data-driven G2P model. Previous work had found implicit G2P models to be weaker than explicit data-driven G2P models trained on pronunciation lexica. The paucity of Tacotron's implicit G2P model was observed when synthesising problematic words identified by dedicated G2P models.

We used AB preference tests to compare listener opinions on pronunciation. Using sentences from the speaker dataset we find no significant differences between grapheme or phone-input. When we use sentences containing G2P "error words" we find the grapheme-based system makes mispronunciations and the phone-based model is preferred.

Liaison is a post-lexical insertion of consonant sounds that obeys complex rules. The rules governing correct *Liaison* insertion are complex and require deep linguistic labels. Knowledge about the etymology of a word may also be required in the case of disallowed *Liaisons* before the aspirated 'h'. Whilst speakers do not always obey strict *Liaison* rules, correct *Liaisons* from a phone-based model were preferred to *Liaison* over-insertion by the grapheme-based model.

We proceeded to test whether pronunciation of enchainement was improved by substituting word boundaries for syllable boundaries. We found that in sentences with word boundaries there were pauses at word boundaries where enchainement should occur. Listeners significantly preferred syllable boundaries in these sentences.

Overall, we find linguistically-motivated stimuli reveal differences in pronunciation learning between graphemes and phones which are not revealed when considering averaged scores from a held-out sample of TTS training data.

Input	Labels
G	Le **h**aricots pousseront plus efficacement en plein air. Il a mi**s** **u**ne chemise.
P	[le aʁiko pusəʁõ plys efikasəmã ã plẽn ɛʁ] [il a mi yn ʃəmiz]

Figure 5: Liaison *inserts sounds at word boundaries according to complex rules, but inadequate insertion such as after aspirated-h or between a past participle and a determiner was dispreferred. Inaequate* Liaisons *are highlighted in bold.*

Input	Labels
G	Le <> ciel <> est <> bleu <> et <> la <> mer <> aussi Les <> sept <> enfants <> ont <> raconté <> une <> histoire <> amusante
P	lə <> sjɛl <> ɛ <> blø <> e <> la <> mɛʁ <> osi le <> sɛt <> ãfɑ̃ <> õ <> ʁakõte <> yn <> istwaʁ <> amyzãt
S	lə . sjɛ **.** **l**ɛ . blø . e . la . mɛ **.** **ʁ**o . si le . sɛ **.** **t**ã . fã . õ . ʁa . kõ . te . **y** **.** **n**i . stwa **.** **ʁ**a . my . zãt

Figure 6: *Input string differences with syllable boundaries. '<>' denote word boundaries, '.' denote syllable boundaries. The boundaries in the S system cross the word boundaries between 'ciel-est', 'mer-aussi', 'sept-enfants' and 'histoire-amusante'.*

6. Acknowledgements

This work was supported by an ESRC doctoral training grant provided via the SGSSS.

7. References

[1] Y. Wang *et al.*, "Tacotron: Towards end-to-end speech synthesis," in *Proc. of Interspeech*, 2017, pp. 4006–4010.

[2] J. Shen *et al.*, "Natural TTS synthesis by conditioning wavenet on mel spectrogram predictions," in *Proc. ICASSP*, 2018, pp. 4779–4783.

[3] W. Ping *et al.*, "Deep Voice 3: Scaling Text-to-Speech with Convolutional Sequence Learning," in *Proc. ICLR*, 2018.

[4] J. Taylor and K. Richmond, "Analysis of Pronunciation Learning in End-to-End Speech Synthesis," in *Proc. Interspeech*, 2019, pp. 2070–2074.

[5] K. Ito, "The LJ speech dataset," 2017, available: https://keithito.com/LJ-Speech-Dataset/.

[6] C. Veaux, J. Yamagishi, and K. MacDonald, "VCTK Corpus: English Multi-speaker Corpus," 2019. [Online]. Available: https://datashare.is.ed.ac.uk/handle/10283/2651

[7] A. Perquin, E. Cooper, and J. Yamagishi, "An Investigation of the Relation Between Grapheme Embeddings and Pronunciation for Tacotron-based Systems," in *Submission to Interspeech*, 2021.

[8] H. Zhang *et al.*, "Neural models of text normalization for speech applications," *Comput. Linguist.*, vol. 45, no. 2, pp. 293–337, Jun. 2019.

[9] R. Sproat and N. Jaitly, "RNN approaches to text normalization: A challenge," 2017. [Online]. Available: https://arxiv.org/abs/1611.00068

[10] R. Sproat and K. Gorman, "A brief summary of the Kaggle text normalization challenge," in *Medium Blog Post*, 2018. [Online]. Available: https://medium.com/kaggle-blog/a-brief-summary-of-the-kaggle-text-normalization-challenge-11\\797b7e696f

[11] S. Ritchie *et al.*, "Unified Verbalization for Speech Recognition & Synthesis Across Languages," in *Proc. Interspeech*, 2019, pp. 3530–3534.

[12] S. Ritchie *et al.*, "Data driven parametric text normalization: Rapidly scaling finite-state transduction verbalizers to new languages," in *Proc SLTU and CCURL*, 2020, pp. 218–225.

[13] J. Fong *et al.*, "Investigating the Robustness of Sequence-to-Sequence Text-to-Speech Models to Imperfectly-Transcribed Training Data," in *Proc. Interspeech*, 2019, pp. 1546–1550.

[14] J. Taylor and K. Richmond, "Enhancing Sequence-to-Sequence Text-to-Speech with Morphology," in *Proc. Interspeech*, 2020, pp. 1738–1742.

[15] J. Fong, J. Taylor, and S. King, "Testing the Limits of Representation Mixing for Pronunciation Correction in End-to-End Speech Synthesis," in *Proc. Interspeech*, 2020, pp. 4019–4023.

[16] K. Kastner, J. F. Santos, Y. Bengio, and A. Courville, "Representation mixing for TTS synthesis," in *Proc. ICASSP*, 2019, pp. 5906–5910.

[17] T. Fujimoto *et al.*, "Impacts of input linguistic feature representation on Japanese end-to-end speech synthesis," in *Proc. SSW*, 2019, pp. 166–171.

[18] Y. Lu, M. Dong, and Y. Chen, "Implementing prosodic phrasing in chinese end-to-end speech synthesis," in *Proc. ICASSP*, 2019, pp. 7050–7054.

[19] Y. Yasuda, X. Wang, and J. Yamagishi, "Investigation of learning abilities on linguistic features in sequence-to-sequence text-to-speech synthesis," *Computer Speech & Language*, vol. 67, p. 101183, May 2021.

[20] M. Aso, S. Takamichi, and H. Saruwatari, "End-to-End Text-to-Speech Synthesis with Unaligned Multiple Language Units Based on Attention," in *Proc. Interspeech*, 2020, pp. 4009–4013.

[21] M. Elyasi and G. Bharaj, "Flavored tacotron: Conditional learning for prosodic-linguistic features," 2021. [Online]. Available: https://arxiv.org/abs/2104.04050

[22] S. Tyagi *et al.*, "Dynamic Prosody Generation for Speech Synthesis Using Linguistics-Driven Acoustic Embedding Selection," in *Proc. Interspeech*, 2020, pp. 4407–4411.

[23] H. Guo, F. K. Soong, L. He, and L. Xie, "Exploiting Syntactic Features in a Parsed Tree to Improve End-to-End TTS," in *Proc. Interspeech*, 2019, pp. 4460–4464.

[24] T. Kenter, M. Sharma, and R. Clark, "Improving the Prosody of RNN-Based English Text-To-Speech Synthesis by Incorporating a BERT Model," in *Proc. Interspeech 2020*, 2020, pp. 4412–4416.

[25] L. Zhao, J. Yang, and Q. Qin, "Enhancing prosodic features by adopting pre-trained language model in bahasa indonesia speech synthesis," in *Proc. ACAI*, 2020.

[26] T. Kenter *et al.*, "CHiVE: Varying prosody in speech synthesis with a linguistically driven dynamic hierarchical conditional variational network," in *Proc. ICML*, vol. 97, 2019, pp. 3331–3340.

[27] S. Ronanki, O. Watts, and S. King, "A hierarchical encoder-decoder model for statistical parametric speech synthesis," in *Proc. Interspeech 2017*, 2017, pp. 1133–1137.

[28] M. S. Ribeiro, O. Watts, and J. Yamagishi, "Parallel and cascaded deep neural networks for text-to-speech synthesis," in *Proc. SSW*, 2016, pp. 100–105.

[29] J. Chevelu *et al.*, "How to compare TTS systems: A new subjective evaluation methodology focused on differences," in *Proc. of Interspeech*, 2015, pp. 3481–3485.

[30] J. Pontes and S. Furui, "Predicting the phonetic realizations of word-final consonants in context – A challenge for french grapheme-to-phoneme converters," *Speech Communication*, vol. 52, no. 10, pp. 847–862, 2010.

[31] A. Greefhorst and A. Bosch, "Predicting liaison: An example-based approach," *Traitement Automatique des Langues*, vol. 57, pp. 13–32, Jan. 2016.

[32] J. Durand and C. Lyche, "French liaison in the light of corpus data," *Journal of French Language Studies*, vol. 18, no. 1, pp. 33–66, 2008.

[33] D. van Esch, M. Chua, and K. Rao, "Predicting pronunciations with syllabification and stress with recurrent neural networks," in *Proc. Interspeech*, 2016, pp. 2841–2845.

[34] Fatchord, "Tacotron implementation," 2020, available: https://github.com/fatchord/WaveRNN.

[35] N. Kalchbrenner *et al.*, "Efficient neural audio synthesis," in *Proc. ICML*, vol. 80, 2018, pp. 2410–2419.

[36] I. Steiner and S. L. Maguer, "Creating New Language and Voice Components for the Updated MaryTTS Text-to-Speech Synthesis Platform," in *Proc. LREC*, May 2018.

[37] B. New *et al.*, "Lexique 2: A new french lexical database," *Behavior Research Methods, Instruments, & Computers*, vol. 36, no. 3, pp. 516–524, 2004.

[38] F. Béchet, "Lia phon: un systeme complet de phonétisation de textes," *Traitement automatique des langues*, vol. 42, no. 1, pp. 47–67, 2001.

[39] K. J. M., *Guide de prononciation française pour apprenants finnophones*. University of Jyväskylä, 2018. [Online]. Available: http://research.jyu.fi/phonfr/20.html

[40] K. Toutanova *et al.*, "Feature-rich part-of-speech tagging with a cyclic dependency network," in *Proc. NAACL*, 2003, pp. 173–180.

[41] G. Klein *et al.*, "OpenNMT: Open-source toolkit for neural machine translation," in *Proc. ACL*, 2017, pp. 67–72.

11th ISCA Speech Synthesis Workshop (SSW 11)
26-28 August 2021, Budapest, Hungary

FeatherTTS: Robust and Efficient attention based Neural TTS

Qiao Tian[1], *Chao Liu*[2*], *Zewang Zhang*[1], *Heng Lu*[1], *Linghui Chen*[1]
Bin Wei[3], *Pujiang He*[3], *Shan Liu*[1]

[1]Tencent
[2]Harbin Institute of Technology(Shenzhen)
[3]Intel Corporation

{briantian, zewangzhang, bearlu, nedchen}@tencent.com

Abstract

Attention based neural TTS is elegant speech synthesis pipeline and has shown a powerful ability to generate natural speech. However, it is still not robust enough to meet the stability requirements for industrial products. Besides, it suffers from slow inference speed owning to the autoregressive generation process. In this work, we propose FeatherTTS, a robust and efficient attention-based neural TTS system. Firstly, we propose a novel Gaussian attention which utilizes interpretability of Gaussian attention and the strict monotonic property in TTS. By this method, we replace the commonly used stop token prediction architecture with attentive stop prediction. Secondly, we apply block sparsity on the autoregressive decoder to speed up speech synthesis. The experimental results show that our proposed FeatherTTS not only nearly eliminates the problem of word skipping, repeating in particularly hard texts and keep the naturalness of generated speech, but also speeds up acoustic feature generation by 3.5 times over Tacotron. Overall, the proposed FeatherTTS can be 35x faster than real-time on a single CPU.

Index Terms: acoustic model, attention, text-to-speech

1. Introduction

In recent years, with the rapid development of deep learning, neural text-to-speech (TTS) can synthesize speech which is more natural and expressive than traditional TTS pipeline. Neural TTS is usually divided into two parts: an acoustic model and a neural vocoder. First, the input text (phoneme) sequence is converted into an intermediate acoustic feature sequence(linear spectrogram or mel-spectrogram) through an acoustic model such as Tacotron [1], Tacotron2 [2], Transformer TTS [3], FastSpeech [4], etc. Then, the Griffin-Lim algorithm [5] or neural vocoder such as WaveNet [6] and WaveRNN [7] is used to generate the final waveform according to the acoustic features. Sequence-to-sequence models with an attention mechanism are currently the predominant paradigm in neural acoustic model and have shown a powerful ability to generate expressive and high-quality speech. Those models learn the alignment between text sequence and frame-level acoustic features through the attention mechanism, and then predict spectral features that contain information such as pronunciation and prosody. The speech quality synthesized by the neural TTS is limited by the alignment generated by the attention mechanism. Although attention-based neural TTS has achieved great success, it is difficult to deploy in the industry due to its accidental alignment errors.

Tacotron [1] with content-based attention mechanism does not take into account the monotonicity and locality of TTS alignment, an improved hybrid location-sensitive attention (LSA) mechanism proposed in Tacotron2 [2] combines content-based and location-based features to achieve the synthesis of longer utterances. However, such hybrid mechanism also causes alignment issues occasionally. The LSA mechanism is borrowed from neural machine translation (NMT) and is not completely applicable TTS. Because the pronunciation is monotonous, for TTS, the alignment process is required to monotonous forward. For machine translation, the alignment process is not necessarily monotonous, It is possible that the last word of the target language corresponds to the first word of the source language. Therefore, many studies have adopted many techniques in the attention mechanism to ensure monotonicity. Such as [8] proposed the forward attention, which only considers the alignment paths that satisfy the monotonic condition at each decoder time step. And [8] further proposed a transition agent for monotonous attention, which achieves faster convergence speed and higher stability. [9] proposed a guided attention loss, which adds the prior knowledge of alignment monotinicity to the training process to help TTS models converge faster. Even, many researches use hard alignment based on duration expansion instead of attention mechanism, such as FastSpeech [4], DurIAN [10]. This type model usually requires an auxiliary model to help training.

Recently, inspired by the purely location-based GMM attention mechanism [11], an improved location-based GMM attention mechanism called GMMv2b is proposed in Google's work [12], which shows that the GMMv2b-based mechanism is able to generalize to long utterances, and can also improve speed and consistency of alignment during training. However, such GMM attention is unnormalized and not strictly monotonic, which leads to unstable performance. In addition, the commonly used stop token architecture in Tacotron often causes early stop phenomenon for complex texts and long sentences.

In this paper, we propose a novel attention-based neural TTS model named FeatherTTS, which can perform stable, fast and high-quality synthesis. Our major contributions are as follows: (1) We introduce the Gaussian attention for acoustic modeling, a monotonic, normalized and stable attention mechanism, which is very interpretable for end to end speech synthesis. (2) To solve the stop early issue, we remove the widely adopted stop token architecture in Tacotron2 and propose the attentive stop loss (ATL), which can determine whether to stop directly based on alignment and fast convergence for Gaussian attention. (3) To improve the inference speed and reduce the number of parameters without sacrificing the speech quality, we propose to adopt block sparse strategy to prune the weights of decoder .

*This work was done during internship in Tencent.

10.21437/SSW.2021-35

2. Related work

2.1. Hybrid attention based Tacotron2

Sequence-to-Sequence models with an attention mechanism are currently the predominant paradigm in neural TTS. Attention-based neural TTS such as Tacotron2 [2] generally uses an encoder to encode input sequence $x_{1:J}$ into hidden representation $h_{1:J}$ as

$$\{\boldsymbol{h}_{1:J}\} = Encoder(\{\boldsymbol{x}_{1:J}\}), \tag{1}$$

where J is the length of input phoneme sequence. Then, the attention RNN generates a state vector s_i, which is used as the query vector of the attention mechanism to generate alignment α_i at decode time i. According to the alignment α_i, a weighted average of the encoder output is calculated, which is the context vector c_i.

$$s_i = RNN_{Att}(s_{i-1}, c_{i-1}, y_{i-1}) \tag{2}$$

$$\alpha_i = Attention(s_i, ...) \qquad c_i = \sum_i \alpha_{i,j} h_j \tag{3}$$

Finally, the context vector c_i is fed into the decoder, and the final acoustic feature sequence $y_{1:T}$ is computed through post-net as

$$d_i = RNN_{Dec}(d_{i-1}, c_i, s_i) \qquad y_i = f_o(d_i), \tag{4}$$

where T is the length of output mel-spectrogram sequence.

Recently, many works have proposed various attention mechanism. Such as Tacotron [1] uses the purely content-based attention mechanism introduced in [13], Tacotron2 [2] uses an improved hybrid location-sensitive mechanism introduced in [14], some works [8, 15, 16] explore the use of monotonic attention mechanisms, and some authors [17, 18] use the location-based GMM attention.

2.2. Location based GMMv2b

Recently, Google's work [12] proposed a modified location-based attention mechanism which is called GMMv2b, has achieved great success. The GMMv2b mechanism is inspired by the location-based GMM attention mechanism introduced in [11]. The GMMv2b attention mechanism uses K Gaussian components to compute the alignment α_i as (5), where $\alpha_{i,j}$ is the weight of j-th element of encoder outputs, K is the number of Gaussian kernels, $\omega_{i,k}$ is the weight of k-th Gaussian component and $\mu_{i,k}, \sigma_{i,k}$ is the mean and standard deviation of k-th Gaussian component at decoding time i, respectively. The mean of each Gaussian component is computed following the recurrence relation in (6). The monotonicity of GMM attention is guaranteed by making Δ_i non-negative.

$$\alpha_{i,j} = \sum_{k=1}^{K} \frac{\omega_{i,k}}{Z_{i,k}} \exp\left(-\frac{(j-\mu_{i,k})^2}{2(\sigma_{i,k})^2}\right) \tag{5}$$

$$\mu_i = \mu_{i-1} + \Delta_i. \tag{6}$$

GMM attention usually calculates the intermediate variables ($\hat{\omega}_i$, $\hat{\Delta}_i$, $\hat{\sigma}_i$) first, and then uses the exponential function to obtain the final variables. In order to stabilize GMM attention, GMMv2b-based attention uses the softmax and the softplus functions to compute the final mixture parameters as

$$\begin{cases} Z_i = \sqrt{2\pi\sigma_i^2}, \\ \omega_i = S_{max}(\hat{\omega}_i), \\ \Delta_i = S_+(\hat{\Delta}_i), \\ \sigma_i = S_+(\hat{\sigma}_i), \end{cases} \tag{7}$$

where S_{max} and S_+ are the softmax function and the softplus function respectively. Besides, GMMv2b-based attention adds initial biases to the the intermediate parameters $\hat{\Delta}_i$ and $\hat{\sigma}_i$, which can encourage the final parameters to take on useful values at initialization.

As shown in [12], the GMMv2b-based mechanism is able to generalize to long utterances and maintains good naturalness, which makes the synthesis of the entire paragraph possible.

3. The proposed method

Although the GMMv2b-based mechanism has good performance, it also has many problems. First, this model still use stop token architecture which can lead to early stop. Second, GMM attention isn't completely monotonic because it uses a mixture of distributions with infinite support. Finally, GMMv2b attention is unnormalized because the attention weights are sampled from a continuous probability density function, this can lead to occasional spikes or dropouts in the alignment. Especially, there are repetition problems for the synthesis of short sentences, such as monophone and vowel. Therefore, we propose FeatherTTS, a more robust attention-based acoustic model, as shown in Fig. 1. Our model is based on the Tacotron2 [2] architecture and consists of a CBHG encoder, Gaussian attention and a block sparse decoder.

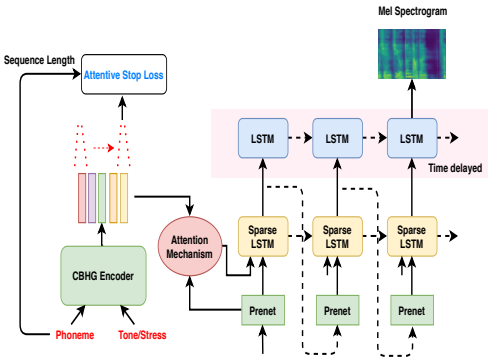


Figure 1: *The architecture of FeatherTTS*

3.1. Gaussian attention

In order to solve the incomplete monotonic and unnormalized problem in GMM attention, we propose to use Gaussian attention mechanism to model alignment, as shown in (8). Unlike the k Gaussian components used in GMM attention, we only use a single Gaussian function to calculate the alignment $\alpha_{i,j}$. Since the Gaussian function is naturally normalized, as long as the mean value of Gaussian attention at each decoding time step is monotonously forward, the monotonicity of the alignment can be guaranteed. We also calculate the intermediate variables ($\hat{\sigma}_i$, $\hat{\Delta}_i$) first, and then get the final parameters(σ_i, Δ_i) through the softplus function simlir to GMM attention.

$$\alpha_{i,j} = \exp\left(-\frac{(j-\mu_i)^2}{2(\sigma_i)^2}\right) \tag{8}$$

$$\mu_i = \mu_{i-1} + \Delta_i \tag{9}$$

We use such simple and normalized Gaussian attention function to calculate the alignment between the input phoneme sequences and the spectrogram frames. The mean μ_i and the variance $(\sigma_i)^2$ of the Gaussian attention mechanism control the position and width of the attention window, respectively. Δ_i is non-negative, so the mean μ_i is monotonically increasing, which guarantees the alignment process of the Gaussian attention mechanism is completely monotonic.

3.2. Attentive stop loss

The stop token architecture used in Tacotron2 [2] will cause stop early problems. Compared with the GMM attention of K Gaussian components, the single Gaussian attention mechanism has a weaker fitting ability and it will difficult to converge. Therefore, we need to add constraints to ensure that it can be aligned to the end of the input sequence at the end of decoding. In order to solve the above problems, we remove the stop token architecture, and propose the attentive stop loss, which directly judges the stop based on alignment. It is calculated as

$$L_{stop} = |\mu_T - (J+1)|, \tag{10}$$

where μ_T is the mean value of Gaussian attention function at last step, and J is the length of input phoneme sequence.

During training, the attentive stop loss forces the mean μ_i of Gaussian attention to go forward to the end of the phoneme sequence to ensure accurate alignment. In the inference stage, FeatherTTS will stop to predict when $\mu_i \geq (J+1)$.

3.3. Sparse autoregressive decoder

It has been demonstrated that, with the same computational complexity, a larger sparse network behaves better than a smaller dense network [7, 19]. In this work, to reduce the amount of computation of LSTM layers in decoder without a significant loss in quality, we reduce the number of non-zero values in each LSTM kernel weight. Inspired by [20, 21], we adopt the weight pruning scheme based on the weight magnitude.

We start to perform weight pruning after 20K steps and every 500 steps, we sort the weights of sparsified LSTM layers and zero out certain number of weights with the smallest magnitudes until the target sparsity 90% is reached at 200K step. After block sparsity, the number of main operations in every sparsified LSTM layer is

$$C = 4(1 - S)(I * H + H^2), \tag{11}$$

where I and H are the dimensions of input and hidden state of the LSTM cell, respectively, and S is the target sparsity.

In FeatherTTS, we used the time-delayed post-net as in [22], which is a vanilla LSTM layer with 256 units. Overall, FeatherTTS is trained to minimize the total loss as

$$Loss = \frac{1}{T}\sum_{i=1}^{T}|y'_i - y_i| + \frac{1}{T-d}\sum_{i=1}^{T-d}|y''_{i+d} - y_i| + \lambda L_{stop}, \tag{12}$$

where d is the number of frames of time delay and λ is a scaling factor. On the right hand side of Eq. 12, the first two items of the loss function are L1 loss between reference mel-spectrogram y_i and the predicted both before and after mel-spectrogram y'_i, y''_i. The last item is the attentive stop loss.

Table 1: *Mean Opinion Score (MOS) with* 95% *confidence intervals for different models.*

Model	MOS on speech quality
Tacotron2(GMMv2b)	4.31 ± 0.03
FeatherTTS w/o Block sparsity	4.32 ± 0.04
FeatherTTS	$\mathbf{4.33 \pm 0.04}$

4. Experiments

4.1. Data Set

We used a corpus containing 20 hours of Mandarin recordings by a professional broadcaster for all experiments. The corpus was split into a training set of approximately 18 hours and a test set of 2 hours. All the recordings were down-sampled to 24KHz sampling rate with 16-bit format. We used 80-band mel-scale spectrogram as training target, and then the mel-scale spectrogram was converted into waveforms by FeatherWave neural vocoder [23].

4.2. Experimental Setup

For comparison, we implemented two models including GMMv2b-based Tacotron2 [12] and FeatherTTS. As the baseline model, the GMMv2b-based model is composed of five mixture components. In order to reduce the model size, training and inference time, two consecutive frames were predicted at each decoding time step. For FeatherTTS, we delayed 5 frames and the rate of attentive stop loss λ was set to 0.001. All models were trained 300k steps with batch size 32 on a single GPU. Other experimental setups are the same as AdaDurIAN [22] if not specified.

4.3. Evaluations

In this section, we evaluated the proposed FeatherTTS and Tacotron2 (GMMv2b) [12] in term of naturalness and robustness, and compared the synthesis speed of the above two models with FastSpeech [4].

4.3.1. Mean Opinion Score

We used the Mean Opinion Score (MOS) to measure the naturalness of the synthesized speech[1]. Through crowdsourcing, we conducted the MOS evaluation on 20 synthesized audios which are unseen during training.The results of subjective MOS evaluation are presented in Table 1. The results show that, under the same vocoder configuration, both FeatherTTS and Tacotron2(GMMv2b) have similar MOS values. In addition, we compared the effect of block sparsity on the sound quality. It can be seen from the experimental results that FeatherTTS with block sparsity outperforms FeatherTTS without block sparsity with a gap of 0.01 in MOS, which is basically in line with our expectations.

4.3.2. Word Error Rate

FeatherTTS is designed to keep the naturalness as Tacotron2(GMMv2b) while avoiding the mispronunciations observed in the Tacotron2(GMMv2b). Wrod error rate (WER) is a general indicator for evaluating ASR and NMT

[1]Part of synthesized samples could be found at this URL: `https://wavecoder.github.io/FeatherTTS/`

Table 2: *The Word Error Rate (WER) for different models.*

Model	Word error rate
Tacotron2(GMMv2b)	4.1%
FeatherTTS	**0.9%**

Table 3: *The inference speed of different models.*

Model	Speed
FastSpeech	13.3x
Tacotron2(GMMv2b)	10.4x
FeatherTTS	**35.0x**
FeatherTTS BF16	**60.0x**

systems, and it can be used in TTS to measure the robustness of TTS synthesized speech. Therefore, we compared the robustness of two systems in terms of generated speech. We used manual listening and checking methods to perform fine-grained error checks on the synthesized speeches, such as pronunciation errors, word skipping, repeating, etc. The synthesized sentences are from different fields and are very hard for TTS, such as website links, alphanumeric combination, etc. There are a total of 50 test sentences and 10 participants, and each sentence is checked by at least 5 different participants. The final experiment results as shown in Table 2. We can see that Tacotron2(GMMv2b) has an error rate of 4.1%, while FeatherTTS is more robust, with an error rate of only 0.9%. This strongly proves the role of Gaussian attention and attentive stop loss in improving model stability.

4.3.3. Synthesis Speed

In this experiment, we proved the effectiveness of the block sparse decoder for accelerating training and inference. We compared the real-time rate of FastSpeech, Tacotron2(GMMv2b) and FeatherTTS to generate mel-spectrograms on a single core CPU(Intel Xeon Platinum 8255C). The results of synthesis speed are presented in Table 3. Tacotron2(GMMv2b) can achieve an inference speed of 10.4 times faster than real time, while FeatherTTS can further be accelerated by 3.5 times over Tacotron2(GMMv2b). In addition, compared with non-autoregressive FastSpeech, FeatherTTS is also about 2.6 times faster . Furthermore, we truncated the parameters and ran them on the BF16 [24, 25] format to reduce the memory consumption, and finally achieve 60 times faster than real-time on a single CPU core (Cooper Lake, 3rd Gen Intel Xeon Scalable processors). The above experiments prove the accelerating performance of the proposed methods for inference, and makes it possible to deploy TTS on edge devices.

5. Conclusions

In this work, we proposed FeatherTTS, an improved neural TTS system with Gaussian attention, attentive stop loss and block sparse decoder. Experiments demonstrate that such attention mechanism is very efficient and would greatly improve robustness of attention-based neural TTS system. With block sparse decoder, our proposed FeatherTTS can speed up the mel-spectrogram generation by 3.5 times faster than Tacotron2 nearly without any performance degradation. The ideas introduced in FeatherTTS pave a new way for both efficient and robust speech synthesis, and could be also applied to other sequence-to-sequence task including automatic speech recognition.

For future work, we will continue to investigate the performance of FeatherTTS on edge-devices.

6. Acknowledgments

The authors would like to thank Yi Xie in IAGS, Intel Asia-Pacific Research & Development Co Ltd.. This member in Intel helped to optimized our algorithm with AVX512 and BF16 intrinsics to get good performance on the 3rd Gen Intel Xeon Scalable processors.

7. References

[1] Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio *et al.*, "Tacotron: Towards end-to-end speech synthesis," *arXiv preprint arXiv:1703.10135*, 2017.

[2] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan *et al.*, "Natural tts synthesis by conditioning wavenet on mel spectrogram predictions," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4779–4783.

[3] N. Li, S. Liu, Y. Liu, S. Zhao, and M. Liu, "Neural speech synthesis with transformer network," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 6706–6713.

[4] Y. Ren, Y. Ruan, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, "Fastspeech: Fast, robust and controllable text to speech," in *Advances in Neural Information Processing Systems*, 2019, pp. 3171–3180.

[5] D. Griffin and J. Lim, "Signal estimation from modified short-time fourier transform," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 32, no. 2, pp. 236–243, 1984.

[6] A. Van Den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio." in *SSW*, 2016, p. 125.

[7] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," *arXiv preprint arXiv:1508.04025*, 2015.

[8] J.-X. Zhang, Z.-H. Ling, and L.-R. Dai, "Forward attention in sequence-to-sequence acoustic modeling for speech synthesis," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4789–4793.

[9] H. Tachibana, K. Uenoyama, and S. Aihara, "Efficiently trainable text-to-speech system based on deep convolutional networks with guided attention," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4784–4788.

[10] C. Yu, H. Lu, N. Hu, M. Yu, C. Weng, K. Xu, P. Liu, D. Tuo, S. Kang, G. Lei *et al.*, "Durian: Duration informed attention network for multimodal synthesis," *arXiv preprint arXiv:1909.01700*, 2019.

[11] A. Graves, "Generating sequences with recurrent neural networks," *arXiv preprint arXiv:1308.0850*, 2013.

[12] E. Battenberg, R. Skerry-Ryan, S. Mariooryad, D. Stanton, D. Kao, M. Shannon, and T. Bagby, "Location-relative attention mechanisms for robust long-form speech synthesis," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6194–6198.

[13] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[14] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *Advances in neural information processing systems*, 2015, pp. 577–585.

[15] C. Raffel, M.-T. Luong, P. J. Liu, R. J. Weiss, and D. Eck, "Online and linear-time attention by enforcing monotonic alignments," *arXiv preprint arXiv:1704.00784*, 2017.

[16] M. He, Y. Deng, and L. He, "Robust sequence-to-sequence acoustic modeling with stepwise monotonic attention for neural tts," *arXiv preprint arXiv:1906.00672*, 2019.

[17] K. Kastner, J. F. Santos, Y. Bengio, and A. Courville, "Representation mixing for tts synthesis," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 5906–5910.

[18] E. Battenberg, S. Mariooryad, D. Stanton, R. Skerry-Ryan, M. Shannon, D. Kao, and T. Bagby, "Effective use of variational embedding capacity in expressive end-to-end speech synthesis," *arXiv preprint arXiv:1906.03402*, 2019.

[19] J.-M. Valin and J. Skoglund, "Lpcnet: Improving neural speech synthesis through linear prediction," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 5891–5895.

[20] S. Narang, E. Undersander, and G. Diamos, "Block-sparse recurrent neural networks," *arXiv preprint arXiv:1711.02782*, 2017.

[21] S. Narang, E. Elsen, G. Diamos, and S. Sengupta, "Exploring sparsity in recurrent neural networks," *arXiv preprint arXiv:1704.05119*, 2017.

[22] Z. Zhang, Q. Tian, H. Lu, L.-H. Chen, and S. Liu, "Adadurian: Few-shot adaptation for neural text-to-speech with durian," *arXiv preprint arXiv:2005.05642*, 2020.

[23] Q. Tian, Z. Zhang, H. Lu, L.-H. Chen, and S. Liu, "Featherwave: An efficient high-fidelity neural vocoder with multi-band linear prediction," *arXiv preprint arXiv:2005.05551*, 2020.

[24] P. Teich, "Tearing apart google's tpu 3.0 ai coprocessor," *Retrieved June*, vol. 12, p. 2018, 2018.

[25] S. Wang and P. Kanwar, "Bfloat16: the secret to high performance on cloud tpus," *Google Cloud Blog*, 2019.

11th ISCA Speech Synthesis Workshop (SSW 11)
26-28 August 2021, Budapest, Hungary

Comparing acoustic and textual representations of previous linguistic context for improving Text-to-Speech

Pilar Oplustil-Gallegos, Johannah O'Mahony, Simon King

The Centre for Speech Technology Research, University of Edinburgh, United Kingdom
p.s.oplustil-gallegos@sms.ed.ac.uk

Abstract

Text alone does not contain sufficient information to predict the spoken form. Using additional information, such as the linguistic context, should improve Text-to-Speech naturalness in general, and prosody in particular. Most recent research on using context is limited to using textual features of adjacent utterances, extracted with large pre-trained language models such as BERT.

In this paper, we compare multiple representations of linguistic context by conditioning a Text-to-Speech model on features of the preceding utterance. We experiment with three design choices: (1) acoustic vs. textual representations; (2) features extracted with large pre-trained models vs. features learnt jointly during training; and (3) representing context at the utterance level vs. word level.

Our results show that appropriate representations of either text or acoustic context alone yield significantly better naturalness than a baseline that does not use context. Combining an utterance-level acoustic representation with a word-level textual representation gave the best results overall.

Index Terms: Text-to-Speech, speech synthesis, context, prosody

1. Introduction and Related Work

Although text alone is not sufficient to predict prosody accurately, Text-to-Speech (TTS) systems are generally trained to generate spoken utterances given textual input only, and utterances are assumed to be independent from one another. While this might be true for certain types of text, utterances in monologues, conversation, audio-books or from any other long-form discourse are not isolated, but influenced by context [1, 2, 3]. Utterances are organized into a discourse structure in which neighbouring utterances are part of the linguistic context [4]. Context can have a global effect on the average and range of F_0 and speech rate, or a localized one such as the absence or presence of prominence.

In this paper we study how linguistic context, specifically the previous utterance, can be exploited to improve TTS. Our proposed method conditions the generation of an utterance on the acoustic and/or textual properties of the immediately preceding one. We experiment with different design choices to answer the general research question: how should linguistic context be represented?

Augmenting TTS model inputs with linguistic context information has been proposed by several authors, including the use of position of sentence inside a larger unit such as a paragraph [1, 5], explicit discourse features such as discourse relations [6] or topic structure [7]. While discourse features can improve synthetic speech, feature extraction relies on models that require supervised training on appropriately-labelled data.

Other approaches include directly labelling emphasis [8, 9, 10] or phrase breaks [11, 12]. Direct labelling can be useful for controllability, but accurately predicting labels only from text is hard.

In order to avoid the need for labelled data, unsupervised approaches can be used to learn contextual representations from acoustic features using encoders, which are later driven by traditional textual features [13, 14, 15]. However, these models still generally use within-sentence textual input features, which are insufficient to accurately predict prosody. [16] takes a different approach by using linguistic features and acoustic distance from the previous utterance to sample from a variational autoencoder of prosody, which synthesizes the current sentence. However, their method is applied at inference time only.

Another approach, closer to what we propose here, uses textual context to enhance a TTS baseline, conditioning mel spectrogram prediction directly on a representation of context [17, 18] in which BERT-derived features represent neighbouring (both preceding and following) sentences. Although neither method uses explicit prosodic features or learns prosodic representations, it was observed that the use of context significantly improves the prosody of the synthesized speech.

How the different features of context are captured is an important design choice. While [17] and [18] only capture textual features, in previous work [19] we saw that acoustic features can also lead to significant improvement. That approach makes use of a prosody transfer module, Global Style Tokens [20], to extract a prosodic representation from the mel spectrogram of the context. That representation is then used to condition the model, in a similar fashion to [17] and [18].

Our previous work was limited to represent acoustic features of the context at the utterance level using mel spectrograms. Here, we substantially expand the scope of our work to consider additional design choices, and to compare against methods proposed by others.

Therefore, the current goal is to experiment with three design choices regarding how to represent context: (1) textual vs. acoustic features; (2) representations extracted with large pre-trained models vs. representations learnt jointly with the TTS training; and (3) context at the utterance-level or at the word-level.

We will show that: either textual or acoustic representations of context can significantly improve speech naturalness, and a combination of both yields the best results; representations extracted with large pre-trained models outperform representations extracted using jointly-trained model components; and, word-level representations seem to be better matched to textual features, while an utterance-level representation is better for acoustic features.

10.21437/SSW.2021-36

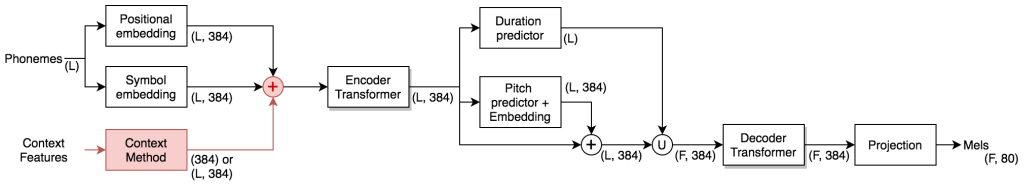


Figure 1: *System diagram. The baseline architecture is FastPitch [21] which we augment with a Context Method (in red) whose output is summed to the embeddings at the encoder input. The Context Method is one of the 4 possible models shown in Figures 2 and 3. L is the length of the current sentence (in phones). F the duration of the output (in frames). U denotes upsampling from phones to frames.*

2. Experimental design

2.1. Baseline

Our baseline model is FastPitch [21], which comprises Transformer-based encoder and decoder, with explicit duration and F_0 predictors. Input symbols (phonemes in the current work) and their positional encoding are embedded, summed, and input to the encoder. F_0 is embedded and summed to the encoder output before going into the decoder during training. The duration of each input symbol determines the upsampling between encoder output and decoder input. F_0 is modelled per-input symbol, with per-speaker mean/variance normalisation. During training, ground-truth values of F_0 and duration are used, whilst a predictor is trained for each of them. For inference, predicted values are used.

FastPitch was selected because it is fast and stable in both training and inference, and has an open source implementation from the original author [22]. All models in the current work were trained from scratch for ∼77k iterations. To vocode the generated mel spectrograms to waveforms, we used the WaveGlow [23] checkpoint included with the FastPitch implementation, which has been trained on the LJSpeech corpus [24].

To condition FastPitch on previous sentence context, we add a module that provides a representation that is summed to the encoder inputs, labelled as *Context Method* in Figure 1. This location was selected as the best place to inject context into the model in prototyping experiments.

2.2. Context features and representations

We compare acoustic vs. textual features, each of which can be input to either a large pre-trained model, or a model jointly trained with the TTS model, to create a Context Representation.

Acoustic: we use the same mel spectrograms extracted for training. As in FastPitch [21], these are 80-band mel spectrograms extracted with a window length of 1024 samples 256 hop size. For the **jointly-learnt** condition, a Context Representation is learnt from the mel spectrograms as described in Section 2.3.

For the **pre-trained** condition, the mel spectrogram is used to obtain a Context Representation from a large pre-trained model. We use the Deep Spectrum [25, 26], which was found in our previous work to be capable of encoding global acoustic characteristics [27]. It extracts a fixed-dimension vector by treating the mel spectrogram as an image and inputting it to a large-scale image classification model. We use the implementation from the original authors [28], using layer *fc2* of the VGG-19 model to obtain a 4096-dim vector. One vector can be obtained for the whole utterance, or for each of a sequence of fixed windows (which, in our experiments, will depend on the word-level or utterance-level condition, see Section 2.3).

Text: we use two types of features derived from the text: phonetic transcriptions for the jointly-trained condition and word tokens for the pre-trained one. To **jointly-learn** a Context Representation, we use the phonetic transcription of the previous sentence. Phonetic transcriptions are obtained as for all the training data for the models (Section 2.4), and use 47 symbols including phones and punctuation. Word or syllable boundaries are not included in the transcription.

To obtain a context representation from a **pre-trained** model, we use BERT, and therefore, the context features used as input correspond to text words (or tokens). BERT embeddings are extracted using an off-the-shelf model in the transformers Python library [29]. 768-dim vectors at the utterance-level are obtained by averaging the activations of second to last hidden layer, or at the word-level by summing the activations of the last four layers of the model [30].

We decided to use a phonetic transcription for the jointly-learnt condition rather than textual words or tokens as it seemed unlike that the Context Method would be able to learn a relationship over sparse combinations of words for our training data (which is why large models as BERT are required to encode such relationships).

2.3. Context methods

The third design choice we are interested in is whether to represent context at utterance- or word-level. We anticipate that the model will learn global prosodic effects from utterance-level representations, and local effects from word-level representations. The utterance-level representations are a fixed-length vector that is constant for every encoder step. In contrast, the word-level method outputs a representation that potentially varies for every encoder step.

Whilst it is desirable to maintain the most similar model architecture for all combinations of design choices, the differences in resolution and nature of the representations do entail some differences, illustrated in Figures 2 and 3. In both figures, Context Features are always extracted from the previous sentence. The resulting Processed Context Representation is the one finally added to the encoder inputs in Figure 1, conditioning the current sentence.

2.3.1. Using an utterance-level representation

Utterance-level Context Methods make use of Global Style Tokens [20] which, as we have already shown [19], can be used to represent context at the utterance-level and have been used in TTS for diverse tasks [31, 32, 33]. GSTs are a set of randomly initialized tokens (vectors). Multi-head attention is used to learn the relevance of each token for every training utterance. Since the tokens are constant, they can be thought of as labels,

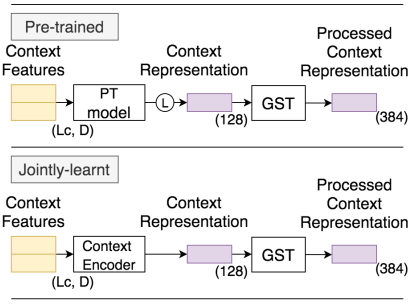


Figure 2: *Context Methods for extracting an utterance-level representation of context. As described in Section 2.2, for the pre-trained condition, Context Features correspond either to mel spectrograms (Lc = frames, D = 80) input to a Deep Spectrum pre-trained model, or to text word tokens (Lc = word tokens, D = embedding dim) input to BERT. Because pre-trained models output Context Representations with a different dimension, a linear layer (circle-L) is used to reduce dimensionality. For the jointly-learnt condition, mel spectrogram (Lc = frames, D = 80) or phonetic transcription (Lc = phones, D = embedding dim) are the Context Features. While pre-trained models (PT model) extract a single vector Context Representation already, for the jointly-learnt condition a single vector Context Representation is obtained through a Context Encoder. Finally, for both conditions, a Processed Context Representation is obtained by applying GST.*

with attention 'labelling' the data in unsupervised fashion.

GST takes as input a fixed-dimension vector. The representations obtained from pre-trained models (Deep Spectrum or BERT) can be obtained at the utterance-level, and therefore are simply reduced in dimensionality before GST. In contrast, the representations obtained from jointly-trained models must be summarised into a single vector. We use a Context Encoder (lower part of Figure 2) with the same architecture as the reference encoder in [20]. We train GST with 10 tokens and 8 heads to output a 384-dim vector. We use the implementation provided by [34].

2.3.2. *Using a word-level representation*

Figure 3 explains how the word-level Context Methods create a Context Representation from the previous sentence, for each word in the Current Sentence, which has the potential to encode local prosodic phenomena.

Pre-trained models output Context Representations at the word-level (or pseudo-word-level for Deep Spectrum) already. For the jointly-learnt condition, the Context Features are first processed by a block of convolutional layers with the same architecture as the transformer (1D conv $>$ Relu $>$ 1D conv $>$ summed to the residual $>$ layer norm). Then, word-level resolution is obtained by averaging frames or phones within word boundaries.

Once the Context Representation is obtained, attention is used to gather elements of it and potentially re-order them in a way that is relevant for the Current Sentence. Finally, the new Processed Context Representation is simply added to the encoder inputs without further processing.

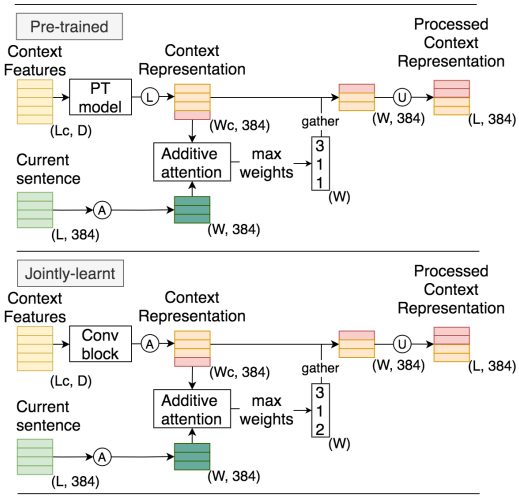
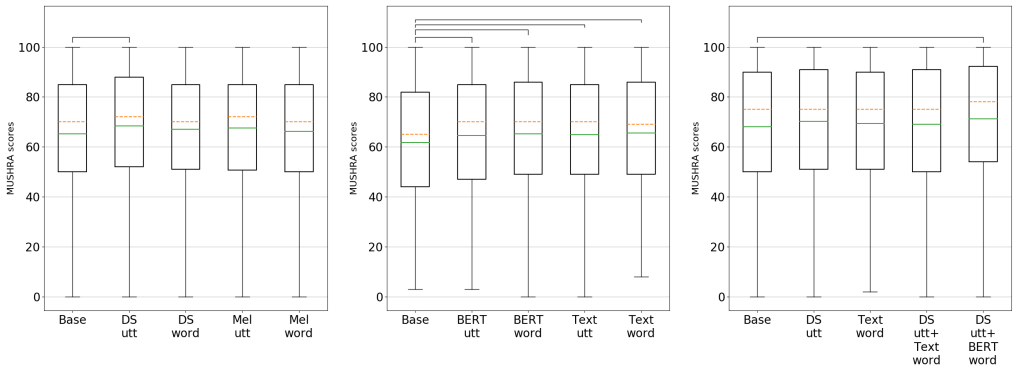


Figure 3: *Context Methods for extracting a word-level representation of context. Context Features correspond to those described in Figure 3. Context Representations are now obtained to match word-like resolution (Wc). For the pre-trained condition, BERT embeddings are obtained for every word token, while for Deep Spectrum, mel spectrograms are divided into one second segments (without overlap). As before, these are reduced in dimensionality by a linear layer (circle-L) to obtain the Context Representation. For the jointly-learnt condition, a block of convolutions is first applied to learn a Context Representation, however this module does not affect the resolution of the features (Lc = frames, for mel spectrograms, Lc = phones, for phonetic transcription). To obtain a word-level representation (Wc), we average (circle-A) using word boundaries. In parallel, word-level representations for the Current Sentence phones are obtained averaging. Next, the attention mechanism calculates how relevant each word in the Context Representation is to each word in the Current Sentence. The maximum attention weight for each word in the Current Sentence is used to identify the most relevant word in the Context Representation; the Context Representation of that word is gathered into a sequence of length W. The resulting Processed Context Representation is upsampled (circle-U) to match the length required to sum it to the encoder inputs.*

2.4. Data and pre-processing

All models used phonetized inputs obtained while force-aligning the data with the Montreal Forced Aligner [35] to extract the ground-truth durations required to train the duration predictor and upsample phones to frames, and to obtain the word boundary information needed for word-level representations. Out-of-vocabulary words were transcribed using G2P [36] and punctuation was restored. We obtained F_0 contours using Praat for Python [37] as in FastPitch [21].

We trained and tested all models using LJSpeech [24], with 12443 training sentences and 525 test sentences. We follow the data naming structure to obtain previous-current sentence pairs.



(a) (b) (c)

Figure 4: *Listening test results (a) acoustic context alone; (b) text context alone; (c) best models compared with acoustic+text combinations. Horizontal bars connect pairs of systems that are significantly different.*

Model name	**Context Feature**	**Context Representation**	**Context Method**
DS-utt	Acoustic	Deep Spectrum	Utterance
DS-word	Acoustic	Deep Spectrum	Word
mel-utt	Acoustic	Learnt from mels	Utterance
mel-word	Acoustic	Learnt from mels	Word
BERT-utt	Text	BERT	Utterance
BERT-word	Text	BERT	Word
Text-utt	Text	Learnt from phones	Utterance
Text-word	Text	Learnt from phones	Word

Table 1: *Summary of models compared in experiments.*

3. Evaluation and Results

Testing all combinations of our three design choices resulted in the 8 models summarised in Table 1. We predicted that text vs acoustic features, and utterance-level vs word-level representation, would be complementary, so we also tested some combinations. To make evaluation feasible, the listening test was conducted in three parts: (1) compare the 4 acoustic feature systems; (2) compare the 4 text feature systems; (3) compare the best acoustic system, best text system, and two systems that combine both.

We did not know whether acoustic or text context would be most informative. However, we did hypothesise that acoustic context would be best represented at the utterance level, and that text context would be best represented at the word level.

Each of the three listening tests used a MUSHRA-like design[1]. and compared 4 models, plus the baseline and the hidden reference (vocoded natural speech). The same 25 sentences were used for all listening tests. Each MUSHRA screen presented the reference audio, then the 6 samples to be rated, without text. Participants were instructed to rate the naturalness of the synthetic speech. For the acoustic systems, features were extracted from a natural rendering of the context utterance: Section 4 comments on the possible effects of using synthetic speech instead.

We implemented the test online using Qualtrics and recruited participants who self-identified as native speakers of English and US citizens, using Prolific Academic. Results from participants who rated any reference sample lower than 50, or were too fast to complete the task, were discarded. For each test, the first 20 participants who passed these checks were used to calculate the results. Each test used different participants.

Statistical significance was determined using the Wilcoxon signed-rank test with Bonferroni correction. Figure 4 shows the results for the three tests.

3.1. First listening test: acoustic context

Results for acoustic context are in Figure 4(a) for the systems listed in the upper 4 rows of Table 1. Only Deep Spectrum features at the utterance level were significantly better than baseline. Although not significant, all other acoustic contexts resulted in slightly higher scores than baseline, with utterance-level representation tending to be better than word-based.

3.2. Second listening test: text context

Results for the text context are in Figure 4(b) for the systems listed in the lower 4 rows of Table 1. All models using text context were significantly more natural than baseline. Although not significantly different between each other, word-level representation tended to lead to slightly higher naturalness than utterance-level.

3.3. Third listening test: best models and combinations

We compared the most effective way to use acoustic context (DS-utt), the most effective way to use text context (Text-word, which had the most significant difference to the baseline), and two combinations of acoustic and text context.

[1]Samples:
`https://pilarog.github.io/ssw2021/index.html`

We trained the combinations: DS-utt + Text-word and DS-utt + BERT-word. Deep Spectrum was clearly the most effective acoustic feature. Since there was no significant difference between the models using text context, we included both Text-word and BERT-word. Results are shown in Figure 4(c). The system using Deep Spectrum features to derive an utterance-level representation of acoustic context, with BERT features to derive a word-level representation of text context, was significantly better than baseline.

It is not surprising that neither DS-utt or Text-word were significantly better than baseline here, even though they were in the preceding listening tests. MUSHRA ratings are relative, with an element of ranking, so a different set of systems under comparison (especially a change in the least natural system; there is no anchor in our tests) will lead to a different rating space.

3.4. Listening test results analysis

Our results illustrate the benefit of using both acoustic and text features of the context utterance, individually or in combination. In every listening test, the baseline was outperformed by at least one model employing context. DS-utt + BERT-word was the best combined system, which supports our hypothesis that acoustic features are most useful when represented at the utterance level, with text features at the word level. Pre-trained models generally outperformed jointly-trained ones.

Informally, we observed that the use of context affected the speech in different ways: in prosody, pauses, and pronunciation, with the most apparent changes being prosodic in nature. Although we did not ask participants to directly judge prosody, it seems likely that they are implicitly doing so, given some of their comments. At the end of each listening test, we included an optional comment box. Several participants mentioned how it was "interesting" or "challenging" to distinguish the different "inflections" in the samples.

3.5. Qualitative analysis

Our results indicate that context is informative. To confirm this and to further analyse its effect, we examined differences in the output when synthesizing the same sentence with different contexts. This differs from what was evaluated in the listening tests of the previous section. Here, we confirm that changes in context produce changes in the output.

Although pronunciation can also be affected by the context, most of the variation we observed was prosodic. Figure 5 provides some example F_0 contours. In (a), using acoustic context represented at utterance level, the overall F_0 pattern tends to stay the same, and changing context has a global effect, shifting F_0 or affecting speech rate. In contrast, (b) shows that representing text context at the word level can modify the position and strength of prominence. Finally, combining acoustic and text context in (c) illustrates both effects.

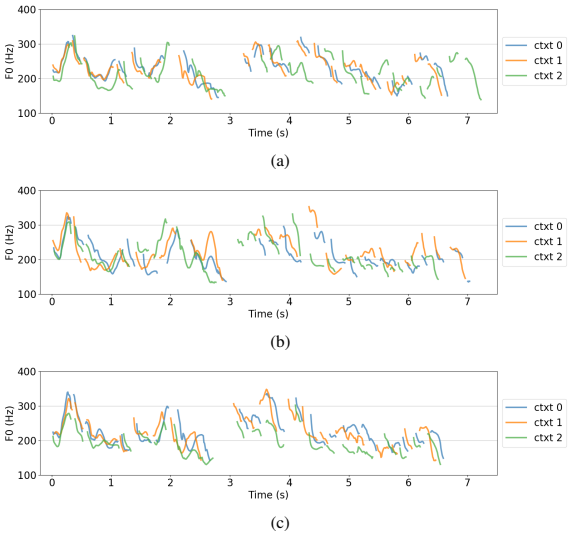


Figure 5: *Illustration of the effect of context for a single sentence synthesized with (a) DS-utt; (b) BERT-word; (c) a combination of both. In each plot, the three F_0 contours are the result of using three different context utterances (the same three across all plots).*

4. Conclusion and future work

Our results provide further evidence that additional context can improve TTS naturalness, and that the way in which context is represented matters. Even if context is not used explicitly to improve prosody, this seems to be the aspect that is affected the most.

We have shown that both acoustic and text context, when suitably represented, can significantly improve naturalness, and that the best results are obtained by combining them. In a real use-case (e.g., long-form synthesis), acoustic context would need to be extracted from the previous *synthesized* utterance. Although we did not test this condition here, we provide samples on the companion web page for DS-utt using features extract from synthesized speech context: degradation appears to be minimal. Text features have the notable advantage of being available for future context, although this was not tried here.

Our results indicate that features extracted using large pre-trained models are more effective than using jointly-trained models, especially for acoustic features. It could be that the acoustic relationships between context and current sentence is very sparse. In contrast, using text features with a jointly-trained model was comparable (in the second listening test) to BERT. Very recent work proposes using BERT on phonetic transcriptions [38], which would be worth trying.

To obtain the best results from a jointly-trained model for extracting a Context Representation, it might be necessary to incorporate an extra loss, as in our preliminary work [19]. We did not include this condition here as the focus was on how best to represent context rather than on the model itself.

There is also evidence that acoustic features give best results when represented at utterance level, and text features when represented at word level. The qualitative analysis in Section 5 suggests that these are associated with producing global and local prosodic effects respectively, without having to model these in an explicit way or through very specific features.

In future work, choice of data and speaker is important [27]. We aim to use more expressive or spontaneous data to better evaluate the effect of using context.

The listening test in this paper was restricted to measuring the naturalness of isolated sentences, which were not presented in context. This was a deliberate choice, but in-context eval-

uation will be a fundamental part of future work. Pioneering work [39] has tested such an evaluation paradigm, but we believe that it still needs to be further developed before we can apply it to our systems, and therefore we are also working on suitable evaluation methods for speech in context [40].

Acknowledgements: This work was supported in part by: ANID, Becas Chile, nº 72190135. This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement Nº 859588.

5. References

[1] M. Farrús, C. Lai, and J. D. Moore, "Paragraph-based prosodic cues for speech synthesis applications," *Speech Prosody*, 2016.

[2] G. M. Ayers, "Discourse functions of pitch range in spontaneous and read speech," *Papers in Linguistics*, 1994.

[3] J. Hirschberg, "Communication and prosody: Functional aspects of prosody," *Speech Communication*, vol. 36, no. 1-2, pp. 31–43, 2002.

[4] H. Bunt, "Dialogue pragmatics and context specification," *Abduction, Belief and Context in Dialogue. Studies in Computational Pragmatics.*, pp. 81–150, 2000.

[5] À. Peiró Lilja and M. Farrús, "Paragraph prosodic patterns to enhance text-to-speech naturalness," in *Speech Prosody*, 2018.

[6] A. Aubin, A. Cervone, O. Watts, and S. King, "Improving speech synthesis with discourse relations." in *Interspeech*, 2019.

[7] J. Hirschberg, "Accent and discourse context: Assigning pitch accent in synthetic speech." in *AAAI*, vol. 90, 1990, pp. 952–957.

[8] Z. Malisz, H. Berthelsen, J. Beskow, and J. Gustafson, "Controlling prominence realisation in parametric dnn-based speech synthesis." in *Interspeech*, 2017.

[9] A. Suni, S. Kakouros, M. Vainio, and J. Šimko, "Prosodic prominence and boundaries in sequence-to-sequence speech synthesis," *arXiv preprint arXiv:2006.15967*, 2020.

[10] S. Shechtman, R. Fernandez, and D. Haws, "Supervised and unsupervised approaches for controlling narrow lexical focus in sequence-to-sequence speech synthesis," in *SLT*, 2021.

[11] A. Rendel, R. Fernandez, Z. Kons, A. Rosenberg, R. Hoory, and B. Ramabhadran, "Weakly-supervised phrase assignment from text in a speech-synthesis system using noisy labels." in *Interspeech*, 2017.

[12] V. Klimkov, A. Nadolski, A. Moinet, B. Putrycz, R. Barra-Chicote, T. Merritt, and T. Drugman, "Phrase break prediction for long-form reading tts: Exploiting text structure information." in *Interspeech*, 2017.

[13] D. Stanton, Y. Wang, and R. Skerry-Ryan, "Predicting expressive speaking style from text in end-to-end speech synthesis," in *SLT*, 2018.

[14] S. Karlapati, A. Abbas, Z. Hodari, A. Moinet, A. Joly, P. Karanasou, and T. Drugman, "Prosodic representation learning and contextual sampling for neural text-to-speech," *ICASSP*, 2021.

[15] Z. Hodari, A. Moinet, S. Karlapati, J. Lorenzo-Trueba, T. Merritt, A. Joly, A. Abbas, P. Karanasou, and T. Drugman, "Camp: a two-stage approach to modelling prosody in context," *ICASSP*, 2021.

[16] S. Tyagi, M. Nicolis, J. Rohnke, T. Drugman, and J. Lorenzo-Trueba, "Dynamic prosody generation for speech synthesis using linguistics-driven acoustic embedding selection," *Interspeech*, 2020.

[17] H. Guo, S. Zhang, F. K. Soong, L. He, and L. Xie, "Conversational end-to-end tts for voice agents," in *SLT*, 2021.

[18] G. Xu, W. Song, Z. Zhang, C. Zhang, X. He, and B. Zhou, "Improving prosody modelling with cross-utterance bert embeddings for end-to-end speech synthesis," *arXiv preprint arXiv:2011.05161*, 2020.

[19] P. Oplustil-Gallegos and S. King, "Using previous acoustic context to improve text-to-speech synthesis," *arXiv preprint arXiv:2012.03763*, 2020.

[20] Y. Wang, D. Stanton, Y. Zhang, R.-S. Ryan, E. Battenberg, J. Shor, Y. Xiao, Y. Jia, F. Ren, and R. A. Saurous, "Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis," in *International Conference on Machine Learning*, 2018.

[21] A. Łańcucki, "Fastpitch: Parallel text-to-speech with pitch prediction," *ICASSP*, 2021.

[22] NVIDIA, "Fastpitch 1.0 for pytorch," https://github.com/NVIDIA/DeepLearningExamples/tree/master/PyTorch/SpeechSynthesis/FastPitch, 2021.

[23] R. Prenger, R. Valle, and B. Catanzaro, "Waveglow: A flow-based generative network for speech synthesis," in *ICASSP*, 2019.

[24] K. Ito and L. Johnson, "The lj speech dataset," https://keithito.com/LJ-Speech-Dataset/, 2017.

[25] N. Cummins, S. Amiriparian, G. Hagerer, A. Batliner, S. Steidl, and B. W. Schuller, "An Image-based Deep Spectrum Feature Representation for the Recognition of Emotional Speech," in *ACM*, 2017.

[26] S. Amiriparian, M. Gerczuk, S. Ottl, N. Cummins, M. Freitag, S. Pugachevskiy, A. Baird, and B. Schuller, "Snore Sound Classification Using Image-Based Deep Spectrum Features," in *Interspeech*, 2017.

[27] P. Oplustil-Gallegos, J. Williams, J. Rownicka, and S. King, "An unsupervised method to select a speaker subset from large multi-speaker speech synthesis datasets," *Interspeech*, 2020.

[28] S. Amiriparian, M. Gerczuk, S. Ottl, and B. Schuller, "Deep spectrum repository," https://github.com/DeepSpectrum/DeepSpectrum, 2021.

[29] T. Wolf, J. Chaumond, L. Debut, V. Sanh, C. Delangue, A. Moi, P. Cistac, M. Funtowicz, J. Davison, S. Shleifer *et al.*, "Transformers: State-of-the-art natural language processing," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 2020, pp. 38–45.

[30] C. McCormick and N. Ryan, "Bert word embeddings tutorial." http://www.mccormickml.com, 2021.

[31] P. Wu, Z. Ling, L. Liu, Y. Jiang, H. Wu, and L. Dai, "End-to-end emotional speech synthesis using style tokens and semi-supervised training," in *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference*, 2019.

[32] H. Li and J. Yamagishi, "Noise tokens: Learning neural noise templates for environment-aware speech enhancement," *Interspeech*, 2020.

[33] S. Kato, Y. Yasuda, X. Wang, E. Cooper, S. Takaki, and J. Yamagishi, "Rakugo speech synthesis using segment-to-segment neural transduction and style tokens—toward speech synthesis for entertaining audiences," in *SSW*, 2019.

[34] NVIDIA, "Mellotron repository," https://github.com/NVIDIA/mellotron, 2021.

[35] M. McAuliffe, M. Socolof, S. Mihuc, M. Wagner, and M. Sonderegger, "Montreal forced aligner: Trainable text-speech alignment using kaldi." in *Interspeech*, 2017.

[36] K. Park and J. Kim, "g2pe," https://github.com/Kyubyong/g2p, 2019.

[37] Y. Jadoul, B. Thompson, and B. de Boer, "Introducing Parselmouth: A Python interface to Praat," *Journal of Phonetics*, vol. 71, pp. 1–15, 2018.

[38] Y. Jia, H. Zen, J. Shen, Y. Zhang, and Y. Wu, "Png bert: Augmented bert on phonemes and graphemes for neural tts," *Interspeech*, 2021.

[39] R. Clark, H. Silen, T. Kenter, and R. Leith, "Evaluating long-form text-to-speech: Comparing the ratings of sentences and paragraphs," *SSW*, 2019.

[40] J. O'Mahony, P. Oplustil-Gallegos, C. Lai, and S. King, "Factors affecting the evaluation of synthetic speech in context," in *SSW (submitted)*, 2021.

11th ISCA Speech Synthesis Workshop (SSW 11)
26-28 August 2021, Budapest, Hungary

Audiobook Speech Synthesis Conditioned by Cross-Sentence Context-Aware Word Embeddings

Wataru Nakata[1], *Tomoki Koriyama*[2], *Shinnosuke Takamichi*[2], *Naoko Tanji*[2]
Yusuke Ijima[3], *Ryo Masumura*[3], *Hiroshi Saruwatari*[2]

[1]Faculty of Engineering, The University of Tokyo, Japan.
[2]Graduate School of Information Science and Technology, The University of Tokyo, Japan.
[3]Nippon Telegraph and Telephone Corporation, Japan.

nakata-wataru855@g.ecc.u-tokyo.ac.jp
t.koriyama@ieee.org

Abstract

This paper proposes an audiobook speech synthesis method that considers a wider range of contexts than a sentence level. The style of the audiobook speech depends not only on the current sentence to be synthesized but also on its neighboring sentences. Therefore, unlike conventional text-to-speech synthesis for isolated sentences, it is necessary to consider the context of the neighboring sentences. Our method utilizes cross-sentence context-aware word embedding, which is obtained by inputting the neighboring and current sentences into BERT. The speech synthesis model, Tacotron2, is conditioned by this word embedding in addition to the current sentence. Experimental results show that taking neighboring sentences into account significantly improves synthetic speech quality.
Index Terms: speech synthesis, cross-sentence context-aware word embedding, BERT, audiobook

1. Introduction

The quality of synthetic speech is getting closer to that of natural human speech [1]. This raises the opportunity of applying text-to-speech (TTS) to a wider range of applications. In this work, we focus on applying TTS for audiobooks. Audiobook speech synthesis is expected to reduce time and monetary requirements by replacing the recordings by professional speakers with automatic generation, and to broaden the selection of available audiobook titles. When applying TTS for audiobooks, we need to keep in mind a series of sentences, that is to be uttered fluently. Specifically, the prosody of human speech often varies on the basis of the neighboring sentences. For example, consider the following passage.

She whispered, "you won't believe it."

When humans read this passage aloud, the style of the second sentence is heavily affected by the first sentence. Considering contexts of neighboring sentences (hereinafter, "*cross-sentence context*") is one of the major challenges when it comes to achieving human-like speech in audiobook speech synthesis.

To model the cross-sentence context in speech, we consider using the techniques of natural language processing (NLP). Taking into account the cross-sentence context in a document is a common practice in NLP tasks, and deep neural networks have been proposed for this purpose. In particular, Bidirectional Encoder Representations from Transformers (BERT) [2] made breakthroughs in various downstream tasks in NLP, such as question answering, natural language inference, and document classification. A key advantage of BERT is that the model parameters of pre-trained BERT can be fine-tuned for a desired task because BERT itself is also a DNN. Moreover, the word embeddings of BERT are context-aware that is, the embedding vectors vary depending on the neighboring linguistic units. BERT can also handle multiple input sentences, which enables us to utilize cross-sentence context for modeling.

It has recently been reported that BERT is also effective for speech synthesis [3, 4, 5, 6]. Hayashi et al. [3] improved the quality of synthetic speech by using context-aware word embeddings from BERT. Fang et al. [4] tried to improve the quality of speech on a relatively small corpus and observed faster convergence during training. Kenter et al. [5] showed that the fine-tuning of BERT is pivotal to improve the quality of synthesized speech. They also demonstrated that a smaller model size of BERT works better. Recently, Jia et al. proposed PnG BERT which is an encoder model for speech synthesis models [6]. In PnG BERT, both phonemes and graphemes are used as the input.

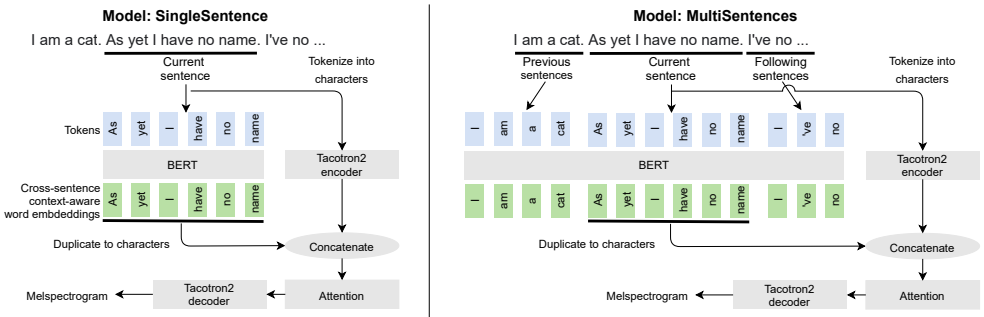
In this study, we propose an audiobook speech synthesis model that reflects the wider context by using the characteristics of BERT. Our proposed model utilizes cross-sentence context-aware word embeddings obtained by inputting multiple sentences to BERT, and Tacotron2 is conditioned by these embeddings. We performed experiments to examine the effectiveness of BERT for audiobook speech synthesis and when using the previous two sentences and current sentence as the BERT input. Subjective evaluation results showed that utilizing the cross-sentence context-aware word embeddings improved the synthetic speech quality. Synthetic speech samples are available online[1].

2. Proposed TTS synthesis model

The proposed model is based on Tacotron2 [1], a widely studied sequence-to-sequence TTS synthesis model. Our proposed model extends Tacotron2 by conditioning its encoder output with cross-sentence context-aware word

[1]https://wataru-nakata.github.io/posts/2021/05/01/ssw11

10.21437/SSW.2021-37



(a) *Model architecture of SingleSentence*

(b) *Model architecture of MultiSentences*

Figure 1: *Model architecture of proposed models. The only difference between SingleSentence and MultiSentences is that the latter also takes neighboring sentences as BERT input. Note that [CLS] and [SEP] tokens and the context-aware word embedding encoder are clipped out for simplification.*

embeddings. The proposed model only changes the structure of the encoder and not the decoder. Therefore, the structure of the decoder is identical in all compared models. We propose two models, SingleSentence and MultiSentences. SingleSentence only takes current sentence as input while MultiSentencse takes the current sentence and neighboring sentences as input.

2.1. SingleSentence

Figure 1a shows the model architecture of SingleSentence. SingleSentence takes the current sentence as input and outputs a melspectrogram. The current sentence is input to both BERT and the Tacotron2 encoder. The context-aware word embeddings from BERT then goes through a context-aware word embedding encoder. The context-aware word embedding encoder consists of two fully connected layers with ReLU activation. This is mainly used for dimensional reduction. This architecture of SingleSentence is similar to the subword-level model in [3]. However, SingleSentence concatenates the outputs of the context-aware word embedding encoder and Tacotron2 encoder, while the subword-level model has an attention mechanism for the BERT output. The word embedding is a word-level vector whereas the encoder output of Tacotron2 is a character-level one. To match the length of the vector sequences, we simply duplicated each context-aware word embedding output with their wordpiece character counts in a similar manner to [5]. With this model, we expect to synthesize speech while reflecting each word's meaning by using context-aware word embedding.

2.2. MultiSentences

Figure 1b shows the model architecture of MultiSentences. This model takes not only a text to be spoken but also neighboring sentences as input. In this study, we use the previous two sentences. This makes the model take the cross-sentence context into account. The Tacotron2 encoder only takes text to be spoken as input, while BERT takes the current sentence and its neighboring sentences as input. Except for taking multiple sentences as input, this model is identical to SingleSentence.

3. Experiments

We evaluated three models: Tacotron2, SingleSentence, and MultiSentences. For SingleSentence and MultiSentences, we evaluated on both before and after the fine-tuning of BERT.

3.1. Experimental conditions

We used the publicly available JSUT [7] and newly released J-KAC (see Appendix A for details) corpora for pretraining and fine-tuning, respectively. These are single-speaker corpora. The JSUT corpus consists of the reading-style speech of isolated sentences by a single female speaker. The J-KAC corpus consists of the very expressive continuous speech of audiobooks and kamishibai (picture stories) by a single male speaker. We downsampled the speech signals to 22.5 kHz in advance and segmented it into a sentence level. For pretraining, we split the JSUT corpus into 7496 and 100 utterances as training and development sets, respectively. For fine-tuning, we split the J-KAC corpus into 4117 (6 hours, 26 books), 100, and 97 (1 book) utterances as training, development, and test sets, respectively. The test set was open to others; no overlap existed in sentences and documents. The generated melspectrogram configurations were 80 dimensions, with the frame length of 1024 samples and frame shift of 256 samples. For input to the Tacotron2 encoder, we used katakana (i.e. Japanese pronunciation symbol) sequence.

Our training procedure involved three steps. First, we pretrained Tacotron2, SingleSentence, and MultiSentences using JSUT with frozen BERT weights. Second, we trained all models using J-KAC with frozen BERT weights. Finally, we performed fine-tuning of BERT using J-KAC for SingleSentence and MultiSentences. During the fine-tuning, the weights unfrozen except for the embedding layer, as some wordpieces did not appear in J-KAC. We conducted evaluations on the following five models.

- Tacotron2
- SingleSentence (**without** fine-tuning of BERT)
- MultiSentences (**without** fine-tuning of BERT)

- SingleSentence (**with** fine-tuning of BERT)
- MultiSentences (**with** fine-tuning of BERT)

For the pretrained BERT model, we used the one provided by akirakubo[2]. Specifically, we used the model trained using AozoraBunko (6 million sentences) and Japanese Wikipedia (3 million sentences) tokenized by SudachiPy with SudachiDict_core-20191224 for 2 million steps.

For optimization, we used an Adam [8] optimizer with $\alpha = 1 \times 10^{-4}$, $\beta_1 = 0.9$, and $\beta_2 = 0.99$ with the L2 weight regularization of 1×10^{-6}. When performing fine-tuning of BERT, we used the small L2 weight regularization of 1×10^{-9} to avoid catastrophic forgetting. Batch size was 128 distributed across four NVIDA V100 GPUs except when fine-tuning BERT. During the fine-tuning of BERT, we used the batch size of 64. For the loss function of Tacotron2, SingleSentence, and MultiSentences, we used the mean squared error of melspectrograms. We also implemented the teacher forcing on the decoder to stabilize the training.

When generating the speech, we applied a temperature softmax function on location-sensitive attention mechanism between the encoder and decoder of Tacotron2 with T = 0.5 to stabilize the speech generation in the same way as [9], which used an expressive speech dataset. In fact, without using temperature softmax, we failed to synthesize speech in most cases. As a vocoder, we used WaveRNN [10] trained on the JVS [7] corpus.

The codes of the experiments were based on NVIDIA's Tacotron2[3] implementation.

3.2. Evaluation methods

We evaluated synthesized speech with two objective metrics: Mel-Cepstral Distortion (MCD) [11] and Gross Pitch Error (GPE) [12]. When calculating these metrics, the duration of synthetic and original speech samples were aligned using FastDTW [13].

Average Mel-Cepstral Distortion (MCD)

MCD is calculated as follows.

$$\mathrm{MCD}_k = \frac{1}{T} \sum_{t=0}^{T-1} \sqrt{\sum_{k=1}^{K} (c_{t,k} - \hat{c}_{t,k})^2} \qquad (1)$$

where $c_{t,k}$ and $\hat{c}_{t,k}$ denote the k-th mel-cepstral coefficients of the t-th frames of original and synthetic speech respectively. We used $K = 13$ in the same way as [4].

Gross Pitch Error (GPE)

Gross pitch error refers to the proportion of voiced frames that deviate by more than a given ratio in pitch signal of the synthesized speech compared to the original speech. In this work, we counted pitch errors of more than 20 % as gross pitch error.

[2] https://github.com/akirakubo/bert-japanese-aozora

[3] https://github.com/NVIDIA/tacotron2

Table 1: *Objective evaluation results.*

Model	MCD[dB]	GPE
Tacotron2	4.228	0.365
SingleSentence	4.161	0.359
SingleSentence (finetuned)	4.229	0.374
MultiSentences	4.250	0.310
MultiSentences (finetuned)	4.205	0.318

3.2.1. Subjective evaluation

We evaluated the five models on three tasks: two Naturalness Mean Opinion Score (MOS) tests and 1 AB test. The naturalness MOS test included the evaluation of speech samples of both one-sentence and five-sentence lengths to examine the naturalness of a series of sentences. On each MOS test, human raters were asked to rate how natural each speech was on a 5-point scale. The number of raters was 60 and each rator evaluated 15 samples in total.

On the AB tests, raters were asked to select which of the five-sentence speech samples was more preferable for reading of a picture book. The number of raters was 40, and each rator evaluated 10 pairs.

For five-sentence speech, we generated speech for each sentence individually and concatenated them with 400 ms of silence between each sentences.

3.3. Results

Table 1 shows the results for objective evaluation using GPE and MCD. The difference of MCD was marginal in all compared models. On the other hand, the GPEs of MultiSentences and MultiSentences (fine-tuned) were significantly smaller than the other methods. This result suggests that the pitch of synthetic speech gets closer to that of natural human speech by using cross-sentence context-aware word embeddings.

Table 2 shows the subjective test results for the naturalness MOS test on one-sentence speech. In all cases, the scores were lower when we incorporated BERT into Tacotron2. One possible reason for this is that generated speech got expressive when we incorporated BERT, which resulted in a lower naturalness MOS score. Table 3 shows the naturalness MOS test results on five-sentence speech. In contrast to the results for one-sentence speech, MultiSentences (fine-tuned) outperformed the other models. This was most likely due to the speaker consistency. Specifically, Tacotron2 often made mistakes when selecting an appropriate speech style. When speech samples were concatenated to make five-sentence speech, this phenomenon became more apparent because the style of speech changes drastically among the sentences. This would result in a lower MOS. In fact, Tacotron2 had a low MOS score for the dialog speech that switches styles among sentences, but MultiSentences (fine-tuned) improved it.

Table 4 shows the results for the AB test. We can see here that MultiSentences was preferable for reading a picture book both before and after the fine-tuning of BERT. Moreover, MultiSentences was preferable to Tacotron2 when BERT was fine-tuned. Even though SingleSentence was able to utilize linguistic information from BERT, we

Table 2: *Naturallness MOS evaluation on one-sentence speech with 95% confidence intervals.*

Model	MOS
Tacotron2	3.450 ±0.141
SingleSentence	2.710 ±0.142
SingleSentence (fine-tuned)	2.676 ±0.162
MultiSentences	2.933 ±0.167
MultiSentences (fine-tuned)	2.900 ±0.164

Table 3: *Naturalness MOS evaluation on five-sentence speech with 95% confidence intervals. The model in* ***bold text*** *shows a significantly better result than Tacotron2.*

Model	MOS
Tacotron2	2.844 ±0.138
SingleSentence	2.628 ±0.144
SingleSentence (fine-tuned)	2.750 ±0.130
MultiSentences	2.767 ±0.130
MultiSentences (fine-tuned)	3.144 ±0.128

did not observe any improvement in either the naturalness MOS or AB test in our settings. These findings are different from what was reported in [3]. However, note that we used different model configurations and trained with a different language from [3].

4. Effect of modifying the previous sentences

We analyzed how the pitch of synthetic speech from MultiSentences (fine-tuned) changes by modifying previous sentences. The original input was as follows:

> ありたちが、ゾロゾロゾロゾロえさをさがしてあるいています。いちばんまえのありくんがいいました。「このあいだは、チョコレートにおせんべい、アイスクリームもおちてたね。」

which means,

> A group of ants were walking around, looking for food. The foremost ant said, *"The other day I found chocolate, rice crackers and ice cream on the ground."*

After the modifications, the previous sentences changed as follows:

> ありたちが、ゾロゾロゾロゾロえさをさがしてあるいています。いちばんまえのありくんが**大声で**いいました。「このあいだは、チョコレートにおせんべい、アイスクリームもおちてたね。」

which means,

> A group of ants were walking around, looking for food. The foremost ant said **loudly**, *"The other day I found chocolate, rice crackers and ice cream on the ground."*

We also prepared the previous sentences with antonymous modification:

> ありたちが、ゾロゾロゾロゾロえさをさがしてあるいています。いちばんまえのありくんが**小**

Table 4: *Results for AB test. Raters were asked to choose which speech was preferable for a picture book speech. Conf. shows 95% confidence intervals.* ***Bold text*** *shows results with significant difference.*

Method A	Scores	Conf.	Method B
Tacotron2	0.533 vs. 0.466	0.048	SingleSentence
Tacotron2	0.423 vs. **0.578**	0.049	MultiSentences
SingleSentence	0.400 vs. **0.600**	0.048	MultiSentences
Tacotron2	0.512 vs. 0.483	0.049	SingleSentence (fine-tuned)
Tacotron2	0.307 vs. **0.693**	0.045	MultiSentences (fine-tuned)
SingleSentence (fine-tuned)	0.302 vs. **0.698**	0.045	MultiSentences (fine-tuned)

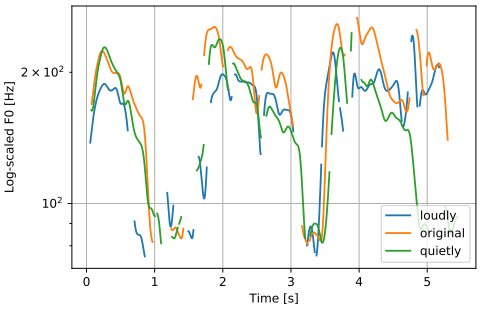


Figure 2: *Pitch change by modifying previous sentences. Note that y axis is shown in log scale. We applied trajectory smoothing [14] to the original F0 for better visualization.*

> **声で**いいました。「このあいだは、チョコレートにおせんべい、アイスクリームもおちてたね。」

which means,

> A group of ants were walking around, looking for food. The foremost ant said **quietly**, *"The other day I found chocolate, rice crackers and ice cream on the ground."*

The difference from the original sentence is shown in **bold** text, and in English translations, the current sentence is shown in *italics*. Figure 2 shows the F0 plot for before and after the modification. From this result, we can see that the generated speech was influenced by the previous sentences. We also tried to control the generated speech's emotion by modifying the previous sentences, but there was no meaningful change. The modified speech is available for listening on our speech sample page[1].

5. Conclusion

In this work, we proposed an audiobook speech synthesis model that utilizes both the current sentence and

the neighboring sentences as input to use cross-sentence context-aware word embeddings from BERT. Subjective evaluation results with generated five-sentence speech samples showed that the quality of speech improved by using the neighboring sentences. We also found that fine-tuning BERT further improved the generated speech quality.

Potential future work includes applying the proposed model for longer sentences, utilizing non-textual information such as paragraph number, and using different BERT configurations or analysis on how the neighboring sentences affect the synthetic speech.

6. References

[1] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan, R. A. Saurous, Y. Agiomvrgiannakis, and Y. Wu, "Natural TTS synthesis by conditioning WaveNet on mel spectrogram predictions," in *2018 ICASSP*, 2018, pp. 4779–4783.

[2] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186.

[3] T. Hayashi, S. Watanabe, T. Toda, K. Takeda, S. Toshniwal, and K. Livescu, "Pre-trained text embeddings for enhanced text-to-speech synthesis," in *Interspeech 2019*, 2019.

[4] W. Fang, Y. Chung, and J. R. Glass, "Towards transfer learning for end-to-end speech synthesis from deep pre-trained language models," *CoRR*, vol. abs/1906.07307, 2019.

[5] T. Kenter, M. Sharma, and R. Clark, "Improving the prosody of rnn-based english text-to-speech synthesis by incorporating a bert model," in *INTERSPEECH 2020*, 2020, pp. 4412–4416.

[6] Y. Jia, H. Zen, J. Shen, Y. Zhang, and Y. Wu, "PnG BERT: Augmented BERT on Phonemes and Graphemes for Neural TTS," 2021.

[7] S. Takamichi, R. Sonobe, K. Mitsui, Y. Saito, T. Koriyama, N. Tanji, and H. Saruwatari, "JSUT and JVS: Free Japanese voice corpora for accelerating speech synthesis research," *Acoustical Science and Technology*, vol. 41, no. 5, pp. 761–768, 2020.

[8] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *International Conference on Learning Representations*, 2014.

[9] X. Cai, D. Dai, Z. Wu, X. Li, J. Li, and H. Meng, "Emotion controllable speech synthesis using emotion-unlabeled dataset with the assistance of cross-domain speech emotion recognition," *ICASSP 2021*, 2021.

[10] N. Kalchbrenner, E. Elsen, K. Simonyan, S. Noury, N. Casagrande, E. Lockhart, F. Stimberg, A. van den Oord, S. Dieleman, and K. Kavukcuoglu, "Efficient neural audio synthesis," in *Proceedings of the 35th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 80. PMLR, 10–15 Jul 2018, pp. 2410–2419.

[11] R. Kubichek, "Mel-cepstral distance measure for objective speech quality assessment," in *Proceedings of IEEE Pacific Rim Conference on Communications Computers and Signal Processing*, vol. 1. IEEE, 1993, pp. 125–128.

[12] L. Rabiner, M. Cheng, A. Rosenberg, and C. McGonegal, "A comparative performance study of several pitch detection algorithms," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 24, no. 5, pp. 399–418, 1976.

[13] S. Salvador and P. Chan, "Toward accurate dynamic time warping in linear time and space," *Intelligent Data Analysis*, vol. 11, no. 5, pp. 561–580, 2007.

[14] S. Takamichi, K. Kobayashi, K. Tanaka, T. Toda, and S. Nakamura, "The NAIST text-to-speech system for the Blizzard Challenge 2015," in *Proc. Blizzard Challenge workshop*, vol. 2. Berlin, Germany, 2015.

A. Japanese audiobook corpus J-KAC

We developed a very expressive corpus for Japanese audiobook speech, named J-KAC (Japanese kamishibai and audiobook corpus). This corpus includes nine hours (26 audiobooks and 17 kamishibai) of studio-quality 48-kHz sampled speech uttered by a single male professional speaker. The audio files are stored for each book, and the documents are structured in chapter, paragraph, style, and sentence levels. The "style" level has a binary label of inner sentences: "narrative style" or "character-acting style." The "sentence" level has a temporal alignment to audio. In addition to audio and documents, the corpus includes illustrations obtained by scanning products, which was done with permission from the book authors and publishers. The illustrations have various characters and background images, etc. The corpus is available for only research purposes. More information is available on our project page[4].

[4] `https://sites.google.com/site/shinnosuketakamichi/research-topics/j-kac_corpus`

11th ISCA Speech Synthesis Workshop (SSW 11)
26-28 August 2021, Budapest, Hungary

Lipsyncing Efforts for Transcreating Lecture Videos in Indian Languages

Mano Ranjith Kumar M, Jom Kuriakose*, Karthik Pandia D. S, Hema A. Murthy*

Department of Computer Science and Engineering, Indian Institute of Technology, Madras

manoranjith@smail.iitm.ac.in, jom@cse.iitm.ac.in,
karthik.pandia@gmail.com, hema@cse.iitm.ac.in

Abstract

This paper proposes a novel lip-syncing module for the transcreation of lecture videos from English to Indian languages. The audio from the lecture is transcribed using automatic speech recognition. The text is translated and manually curated before and after translation to avoid mistakes. The curated text is synthesized using the Indian language end-to-end-based text-to-speech synthesis systems. The synthesized audio and video are out-of-sync. This paper attempts to automate this process of producing video lectures lip-synced into Indian languages using different techniques.

Lip-syncing an educational video with the Indian language audio is challenging owing to (a) the duration of Indian language audio being considerably longer or shorter than that of the original audio, (b) the extempore speech causes the audio in the source videos to have long silences. Any modification to the speed of audio can be unpleasant to listeners. The proposed system non-uniformly re-samples the video to ensure better lip-syncing. The novelty of this paper is in the use of HMM-GMM alignments in tandem with syllable segmentation using group delay, as visemes are closer to syllables. The proposed lip-syncing techniques are evaluated using subjective evaluation methods. Results indicate that accurate alignment at the syllable level is crucial for lip-syncing.

Index Terms: Automatic dubbing, Lip-syncing, Transcreation, Group delay.

1. Introduction

The medium of instruction for higher education globally is predominantly English. Educational content that is available online has been growing exponentially in recent times. Language is the major barrier to use these resources in places like India, where people use almost 1652 different languages and 22 official languages to communicate. Recent advancements in speech recognition (ASR), machine translation (MT), and speech synthesis (TTS) have enabled us to develop systems that automatically dub educational videos in Indian languages. This paper proposes and analyses various lip-syncing methods that improve the quality of transcreation of lecture videos to Indian languages.

Automatic dubbing is an extension of speech-to-speech translation [1]. It involves (i) transcription of speech from the video, (ii) translation of the transcribed text into the target language, (iii) synthesizing target language audio, and (iv) lip-syncing synthesized audio with the original video. In automatic dubbing, generally, the machine translation is done carefully, such that the number of syllables in the source and target audio is almost matched. For example, dubbing of movies preserves the duration of the video and forces the audio to align within the duration of the video. This is attained by manually curating the translated text to match the video duration and lip movements in such a way that the lip-synced video does not sound unnatural. The lip-syncing techniques proposed in this paper does not force the synthesised audio length to match the video length. Instead, the video is re-sampled to match the audio duration. In lecture videos, conveying the underlying concepts is prioritized more than matching the syllable rate, so that the conveyed information is not lost in the process. Changing the speech rate to match the duration of the video is also not preferred since our analysis showed that the lip-synced videos with variable speech rate are not preferred for by the viewers. Adding to this, unlike English, Indian languages are word order free, and hence the length of the translated text is generally longer or shorter than that of source English text. Due to this, the synthesized audio is significantly longer or shorter than that of the source audio. Hence, we prefer transcreation of video to target language over simple dubbing by preserving the information conveyed in the source video lectures.

The previous works in lip-syncing focused on different techniques for aligning source video and target audio or text. In recent work by [1, 2], TED talks are automatically dubbed using a prosodic alignment module that aligns the speech segments from source audio with the machine-translated text. The attention mechanism is used in [3] to find a plausible phrasing for the translated text, which is synthesized and added to the source video. Other than finding alignments, changing the speed of synthetic speech to fit into the subtitle duration is attempted in [4, 5]. An end-to-end audio-visual translation system is trained on thousands of hours of data from all domains in [6], and adapted to a specific domain and speaker. Many works in the literature, including [7, 8] suggest that there is a co-relation between visual speech unit (visemes) and phonetic speech unit (phonemes). Syllables are combination of phonemes. Using syllable level boundaries for lip-syncing makes sure that the visemes are not spliced in the video.

This is the first attempt in the literature to transcreate educational lectures from English to Indian languages. Lip-syncing video with the Indian language audio is challenging due to the longer duration of the translated, and synthesised audio. The domain of educational videos also make it more challenging, as the original speaker in the lecture videos have long silences when the lecturer is dis-fluent. The transcreated video's naturalness depends not only on exact alignments of synthesized audio with lip movements but also on the long pauses, head and hand movements, and expressions of the speaker in the original video. Hence in our lip-syncing method, these attributes of the source video are preserved in the final video.

The paper proposes lip-syncing methods using word-level alignments on the synthesized audio. A group delay (GD) based syllable segmentation is used to fine-tune the word boundaries, which further improves the naturalness by preserving the viseme units is proposed. The lip-syncing developed in this pa-

*Equal contribution by both authors.

10.21437/SSW.2021-38

per attempts to attain isochrony [9], where audio is synchronized with the speaker's lip movements. These systems are evaluated using subjective evaluation method; mean opinion score (MOS) to show that word-level alignments are very important for better lip-syncing output.

The rest of the paper is organized as follows. In Section 2, the video transcreation system is discussed. Different lip-syncing systems proposed along with preliminary lip-syncing systems are detailed in Section 3. Section 4 explains the evaluation setup for these systems, followed by results and discussions in Section 5 and Section 6 concludes the paper.

2. Pipeline of Video Transcreation System

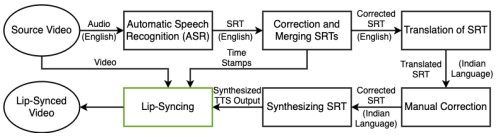


Figure 1: *Flowchart of video transcreation system. Lip-syncing module attempted in this paper is shown in green color.*

The flowchart for video transcreation is given in Figure 1. The input lecture video is split into segments by detecting long silence regions using speech activity detector of Kaldi toolkit [10], and these segments are transcribed using the ASR system[1] [11] to create the Sub-Rip Text (SRT) file. Technical lectures contain words that are domain-specific and includes a significant amount of mathematical equation and notations. These are preserved, manually verified and corrected before translating to the target language, as shown in the flowchart (Figure 1). The machine translation to Indian languages is done using language translation APIs [12, 13]. The translated text after manual verification is synthesized using the end-to-end (E2E) [11, 14] based TTS system proposed in [15]. The TTS models are trained on Indic TTS [16] data-set. The synthesized audio along with the source video and subtitle file, is given as input to the lip-syncing system. The lip-syncing system is described in detail in Section 3.

3. Lip-Syncing System

The lip-syncing techniques discussed in this paper segment the TTS synthesized audio and aligned them with the source video by re-sampling the video. In the following sub-sections, we discuss two simple preliminary systems based on re-sampling, silence detection, and two proposed systems using ASR alignments obtained from TTS synthesized audio and group delay-based segmented ASR alignments.

3.1. Preliminary Systems

As an initial attempt at lip-syncing, we did a simple re-sampling of the video segments. The trailing and beginning silences of the synthesized audio are discarded before re-sampling. The source video is then re-sampled to match the audio segments' duration by interpolating the video to match the duration of synthesized audio. The flow chart of this method is highlighted in red color in Figure 2. This attempt is a basic approach to lip-syncing without using any silence alignments or word boundaries. This method does not detect long-pause regions in the

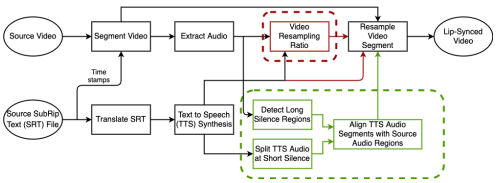


Figure 2: *Flow chart of preliminary methods attempted. Pipeline for re-sampling based lip-syncing is shown in red color and pipeline for silence detection based lip-syncing is shown in green color.*

source video within a SRT segment. Hence, in the transcreated video, there are sections where the lecturer's lips are not moving while synthesized audio is playing. The final video also has sections where the video speeds up or slows down very aggressively due to the large difference between source audio and target audio duration. This is mainly due to the duration differences between sentences in English and Indian languages and the long pauses in source video within the SRT segment.

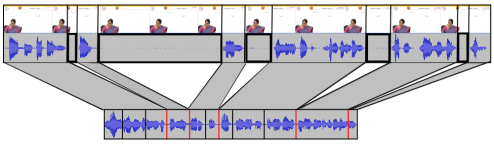


Figure 3: *Alignment of audio after silence detection. Box with black border shows long-silence regions in source video. Red line denotes location of spliting in TTS audio. The alignment between video and audio is also shown.*

To address this issue, transcreated videos need to ensure that the audio plays only when the lips are moving. The lip movements of the speaker should synchronize with the regions of speech in a source audio segment. Therefore in the next attempt, we try to detect the silence regions and keep it as it is and align the synthesized audio with the regions where the speaker is speaking as shown in Figure 3. The flowchart of this system is given in Figure 2, where the green box highlights the flow of this attempt. We try to detect the long silence regions in the audio from the source video, and short pauses in the TTS synthesized audio and try to align them using Algorithm 1. The algorithm tries to map the TTS segments with the segments of the source audio. This is done by finding the silence regions in the source audio and splitting the TTS audio in the same ratio as in the source audio. This process of splitting and aligning the TTS audio is initially done for the longest silence region in the source audio, followed by the second-longest silence, and so on recursively until the TTS audio is completely mapped with the source audio. Refer Algorithm 1 for more details. The same algorithm is used in Proposed System 1 and Proposed System 2 for aligning the source video with the TTS audio.

Silence detection works well in finding the silence regions. However, since the silence detection only look for the silence regions, the silence detection will often detect short silence regions in the middle of certain words that are often occur together. This can lead to the system splicing the synthesized audio in the middle of those words and inserting silences. Even

[1] https://asr.iitm.ac.in/NPTEL/Transcribe/

Algorithm 1 Mapping of source intra-SRT-segment alignments to TTS intra-SRT-segment alignments

Input: *src_segs(1:N):- List of source intra-SRT-segments of length N obtained after silence detection of a source audio segment*

tts_segs(1:M):- List of TTS intra-SRT-segments of length M after silence detection after TTS audio segment (Obtained from word boundaries of ASR incase of Proposed System 1 and Proposed System 2)

Format *of src_segs(1:N) & tts_segs(1:M):-*
<start-time> <end-time> <duration> <label>
where (label==0) => silence & (label==1) => speech

Output: *map(,):- List of mappings between src_segs indices & tts_segs indices*

procedure ALIGN_SEGMENTS(*src_segs(1:N),tts_segs(1:M)*)
▷ *This function returns alignment of source segements and target segments*

variables
map(a,b):- represents that $index$ *a in src_segs is aligned with* $index$ *b in tts_segs.*
end variables

if *length(src_segs)==1* ***OR*** *length(tts_segs)==1* ***then***
▷ *This is the base condition for recursion*
return
else
src_split = SOURCE_SPLIT_INDEX(*src_segs(1:N)*)
speech_ratio =
SPEECH_RATIO(*src_segs(1:N),src_split*)
tts_split =
TTS_SPLIT_INDEX(*tts_segs(1:M),speech_ratio*)
return ALIGN_SEGMENTS(*src_segs(1:src_split), tts_segs(1:tts_split)*) + ***map***(*src_split,tts_split*)
+ ALIGN_SEGMENTS(*src_segs(src_split:N),tts_segs(tts_split:M)*)

though the lip will sync with the video, the synthesized audio being played will lose its understand-ability considerably in this case. We address this issue of splicing the audio between words in Proposed System 1 using word-level alignments.

3.2. Proposed System 1: Word level alignment method

Silence boundaries do not have information of the target text and hence can have intra-word splits. Word level alignments are required to avoid splitting between words. Word level boundaries are obtained using hidden Markov model - Gaussian mixture model (HMM-GMM) based ASR system [10]. For training the ASR systems, a common label set (CLS) [17] lexicon representation of the transcription is obtained using the unified parser [18] for Indian languages. The word-level alignments obtained are used to align the synthesized audio at the word level with the source video. The word-level alignments using HMM-GMM ASR are only obtained for the synthesized audio since silence regions are better-detected using signal processing techniques using short-term-energy (STE) for source audio. The detection of silence in the source is very important for better lip-syncing. The flowchart of Proposed System 1 is highlighted in red in

procedure SOURCE_SPLIT_INDEX(*src_segs(P:Q)*)
▷ *This function returns a index of largest silence region in source segments*

variables
sil_dur_list:- list of duration of silence segments in src_segs.
end variables

sil_dur_list = (src_segs(P:Q)(label==0).duration)
max_dur = max(sil_dur_list)
return *max_dur*

procedure SPEECH_RATIO(*src_segs(P:Q),src_split)*
▷ *This function returns ratio of speech on both sides of silence region*

variables
speech_dur:- duration of speech before split.
full_dur:- duration of speech in whole src_segs.
end variables

speech_dur = $\sum$ *(src_segs(P:src_split)(label==1).duration)*
full_dur = $\sum$ *(src_segs(P:Q)(label==1).duration)*
return *speech_dur/full_dur*

procedure TTS_SPLIT_INDEX(*tts_segs(P:Q),speech_ratio)*
▷ *This function returns index of split in TTS segments,which is close to speech ratio*

variables
X*:- list of indices of tts_segs(P:Q) segments.*
tts_ratio(x)*:- TTS speech ratio at split index* x*.*
end variables

for x `in` X ***do***
tts_ratio(x) =
$\sum$ *(tts_segs(P:*x*)(label==1).duration) /*
$\sum$ *(tts_segs(P:Q)(label==0).duration)*
return *Index* x *for min of (abs(speech_ratio* $-$ *tts_ratio(*X*)))*

Figure 4. An example for ASR alignment obtained is given in Figure 5.

The system detects the words correctly, but in some cases, the beginning or ending of the word can be misaligned by few milliseconds. While this is not an issue in ASR results, the misalignment can be due to the word ending not being an ending of a syllable. These boundaries may result in clicking noises, which can sound unnatural. Proposed System 2 tries to correct these boundaries to obtain better word-level boundaries by using group delay and energy-based correction.

3.3. Proposed System 2: Word alignments correction using group delay (GD) & spectral flux (SF)

Minimum phase group delay (GD) of short-term-energy (STE) and sub-band spectral flux (SBSF) together are used for identifying syllable boundaries. The algorithm for finding sylla-

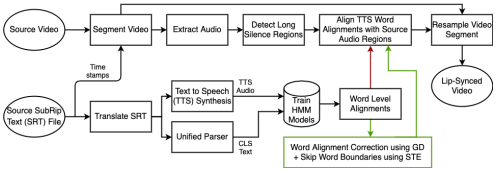


Figure 4: *Flowchart of proposed lip-syncing methods. Pipeline for Proposed System 1 is shown in red color and pipeline for Proposed System 2 is shown in green color.*

ble boundaries using GD and SBSF is explained in [19]. The word boundaries obtained from the HMM-GMM ASR model is corrected using the syllable boundaries obtained from GD and SBSF. After alignment correction, the short-term energy (STE) is calculated at these boundaries, and high energy boundaries are excluded from alignments. Similar to Proposed System 1, the silence detection algorithm is used to find the long silence regions inside the source audio segments, and corrected word boundaries of synthesized audio are aligned with source audio segments using the Algorithm 1.

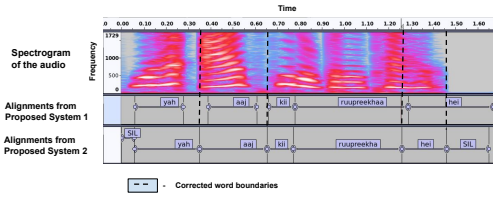


Figure 5: *Word level alignment of Hindi utterance "Yah aaj kii ruupreekhaa hei" (English: This is today's outline) obtained using Proposed System 1 and Proposed System 2.*

During the speech, some words are articulated together (Ex. metro network, deep learning). Splitting the synthesized audio in between those words can lead to clicking sounds. This is primarily owing to the co-articulation between adjacent syllables that make up a word(s). In Indian languages, it is very common to find that the word morpheme at the end of one word is merged with the succeeding word. Clearly, more than one syllable or syllables across word boundaries is perhaps associated with a viseme. We attempt to correct this by using word boundaries and syllable boundaries, where the short-term energy is very low, suggesting that the co-articulation is small.

4. Evaluation

In addition to the systems discussed in Section 3, we compare our systems with an additional system that transcreates video using Google cloud API[2]. This system uses ASR, MT, and TTS modules from Google cloud APIs, to transcreate videos. This system will be henceforth referred to as Baseline System. Unlike the systems discussed above, in the baseline system, the TTS audio is generated based on the duration of the video segment. By controlling the speaking rate in the TTS module, the duration of audio is matched with the source video.

Since it was evident that proposed systems are better than preliminary attempts, we evaluated only Proposed System 1 and 2 against the Baseline System. These three systems were evaluated using, mean opinion scores (MOS) test. The configuration of the test is given in the following subsection. Samples of final trans-created videos for all preliminary, proposed, and baseline systems are given in this link: https://www.iitm.ac.in/donlab/preview/lip_sync_21/index.html

[2]https://github.com/google/making_with_ml/tree/master/ai_dubs

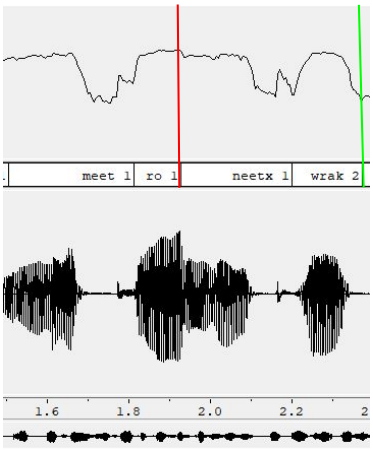


Figure 6: *The synthesized waveform of "metro network" and it's corresponding syllable boundaries (Metro ("meet","ro"), Network ("neetx","wrak")) along with STE plot. The red line corresponds to the word boundary where the energy is high and the green line corresponds to the word boundary, where the energy is low.*

4.1. MOS Test

The mean opinion scores (MOS) test is originally designed to evaluate TTSes. A set of 15 evaluators were asked to evaluate 5 lip-synced videos of approximately one and half minutes to two and half minutes, in comparison with the corresponding source video. Two male and two female speaker video segments are chosen for evaluation from Proposed System 1 and Proposed System 2, along with a Baseline video segment. The video segments are chosen at random from the whole lecture. The speakers are not repeated in the 5 randomly chosen video segments to make sure that the viewer does not get used to the speaker. The evaluators are asked to rate the naturalness of the video, including the lip movement. The videos are played at random to the evaluators. The evaluators are asked to give a rating from 0 to 100 for each video segment, where 0 being no synchronization between the audio and the video, and 100 being perfectly lip-synced.

5. Results and Discussion

As seen in Table 1, the Proposed System 2 has highest MOS score of *82.55*, whereas Proposed System 1 has a MOS score of *75.64*. The lip-synced videos using Proposed System 2 are rated higher due to the correction of word boundaries using group delay-based segmentation, along with the splitting of synthesised audio at low energy word boundaries. The importance of correcting the word boundary using group delay-based segmen-

Table 1: *Mean opinion score (MOS) of Hindi language lip-syncing for Proposed System 1 and Proposed System 2 along with Baseline System.*

Systems	MOS Score
Proposed System 1: Word-level alignment method	75.64
Proposed System 2: Word alignments correction using group delay (GD) spectral flux (SF) correction	**82.55**
Baseline System: Google Cloud API based application: ML-Powered Translation	62.98

tation is shown by an example in Figure 5. The figure shows spectrogram of Hindi utterance "*yah aaj kii ruupreekhaa hei*" (English: "*This is today's outline*") in common label set (CLS) format defined in [17]. Word boundaries of the word "*aaj*" and "*ruupreekha*" are more accurate in Proposed System 2 after boundary correction. Silence regions are also recognised with very high precision after correction, compared to the ASR boundaries obtained using Proposed System 1 as shown in Figure 5.

The word boundaries don't always correspond to the decay of the speech signal but can also be an onset for the next word. This can be due to the co-articulation of two words together. This can be seen in Figure 6, where the words "*metro network*" are pronounced together. The STE value at the end of the word "*metro*" (refer red line) is higher than that of the STE value at the end of the word "*network*" (shown in green). Thus, splitting in-between "*metro network*" is avoided. Splitting the signal between these words will lead to unnatural artifacts like clicking sounds. Using STE to find these word boundaries has also contributed to the higher performance of Proposed System 2 over Proposed System 1.

The Baseline system has a lower MOS score of *62.98* compared to Proposed System 1 and 2. The Baseline system increases or decreases the rate of speech based on the length of the video segment to which it has to be merged and hence has a varied rate of speech throughout the lip-synced video. This can also be due to the fact that the sentences in Indian languages are generally longer in comparison to those in English. The variable speech rate can be unpleasant to the viewers. In terms of constant speech rate, the Proposed System 1 can be considered as a baseline, and results show that the Proposed System 2 provides an improvement.

Signal processing techniques like group delay and STE can be used in tandem with the machine learning methods (ASR) to find accurate word-level boundaries. The results show that the improvement in word-level alignments has significantly improved the quality of the lip-synced video.

6. Conclusions

Professional dubbing is an expensive and labour intensive process. This work proposes novel techniques to improve the naturalness of auto-transcreated videos. The discussed results shows that syllable level segmentation (Proposed System 2) provides an absolute improvement of *6.9* over simple ASR word level alignment based technique (Proposed System 1). Using visual speech units (visemes) along side syllable segmentation may further improve the observed results for lip-syncing.

7. Acknowledgements

We want to extend our gratitude to Speech Lab, Indian Institute of Technology, Madras (IITM) for their help in generating the SRT files for the lecture videos. We would like to thank the Department of Science and Technology (DST), the Ministry of Electronics and Information Technology (MeitY), Office of the Principal Scientific Adviser (PSA) to the Government of India, for funding the projects, "Text to Speech Generation with chosen accent and noise profile for Aerospace and Industrial domains" (CSE1819172MIMPHEMA), "Natural Language Translation Mission" (CS2021012MEIT003119), "Speech to Speech Machine Translation" (CS2021152OPSA003119), respectively.

8. References

[1] M. Federico, R. Enyedi, R. Barra-Chicote, R. Giri, U. Isik, A. Krishnaswamy, and H. Sawaf, "From speech-to-speech translation to automatic dubbing," *arXiv preprint arXiv:2001.06785*, 2020.

[2] M. Federico, Y. Virkar, R. Enyedi, and R. Barra-Chicote, "Evaluating and Optimizing Prosodic Alignment for Automatic Dubbing," in *Proc. Interspeech 2020*, 2020, pp. 1481–1485. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2020-2983

[3] A. Öktem, M. Farrús, and A. Bonafonte, "Prosodic phrase alignment for machine dubbing," *arXiv preprint arXiv:1908.07226*, 2019.

[4] J. Matoušek and J. Vít, "Improving automatic dubbing with subtitle timing optimisation using video cut detection," in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2012, pp. 2385–2388.

[5] "How to dub a video with ai youtube," https://youtu.be/T2TAAHmNBnE, 02 2021, (undefined 4/5/2021 0:57).

[6] Y. Yang, B. Shillingford, Y. Assael, M. Wang, W. Liu, Y. Chen, Y. Zhang, E. Sezener, L. C. Cobo, M. Denil *et al.*, "Large-scale multilingual audio visual dubbing," *arXiv preprint arXiv:2011.03530*, 2020.

[7] S. Taylor, "Discovering dynamic visemes," Ph.D. dissertation, University of East Anglia, 2013.

[8] H. L. Bear and R. Harvey, "Phoneme-to-viseme mappings: the good, the bad, and the ugly," *Speech Communication*, vol. 95, pp. 40–67, 2017.

[9] F. C. Varela, "Synchronization in dubbing," *Topics in audiovisual translation*, vol. 56, p. 35, 2004.

[10] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlíček, Y. Qian, P. Schwarz, J. Silovský, G. Stemmer, and K. Vesel, "The kaldi speech recognition toolkit," *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*, 01 2011.

[11] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. E. Y. Soplin, J. Heymann, M. Wiesner, N. Chen, A. Renduchintala, and T. Ochiai, "Espnet: End-to-end speech processing toolkit," *CoRR*, vol. abs/1804.00015, 2018. [Online]. Available: http://arxiv.org/abs/1804.00015

[12] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey *et al.*, "Google's neural machine translation system: Bridging the gap between human and machine translation," *arXiv preprint arXiv:1609.08144*, 2016.

[13] V. Mujadia and D. Sharma, "NMT based similar language translation for Hindi - Marathi," in *Proceedings of the Fifth Conference on Machine Translation*. Online: Association for Computational Linguistics, Nov. 2020, pp. 414–417. [Online]. Available: https://www.aclweb.org/anthology/2020.wmt-1.48

[14] R. Prenger, R. Valle, and B. Catanzaro, "Waveglow: A flow-based generative network for speech synthesis," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 3617–3621.

[15] A. Prakash, A. L. Thomas, S. Umesh, and H. A. Murthy, "Building multilingual end-to-end speech synthesisers for indian languages," in *Proc. of 10th ISCA Speech Synthesis Workshop (SSW'10)*, 2019, pp. 194–199.

[16] A. Baby, A. L. Thomas, N. Nishanthi, T. Consortium *et al.*, "Resources for indian languages," in *Proceedings of Text, Speech and Dialogue*, 2016.

[17] B. Ramani, S. Lilly Christina, G. Anushiya Rachel, V. Sherlin Solomi, M. K. Nandwana, A. Prakash, S. Aswin Shanmugam, R. Krishnan, S. Kishore, K. Samudravijaya, P. Vijayalakshmi, T. Nagarajan, and H. A. Murthy, "A common attribute based unified HTS framework for speech synthesis in Indian languages," in *Speech Synthesis Workshop (SSW)*, 2013, pp. 291–296.

[18] A. Baby, N. Nishanthi, A. L. Thomas, and H. A. Murthy, "A unified parser for developing indian language text to speech synthesizers," in *International Conference on Text, Speech, and Dialogue*. Springer, 2016, pp. 514–521.

[19] S. A. Shanmugam, "A hybrid approach to segmentation of speech using signal processing cues and hidden markov models," Ph.D. dissertation, MS Thesis, Department of Computer Science Engineering, IIT Madras, India, 2015.

11th ISCA Speech Synthesis Workshop (SSW 11)
26-28 August 2021, Budapest, Hungary

Homograph disambiguation with contextual word embeddings for TTS systems

Marco Nicolis, Viacheslav Klimkov

Amazon

nicolism@amazon.com, vklimkov@amazon.com

Abstract

We describe a heterophone homograph (simply 'homograph' henceforth) disambiguation system based on per-case classifiers, trained on a small amount of labelled data. These classifiers use contextual word embeddings as input features and achieve state-of-the-art accuracy of 0.991 on the English homographs on a publicly available dataset, without any additional rule system being necessary. We show that as little as 100 sentences are sufficient to train a light-weight dedicated classifier, provided the dataset is sufficiently balanced, i.e. all versions of the homograph are adequately represented. We further add data in cases where the original dataset is deeply unbalanced (i.e. one homograph version overwhelmingly represented). This is effectively a special case of active learning, by which we select additional cases of the under-represented homograph versions and show an 11% relative improvement for such cases. We finally provide a solution to drastically reduce the size of our models, via sparsification.

Index Terms: homograph disambiguation, TTS, front-end

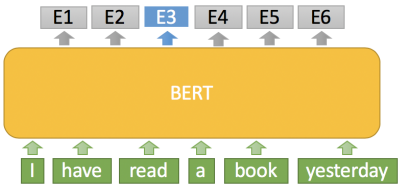
1. Introduction

A homograph is a 'word that is spelled the same as another [...] but which differs in sound *and* meaning, such as *tear* (to separate or pull apart) and *tear* (a secretion from the eye)' ([1, p. 3]). The correct classification of homographs is a long-standing issue affecting the text analysis component of text-to-speech (TTS) systems. While homographs are present in many of the world's languages, this paper focuses on American English only. For the sake of reproducibility we report the performance of the proposed approach on the openly available dataset from [2][1]. We additionally use a small internal dataset to augment the main dataset in cases where one of the two possible pronunciations of a given homograph is severely under-represented in the main dataset. We show this brings about a large improvement for these cases.

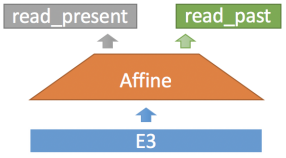
All approaches to homograph disambiguation agree that correctly interpreting the context surrounding the homograph word is key to their disambiguation. Different types of linguistic information are, however, crucial in different cases. The lexical semantic context allows the disambiguation of *lexical* homographs like *bass* (fish vs. musical instrument, both nouns), whereas morphosyntactic constraints regulate the *morphosyntactic* class, which can generally be disambiguated via POS tags (e.g. nominal vs. verbal *impact*).

Rule based and supervised machine learning systems have historically directly encoded the multiple sources of homography in the features used for disambiguation. The algorithm described in [3] heavily depends on collocations and their statistical distribution. For example, the presence of semantically related words within a certain distance from the homograph word (e.g. does the word *water* occur in the ± 20 word window around the target word *bass*?) or the co-occurrence of certain word sequences (e.g. the [*of* lead *in*] sequence correlates with the [lɛd] pronunciation of *lead*, whereas the [*the* lead *in*] sequence correlates with the [lɪd] pronunciation). Twenty years later, [2] feed a multinomial log-linear model (one for each homograph) with similar types of features: word context features (left and right bigrams around the homograph), POS tag and capitalization (*Polish* vs. *polish*).

Transformer-based contextual word embeddings (CWE), i.e. the BERT family ([4], [5], [6], etc.) appear to be ideally suited for the purpose of homograph disambiguation. The vectors they produce are inherently context-sensitive: the vector for *bass* will be different depending on the surrounding context and the information they encode spans across the whole traditonal NLP pipeline, including the lexical semantics and morphosyntactic information needed to disambiguate homographs. See ([7] and [8] for a recent overview). It follows that the adoption of CWEs allows for a unified system encompassing all homographs, be their disambiguation based on morphosyntactic or lexical semantic factors. In the remainder of this paper we show that this approach yields SOTA accuracy; more importantly it does not require complex, expensive to produce and hard to maintain rule-based components.



(a) *Feature extraction for homograph disambiguation*



(b) *Homograph disambiguation with simple classifier*

Figure 1: *Homograph disambiguation pipeline*

2. Model description

Contextual word embeddings (CWEs) provide numerical representation of the word in the given context. They are trained on large text corpora on the masked language modeling task (i.e.

[1] https://github.com/google/WikipediaHomographData

10.21437/SSW.2021-39

prediction of masked words in a sentence). With reproducibility in mind, in this work we use the BERT pretrained model[2] distributed for MXNet ([4]), one of the most widely adopted models openly available. CWE models are resource hungry, which makes makes them not always suitable for a production environment. Thus, we also experimented with the ALBERT model, a light-weight CWE model ([5]). We use the pretrained model[3] distributed for Pytorch.

We only utilize the relevant embedding for the token of interest (homograph), under the assumption that sufficient relevant contextual information is expressed in it. This is depicted on Figure 1. The extracted embedding is then fed to a simple per-homograph classifier. In our experiments we used a logistic regression classifier, trained on pairs of contextual embeddings and correspondent homograph cases. For example, for the homograph "*read*", the training set contains 52 and 60 sentences with past and present form of it respectively.

The per-homograph logistic regression model is trained using the MXNet framework. More complex models, such as multi-layer perceptron, were explored, but did not prove to be beneficial. We also prefer to present the simplest approach, as this makes it easier to reproduce, and it highlights the usefulness of the underlying CWEs for the homograph disambiguation task. The model is trained using stochastic gradient descent using Adam optimizer with a learning rate of $1e-3$. The models were optimized on all training examples for the given word simultaneously. To compensate for the small amount of training data and excessive representation capabilities of input features, it proved beneficial to apply *l2*-regularization with weight 0.01.

Each classifier is a matrix of size (*embedding_dimension* x *homograph_cases_num*), where *embedding_dimension* is either 1024 or 768 depending on the version of the BERT model used, and *homograph_cases_num* is typically 2, corresponding the two pronunciations of a given homograph. That results in 3kB per model or 0.5MB for all 162 homographs from the dataset in case of quantization to float16.

3. Experiments

[2] provide an overview of Google's homograph disambiguation system. The baseline model they describe as Google's production model at the time of writing is rule based, while the paper introduces an ML approach where the features described in section 1 (positional features, capitalizaton, POS tag) are fed to a multinomial log-linear model. Combining rules and ML model yields an important boost in accuracy (see Table 1). We use the results in [2] as our baseline, and report results of our model trained on identical data (Experiment 1) and augmented data (Experiment 2). We follow [2] in reporting both micro accuracy (percentage of correctly classified examples) and macro accuracy (arithmetic mean of per-homograph accuracies) for all our models.

3.1. Experiment 1: Data from [2]

One of the stumbling blocks of systems relying on POS tags for disambiguation concerns the taggers' accuracy for non-trivial cases, for example cases where nouns and verbs share the same orthography (e.g. verbal and nominal *impact*, TIPA). While the accuracy of SOTA taggers is at almost 98%[4] on Penn Treebank data, [9] show that the accuracy of top parsers ranges between 57% and 74% on their crowdsourced dataset specifically targeting ambiguous nouns and verbs. In that paper, trivial cases were filtered out from the dataset[5]: for example, given a N/V ambiguous word (e.g. *impact*), both the [Det + Word] and [Aux + Word] case will provide trivial and exceptionless disambiguation evidence (N for Det + Word: *the impact* and verbal for Aux + Word *will impact*). [9] use the extra data from their challenge dataset to retrain the tagger in [11]; they in addition add ELMo embeddings [10] to their pipeline. The overall absolute improvement over the [11] baseline was 14% (from 75% to 89%); adding only ELMo yielded an improvement of 7.2%, while only adding the new training data improved the result by 10%. In the same paper, the authors measure the improvement achieved by their tagger on the dataset from the [2] paper. The improvement over the ML model in [2] was 1.3% when adding ELMo embeedings to the tagger, 0.3% when adding the challenge dataset as training data, while the combined effect Elmo+dataset was again an improvement of 1.3% (see table 1). This result suggests that CWEs can successfully be used to improve parsers' performance, and thus indirectly improve homograph disambigutation.

Table 1 reports the reults of the baselines described in [2] as well as the accuracy obtained by [9] on the same dataset. As mentioned above, the source of improvement introduced in that paper is twofold: training a POS tagger with additional 'challenging' data including Noun/Verb homographs (e.g. *impact*), and adding ELMo [10] embeddings to the pipeline. Virtually all of the reported improvement on the homograph disambiguation task stemmed from adding ELMo. This result strongly suggests that the use of CWEs is very beneficial for the disambiguation of homographs.

For our first experiment we created one model per each of the 138 homographs listed in [2] and fed the model with CWEs obtained from BERT and ALBERT pre-trained models of varying size. Both the training and test data were exactly the same as those used in [2]: for each homograph there are about 90 sentences in the training data and 10 in the test data. We did exclude the homograph *conglomerate* from our analysis, since there are no cases of verbal *conglomerate* in the training data, while there is one instance in the test data[6].

The results indicate that all the models we trained outperform both the ML model of [2] and the POS-tagger based solution in [9]; in the case of ALBERT-base embeddings, the result obtained outperforms both the ML and ML + Rules models reported in [2]. Models with BERT and ALBERT embeddings perform quite similarly to each other on this task. We leave a more careful analysis of the differences to future work.

Following a reviewer's suggestion, we carried out Leave-One-Out Cross-Validation for our BERT base model, given the very small size of our dataset. The results are very similar to those reported for the original split (98.5% accuracy with cross-validation vs 98.8 accuracy for original split).

For most of the remainder of the paper we will focus on BERT large, as we are independently carrying out additional

[2] https://pypi.org/project/BERT-embedding/

[3] https://huggingface.co/transformers/model_doc/albert.html

[4] http://nlpprogress.com/english/part-of-speech_tagging.html

[5] urlhttp://goo.gl/language/noun-verb

[6] This affects our numbers minimally: 10/10 cases in the training data feature the nominal variant, and so do 9/10 cases in the test set. We would thus have scored 9/10 on this homograph; however, given the verbal variant is not represented in the training data, it is simply impossible to learn anything about it, and we thus feel excluding this homograph is the correct solution

	Experiment 1	
System	Micro %	Macro %
[2] Rules	89.0	88.6
[2] ML	95.4	95.1
[2] ML + Rules	99.0	99.0
[9] POS + ELMo	96.7	96.7
Ours + BERT base	98.8	98.8
Ours + BERT large	98.7	98.7
Ours + ALBERT base	**99.1**	**99.1**
Ours + ALBERT large	98.3	98.3
Ours + ALBERT x-large	97.6	97.6
Ours + ALBERT xx-large	98.6	98.6
Ours + BERT base (leave-one-out)	98.5	98.5

Table 1: *Our models' accuracies vs. baselines for Exp. 1*

work on this model and the accuracy difference between this model and other BERT variants is rather small.

We conclude this section by observing that models appear to have learned significant cues from the surrounding context, beyond what provided by the immediately surrounding words. We for example get the correct result for the two sentences considered challenging in [2] ((1), (2) in Table 2). The challenge in sentence (1) is the sequence '*wind up*', which is frequently associated with a phrasal verb interpretation. In sentence (2) the sequence '*to present*' frequently appears as an infinitival verb. For both sentences our models assign almost 100% probability to the correct tagging. We report the implied probabilities as $softmax(logits)$ in table 2. As some sentences are genuinely ambiguous, we checked whether our models assign lower probabilities to such cases. This is indeed the case. Sentence (3) is ambiguous, as the verb *read* can be interpreted as present or past tense depending on the surrounding context. A survey conducted with 10 English native speakers confirms that the preferred interpretation of *read* in (3), when uttered in isolation, is overwhelmingly with the verb in the past tense: this ultimately has to do with English quite rigidly mandating the present progressive form to express continuous/progressive aspect (contrary to, say, Romance languages where present tense can be used to express continuous aspect). Our models correctly assign a mild preference to the VBD interpretation.

Sentence	% [Tag]	% [Tag]
(1) There may be **winds** up to 20 miles per hour	> 99.9 [N]	< 0.01 [V]
(2) Smith has played Trophy matches for the county from 1993 to **present**	> 99.9 [N]	< 0.01 [V]
(3) I **read** a book	63.52 [VBD]	36.48 [VB]

Table 2: *Challenging sentences and implied probability (data for model trained on BERT large embeddings)*

3.2. Experiment 2: Data augmentation for highly unbalanced train sets

The 19 homographs for which we do not obtain 100% accuracy on the test set (10 sentences) for the BERT large models are listed in Table 3.

Table 3 shows that 10 out of 19 homographs for which accuracy is less than 10/10 are homographs whose training set is deeply unbalanced, containing less than 10 cases for one of the two versions of the homograph word (column 'Split' in table 3). We selected 9 of the 10 cases (indicated by the light shading in the table) and manually selected additional training sentences from an internal dataset made up mostly by fiction materials. We added 50 additional sentences for each of these homographs, ensuring that we would include at least 10 sentences for the 'weak', under-represented version of the homograph.[7]. It is worth reiterating that the annotator selecting the additional sentences was not aware of the results obtained by our model, and thus the selection was completely unbiased (i.e. there was no overt attempt to fix the incorrect cases). The new splits obtained are included in the NewSplit column of table 3. A reviewer correctly points out that the numbers in the 'Split' column in table 1 does not always add up to 90. This is an issue with the original dataset described in [2], which we are adopting in this paper. In several cases, the train data does not contain 90 sentences, but fewer.

	Experiment 2 data			
Word	Split	OrigAcc	NewSplit	NewAcc
approximate	78/11	9/10		
compress	84/4	9/10	109/29	10/10
confines	73/16	9/10		
content	1/89	9/10		
correlate	7/83	8/10	17/123	10/10
duplicate	67/23	9/10		
escort	81/9	9/10	109/31	9/10
graduate	87/3	8/10	127/13	9/10
incline	86/3	9/10	126/13	10/10
increment	78/11	9/10		
insert	47/43	9/10		
intrigue	86/4	9/10	126/14	10/10
invalid	67/22	9/10		
invert	27/62	9/10		
laminate	87/3	9/10	126/14	10/10
pasty	63/22	8/9		
perfume	88/2	9/10	134/6	10/10
upset	65/24	9/10		
transplant	85/5	9/10	126/14	10/10
Total	**79/90**	**87.8%**	**88/90**	**97.8%**

Table 3: *Split, old and new accuracy for homographs with* $< 100\%$ *in Experiment 1*

	Experiment 2	
System	Micro %	Macro %
Ours + BERT large	**99.3**	**99.2**

Table 4: *Our model's overall accuracy for Exp. 2*

3.2.1. Results

We re-trained our classifier fed with BERT large embeddings with the new added materials described in section 3.2, in the same way as in Experiment 1. We report the new accuracy obtained in the NewAcc column in Table 3. For the nine homographs targeted the accuracy improved from 87.8% to 97.8%. Adding a small amount of extra labelled data to very unbalanced cases thus yielded an absolute improvement of 10% and a relative improvement of 11.38%.

[7] In cases for which we couldn't find 10 relevant cases in the first 1000 lines of the relevant corpus, we added however many 'weak' cases we found up to that point. We did not consider the homograph *content* because all 1000 cases at the top of our corpus featured nominal *content*

The overall accuracy of the our system built using BERT large embeddings improved to 99.3% and 99.2% (for micro and macro accuracy respectively), as shown in Table 4.

These results confirm that making the training set less unbalanced improves accuracy significantly. As the number of sentences added was just ten per target word, a new approach to fixing bugs related to homographs emerges: while relying on hand-crafted rules requires careful and skilled labour, attempting to fix a bug via data augmentation simply requires a native speaker of the language to select a handful of sentences matching the desired version of the homograph.

3.3. Experiment 3: Gradual sparsification of affine transform weights

Any online TTS system benefits from low latency. Experiment 3 investigates whether the size of our models can be reduced, thus improving the overall latency of our system. We aim at reducing the size of the model by applying gradual sparsification of affine transform weights during training.

The amount of non-zero weights was decreased throughout the training to 10% of the original amount, masking weights with the lowest amplitude every 100 training steps. This yielded a reduction of the size of the classifier for each homograph to 300 bytes, with minimal effect on the accuracy of the resulting model. Explicit sparsification required to decrease weight of *l2*-regularization to 0.001.

We applied sparsification to two models from Experiment 1: the models built on BERT base and ALBERT base. The results are displayed in 5.

	Experiment 3	
System	Micro %	Macro %
Ours + BERT base	98.8	98.8
Ours + BERT base (sparse)	98.8	98.8
Ours + ALBERT base	99.1	99.1
Ours + ALBERT base (sparse)	98.8	98.8

Table 5: *BERT base and ALBERT base with and without sparsification*

The results of this experiment show that sparsification introduces no accuracy loss in the case of BERT base, and a mild accuracy decrease (from 99.1% to 98.8%) in the case of ALBERT base. The size of the models obtained by way of sparsification is about 10 times smaller than the models described above for Experiment 1 (from 3kb per model to 300 bytes per model).

Model sparsification is thus a viable solution for online TTS systems where model size and latency are a concern.

4. Conclusions

This paper introduces a fully ML-based homograph disambiguation system based on contextual word embeddings. The proposed approach achieves SOTA results, without the need of any *ad-hoc* rules. This paves the way a fully automated approach to homograph disambiguation for TTS systems: the need for expensive, language-specific and hard to maintain rule systems becomes much less central. While this paper has only dealt with English, the availability of CWEs in many languages suggests that the simple approach described can be successfully extended cross-linguistically. We have further shown that balanced train datasets are crucial, as adding a few examples of the under-represented variant in unbalanced cases improved our accuracy by over 11% (relative). The method proposed, although performed manually, is effectively a special case of active-learning: relevant data is selected to increase the number of under-represented variants. Throughout the paper we keep in mind integration of proposed homograph disambiguation system into production environment: using light-weight ALBERT features, using pretrained CWE features that can be reused for other purposes within TTS framework, using sparsified per-homograph models.

5. References

[1] J. Hobbs, *Homophones and Homographs: An American Dictionary, 4th ed.* McFarland, Incorporated, Publishers, 2006. [Online]. Available: https://books.google.co.uk/books?id=vCUTBQAAQBAJ

[2] K. Gorman, G. Mazovetskiy, and V. Nikolaev, "Improving homograph disambiguation with supervised machine learning," in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*. Miyazaki, Japan: European Languages Resources Association (ELRA), May 2018. [Online]. Available: https://www.aclweb.org/anthology/L18-1215

[3] D. Yarowsky, "Homograph Disambiguation in Text-to-Speech Synthesis," in *Progress in Speech Synthesis*, J. P. H. van Santen, J. P. Olive, R. W. Sproat, and J. Hirschberg, Eds. New York, NY: Springer New York, 1997, pp. 157–172. [Online]. Available: http://link.springer.com/10.1007/978-1-4612-1894-4_12

[4] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186. [Online]. Available: https://www.aclweb.org/anthology/N19-1423

[5] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "ALBERT: A Lite BERT for Self-supervised Learning of Language Representations," *arXiv:1909.11942 [cs]*, Feb. 2020, arXiv: 1909.11942. [Online]. Available: http://arxiv.org/abs/1909.11942

[6] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "RoBERTa: A Robustly Optimized BERT Pretraining Approach," *arXiv:1907.11692 [cs]*, Jul. 2019, arXiv: 1907.11692. [Online]. Available: http://arxiv.org/abs/1907.11692

[7] I. Tenney, D. Das, and E. Pavlick, "BERT Rediscovers the Classical NLP Pipeline," *arXiv:1905.05950 [cs]*, Aug. 2019, arXiv: 1905.05950. [Online]. Available: http://arxiv.org/abs/1905.05950

[8] C. D. Manning, K. Clark, J. Hewitt, U. Khandelwal, and O. Levy, "Emergent linguistic structure in artificial neural networks trained by self-supervision," *Proceedings of the National Academy of Sciences*, 2020. [Online]. Available: https://www.pnas.org/content/early/2020/06/02/1907367117

[9] A. Elkahky, K. Webster, D. Andor, and E. Pitler, "A challenge set and methods for noun-verb ambiguity," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, Oct.-Nov. 2018, pp. 2562–2572. [Online]. Available: https://www.aclweb.org/anthology/D18-1277

[10] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," *CoRR*, vol. abs/1802.05365, 2018. [Online]. Available: http://arxiv.org/abs/1802.05365

[11] B. Bohnet, R. T. McDonald, G. Simões, D. Andor, E. Pitler, and J. Maynez, "Morphosyntactic tagging with a meta-bilstm model over context sensitive token encodings," *CoRR*, vol. abs/1805.08237, 2018. [Online]. Available: http://arxiv.org/abs/1805.08237

11th ISCA Speech Synthesis Workshop (SSW 11)
26-28 August 2021, Budapest, Hungary

Analysing Temporal Sensitivity of VQ-VAE Sub-Phone Codebooks

Jason Fong, Jennifer Williams, Simon King

Centre for Speech Technology Research, University of Edinburgh, UK
{jason.fong, j.williams, simon.king}@ed.ac.uk

Abstract

In this work we present an analysis of temporal sensitivity of VQ-VAE sub-phone token sequences. Previous work has demonstrated that VQ-VAE systems learn a type of sub-phone representation. However, a detailed examination of the representations themselves is currently lacking. We address this gap by exploring linguistic unit reorganisation. Our experiments show that sub-phone codebook sequences are temporally correlated enough to identify VQ codes that correspond to distinct linguistic units. We found that it is possible to extract VQ codes and re-arrange these linguistic units in a meaningful way (i.e. changing the word-order of a sentence). This work puts us one step closer to understanding how to modify pronunciations at a fine granularity, such as below the phone-level unit.

Index Terms: VQ-VAE, speech synthesis, representation learning

1. Introduction

Speech applications such as automatic speech recognition (ASR) and text-to-speech synthesis (TTS) have traditionally employed phones to describe speech. While they were originally conceived for rapid linguistic field transcription, they are now used as a representation of pronunciation. Phones omit much of the nuance that is inherent in human speech production. For example, phones do not represent the effects of co-articulation and are inadequate for capturing other connected speech effects such as partial vowel reduction or elision. There are established approaches to work around these limitations, including expanding the phone set to include allophones and syllabic consonants, or by constructing a set of application-specific context-dependent categories such as diphones (for TTS), or triphones / quinphones (for ASR). Until recently, these were the only viable choices for a discrete representation of speech in speech technology applications.

However, recent advances in neural modelling, notably self-supervised and semi-supervised techniques, offer an ability to learn speech representations which – by definition – *must* capture nuances of speech: the training objective is to be able to reconstruct speech [1, 2] or make contextual predictions [3, 4, 5]. Now with the aid of such learned, informationally-dense speech representations, we are in a position to rethink our approach to representing speech for applications such as TTS. Recent work has already shown that neural speech models learn semi-supervised representations that capture high-level linguistic aspects of speech from speech waveforms. These representations can greatly improve ASR with little data [5], and can also be used to generate speech [6, 7]. But no prior work has sought to use these representations to create new speech applications altogether.

In this paper we encode speech into a sequence of tokens drawn from a finite set of categories using a VQ-VAE model [2], then reconstruct it from that sequence of tokens. We wish to evaluate the adequacy of the token sequence as a representation of speech pronunciation. That evaluation takes form of concatenating token sequences to construct word sequences not seen in the training data. That is, we use "concatenative VQ-VAE synthesis" as a methodology for evaluating whether the learned inventory of categories would be a useful pronunciation representation for neural TTS.

The novelty of this work is in the manipulation of learned VQ token sequences. Our results bring us a step closer to nuanced control of speech pronunciation, beyond the capabilities of phone-based representations. The ability to manipulate and control speech using VQ tokens would open up a plethora of possible future applications, including accent modulation, targeted pronunciation feedback, voice actor performance post-production, and pronunciation control for TTS.

The main contribution of this work is measuring the extent to which VQ token sequences can be manipulated. It is desirable that the tokens correspond to speech in a monotonic and predictable manner. However, because the tokens are always learned as a temporal sequence from natural speech, they have an as-yet-unknown *temporal sensitivity*: they are not guaranteed to have a monotonic relationship to the acoustics, and they might be context- or even speaker-dependent. We offer what we believe to be the first demonstration that it is feasible to manipulate and exchange token sequences. We analyse increasingly challenging tasks, from copy synthesis to the production of novel speech by concatenating short phrases with matched vs. mismatched phonetic context and speaker identity

2. Related Work

In the work of [8] they compared how graphemes and phones affected the learned pronunciations in Tacotron sequence-to-sequence TTS. They found that the internal representations for graphemes and phones were consistent, suggesting it is possible to control pronunciations directly from graphemes. They also evaluated the representations externally to Tacotron. However, graphemes are a large unit and the corresponding neural embeddings may not be able to control nuances of pronunciation. In our work, we are modelling a smaller unit with VQ tokens which can provide much finer control for pronunciation over grapheme embeddings.

A growing area of interest involves methods to discover meaningful acoustic units from speech, often in an unsupervised manner, and then utilise them in downstream tasks. In [9], they explored Transformer VQ-VAE for zero-shot synthesis: generating speech without text or phone labels. They showed that the VQ-VAE architecture is well-suited to discover phone and sub-phone units and it is entirely self-supervised. High-quality speech can be synthesised directly from these small units. While this work is encouraging, they have not manipulated the sub-phone units directly, which is something that we explore.

Perhaps one of the best examples of how VQ tokens can be

10.21437/SSW.2021-40

applied to speech applications comes from the text-to-speech system called DiscreTalk [6]. In this work, the VQ tokens were generated with different down-sampling factors, which effectively controlled the duration of a single VQ token. If the down-sampling factor is large, then individual tokens in a sequence have a longer duration, and vice-versa. They trained a neural machine translation (NMT) system to predict a sequence of VQ tokens from text input, which is similar to grapheme-to-phoneme prediction in conventional TTS systems. The predicted VQ tokens were then used in TTS. They showed that tokens of longer duration facilitated learning TTS, but sometimes by sacrificing overall speech quality. While this finding is important, it is not clear how the size of the token affects more nuanced elements of speech such as co-articulation, or what other aspects of pronunciation could be optimized. Furthermore the VQ tokens predicted from text by their NMT model (and subsequently its resulting pronunciations) are fundamentally both unpredictable and uncontrollable. In this work we make the crucial first steps towards ascertaining whether VQ tokens specifically and neural speech representations more generally are a good candidate for controlling synthesised speech.

3. Data and Model

3.1. Data

We use VCTK [10] to both train our VQ-VAE model and generate samples for our listening test. Although it is a relatively small dataset (44 hours over 109 speakers), since the recordings are of high quality, and our WaveRNN decoder [11] is of sufficient model capacity, our resulting system is able to accurately generate speech for each of the VCTK speakers. VCTK contains voices from a variety of different ages and UK accents.

3.2. VQ-VAE Model

To discover discrete units into which speech can be encoded, then subsequently synthesised, we use a VQ-VAE model based on [2]. Our model differs from the original by using WaveRNN[1] as the vocoder instead of WaveNet [12], for faster training and inference.

The VQ-VAE encoder uses 10 1D convolutional layers to encode a sequence of waveform samples $\mathbf{x}_{1:T}$ into a sequence of 128-dim continuous latents $\mathbf{z}_{1:U}$. These latents are then discretised using a vector quantisation layer to create a sequence of code-words (i.e., codebook entries) $\mathbf{d}_{1:U}$, which henceforth we will call **tokens**. Our codebook contains 512 128-dim entries. We do not examine the effect of varying the codebook size, leaving it for future work. A WaveRNN decoder produces waveform samples at 22.05 kHz sample rate, conditioned on the sequence of tokens and a single speaker one-hot vector that is broadcast across all timesteps $\mathbf{s}_{1:U}$.

We train our model using all VCTK speakers for 1000 epochs (2.7 million iterations), which takes approximately 1 week on a single NVIDIA 2080Ti GPU. Since the focus of this study is to explore concatenative synthesis and *not* to examine VQ-VAE's ability to generalise to unseen speakers, we choose to train the model on all of VCTK. Thus we perform concatenative VQ-VAE synthesis from parts of training utterances, just as would be the case in waveform-domain concatenation [13].

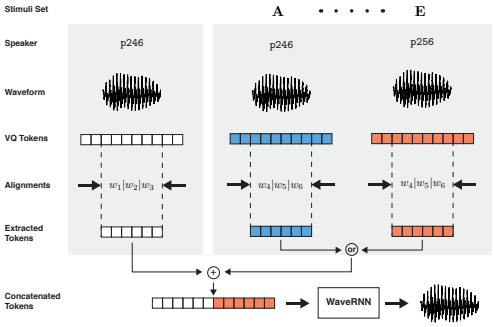


Figure 1: *Overview of the concatenative VQ-VAE synthesis method. The token sequence corresponding to word sequence w_1, w_2, w_3 is being concatenated with the token sequence corresponding to w_4, w_5, w_6, from which a waveform is synthesised using WaveRNN. Pictured is a set* **E** *stimuli being generated for the* `p246+p256` *speaker combination. The particular sub-sequence of words w_4, w_5, w_6 used depends on the stimuli set currently being generated, either* **A**, **B**, **C**, **D**, *or* **E**.

4. Method

We describe our method for re-arranging the VQ token subsequences that correspond to 3-word chunks. What is most important is that we find a way to meaningfully manipulate speech in the VQ token domain. Here, we work at the word-level because it is a first approximation to being able to manipulate VQ tokens at a lower level such as below the phone-level. As we have described, a successful outcome at the word-level signifies that VQ tokens may be a feasible representation for controlling pronunciation.

Figure 1[2] shows an overview of the concatenative VQ-VAE synthesis pipeline that extracts tokens corresponding to words in two separate utterances, these tokens are then used to generate a single listening test stimuli. The extraction process for one utterance works as follows: we first encode the audio of one utterance U_1 and encode it into a stream of tokens using the VQ-VAE encoder. We then use the timestamps for a sub-sequence of words (e.g. w_1, w_2, w_3) to identify and extract their corresponding subset of VQ-VAE tokens $\mathbf{d}_{left}$. We then repeat this process for a second utterance U_2 to get another subset of tokens $\mathbf{d}_{right}$ corresponding to w_4, w_5, w_6. We then concatenate them together to get $\mathbf{d}_{left} \bigoplus \mathbf{d}_{right}$, and use them to condition the WaveRNN to generate speech that resembles $w_1, w_2, w_3 \bigoplus w_4, w_5, w_6$. Note that we use '$\bigoplus$' to signify the concatenation point between sequences of words or tokens.

Word-level alignments are required to find the correspondence between tokens and words, and are found using the Montreal forced aligner [14]. We do not perform any analysis or adjustment of alignments here, so they are a potential source of error.

We also found that the generation of words corresponding to the start of $\mathbf{d}_{left}$ and the end of $\mathbf{d}_{right}$ were sometimes cut-off or not realised. Subsequently we experimented with inserting $\mathbf{d}_{sil}$ before $\mathbf{d}_{left}$ and after $\mathbf{d}_{right}$, finding that it helps recover some words. Examples can be found on our sample

[1] https://github.com/mkotha/WaveRNN

[2] Credit to Christine Wan for help with this diagram.

page[3]. When generating our listening test stimuli we pad by 50 timesteps of $\mathbf{d}_{sil}$ on both sides. We also chose to use 6-word stimuli rather than 4-word ones so that the words adjacent to those at the concatenation point are not cut off.

5. Experiments

5.1. Word Transcription Task

We used a fill-in-the-blank transcription task. We generated 6-word stimuli of the form $w_1, w_2, w_3 \bigoplus w_4, w_5, w_6$. For each stimulus, participants were asked to transcribe the word *after* the concatenation point (i.e., w_4) by being presented with the transcription *minus* the word-in-question. For example, for the sentence 'red and green $\bigoplus$ looking any further' using $\mathbf{d}_{left}$ corresponding to 'red and green' and $\mathbf{d}_{right}$ to 'looking any further', participants were presented with the transcription 'red and green <blank> any further'.

5.2. Experimental Conditions

Our listening test contains 5 sets each with 40 stimuli, so that each participant rates the same 200 stimuli:

- **A**: Copy-synthesis
- **B**: Matched context + Matched Speaker
- **C**: Matched context + Mismatched Speaker
- **D**: Mismatched context + Matched Speaker
- **E**: Mismatched context + Mismatched Speaker

These sets are designed to help us answer the following three questions: Does concatenative synthesis result in less intelligible speech than copy-synthesis (Set **A** vs. Sets **B**, **C**, **D**, **E**)? Is intelligibility affected by extracting tokens from audio spoken by two different speakers (Sets **B**, **D** vs. Sets **C**, **E**)? Is intelligibility affected when the two words adjacent to the concatenation boundary, w_3 and w_4, are chosen so that their linguistic contexts, according to their surrounding words in their original sentences, mismatch (Sets **B**, **C** vs. Sets **D**, **E**)?

We determine whether linguistic context matches between two words by comparing their adjacent triphones. For example if w_3 is 'hello' (HH EH L OW) and w_4 is 'world' (W ER L D), then we compare the rightmost triphone of 'hello' to the leftmost one of 'world'. Given that the rightmost triphone of 'hello' is L OW W, if the leftmost one of 'world' is OW W ER then we consider it a matching linguistic context, and if it were AH W ER then we consider it mismatching. When choosing a w_4, w_5, w_6 for a given w_1, w_2, w_3 we make sure not to choose those which contain a w_4 that is either a stopword or a word that has been generated before for a particular speaker.

5.3. Materials

We used 40 unique sequences w_1, w_2, w_3 each of which could be followed by one of 5 unique sequences of words w_4, w_5, w_6 (thereby creating 200 unique sentences in total, described in 5.2). Each unique sequence w_1, w_2, w_3 was presented 5 times during the listening test (once per stimuli set), and each time is coupled with a unique w_4, w_5, w_6, making a total of 200 sentences, noting that these may not all be grammatically-correct.

[3] https://jonojace.github.io/SSW21-concatenative-vqvae

Table 1: *Speaking rate information (average number of seconds per phone).*

Duration Type	p246	p256	p345	p374
non-sil Phone	0.088	0.085	0.083	0.118
sil Phone	0.114	0.106	0.278	0.278

5.4. Speakers

We took care in choosing the speakers to build our stimuli. We found in preliminary experiments that conditioning the WaveRNN using tokens extracted from slower-speaking voices resulted in more intelligible speech, when performing either copy-synthesis or concatenative reconstructions. Subsequently we chose 4 slow-speaking voice talents for our experiments; p246, p256, p345, and p374 whose speaking rates are summarised in Table 1.

We generate 50 stimuli from each of our 4 speakers, made up of 10 stimuli from each stimuli set. For example, if the main speaker is p246 and the secondary speaker is p256 (for answering mismatched speakers question) then we will generate 10 stimuli for **A** **B** and **D** using tokens only from p246, and 10 stimuli each for **C** and **E** using $\mathbf{d}_{left}$ from p246 and $\mathbf{d}_{right}$ from p256. The 4 main and secondary speaker combinations that we use are p246+p256, p256+p345, p345+p374, and p374+p246. To condition the WaveRNN we always use the main speaker to condition $\mathbf{d}_{left} \bigoplus \mathbf{d}_{right}$ even if the original speaker of $\mathbf{d}_{right}$ is the secondary speaker, therefore our model performs voice conversion on the latter halves of the stimuli in sets **C** and **E**.

5.5. Listening Test

We built our listening test on the Qualtrics[4] platform [5], and recruited participants using Prolific[6]. Using the following filters we recruited 50 participants, each of whom are from the UK, have no literacy difficulties, and have at least a 90% approval rating on Prolific. The order of the 200 stimuli are randomised on a per participant basis.

In order to ensure that our results were accurate we performed a manual check of all participants' answers to correct misspellings (e.g. contenders and contendors), typos (e.g. fresh and frsh), and homophone ambiguity errors (e.g. rode, rowed, and road).

6. Results

6.1. Stimuli Set Comparisons

We present results from our intelligibility test partitioned across each stimuli set in Figure 2. We find that the copy-synthesis stimuli (**A**) are the most intelligible. This is likely because no concatenative synthesis is performed, and as such the resulting token sequences will not suffer from potential alignment errors and will appear 'natural' to the WaveRNN.

The results of **B** and **C** show that concatenative VQ-VAE synthesis can produce intelligible speech reliably, additionally since they outperform **D** and **E** we can conclude that concatenative synthesis works better when linguistic contexts match.

[4] https://www.qualtrics.com/uk/

[5] Test building automated using https://github.com/CSTR-Edinburgh/qualtreats

[6] https://www.prolific.co/

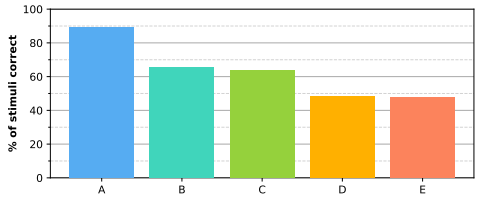


Figure 2: *Intelligibility results across the stimuli sets* **A**, **B**, **C**, **D**, **E** *described in Subsection 5.2. We present the percentage of stimuli within a set answered correctly by participants.*

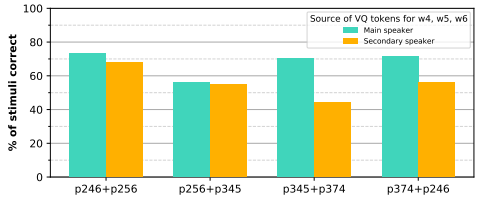


Figure 3: *Intelligibility results for the 4 speaker combinations described in Subsection 5.1. For each speaker combination we present the percentage of stimuli correct when both* $\mathbf{d}_{left}$ *and* $\mathbf{d}_{right}$ *are extracted from the main speaker's speech (left-side teal coloured bars) and when* $\mathbf{d}_{left}$ *is extracted from the main speaker and* $\mathbf{d}_{right}$ *is extracted from the secondary speaker (right-side orange coloured bars). Note that due to the copy-synthesis set* **A***, the total number of stimuli when speakers are matched and when speakers are mismatched differ, being 120 and 80 respectively.*

Comparing the results of **B** vs **C** and **D** vs **E** we find that synthesising using concatenated token sub-sequences extracted from different speakers has only a small negative on intelligibility. The closeness of these results combined with our observations that the speaker identity of samples do not change mid sentence are testament to the ability of VQ-VAE to learn codebook embeddings that are disentangled from speaker identity. Disentanglement is achieved due to the use of speaker inputs to the decoder and the extreme bottle-necking of the input signal in both the time and feature dimensions (resulting in a low bit-rate encoding). These results are promising for future concatenative neural synthesis work: it is clearly possible to mix and match VQ tokens between different speakers. This could enable new applications such as correcting a system's pronunciations via cheap-to-obtain speech exemplars rather than more expensive phonetic transcriptions.

6.2. Speaker Combination Experiments

Figure 3 shows our intelligibility results partitioned across the 4 speakers. We observe three findings: First, that mismatched speakers across the concatenation boundary results in lower intelligibility in general. Second, that mismatched vs matched speaker intelligibility differs between speaker combinations, e.g. the difference for `p256+p345` is very small (1.9% increase), whereas the difference is larger for `p345+p374` (59% increase). Upon reflection of the speaking rates in Table 1 we do not find a strong correlation between speaking rate and intelligibility, subsequently differences in intelligibility could be due to other factors such as per speaker co-articulation habits or accent differences for example. Third, we find that certain speakers are less intelligible in general, speaker `p256` for example. We further investigated this by visualising the total duration of silences for each speaker in Figure 4 where we see that speaker `p256` has the smallest amount of total silence, which may make interword boundaries harder to identify for both the VQ-VAE encoder and forced aligner. Additionally the speaker's heavier more informal accent may also have an effect on intelligibility.

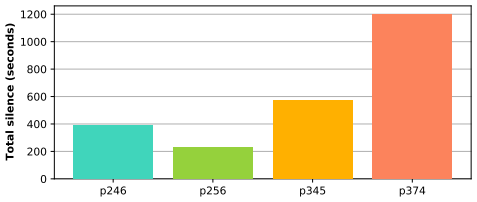


Figure 4: *Total duration of all silence phones for each speaker.*

7. Qualitative Analysis

In this section we present our findings regarding the types of errors that participants made. We are particularly interested in the cases of incorrect transcription because it helps us better understand how well we can manipulate VQ tokens at a finer-granularity. One of our hypotheses was that concatenating $\mathbf{d}_{left}$ and $\mathbf{d}_{right}$ together would mainly cause the sounds adjacent to the boundary to be affected. Surprisingly however we found that there are instances where the first phone of w_4 was correctly heard, but the rest of the word was largely intelligible, causing participants to hallucinate an incorrect answer. Since we did not discover a cause for this phenomena we include examples on our samples page and leave further investigation to future work.

8. Conclusion

In this work we present an analysis of the sensitivity of VQ-VAE tokens to their surrounding context by using concatenations of tokens extracted from disparate sentences to decode audio. We primarily find that 'unit selection' speech generation is possible in the discrete latent space. Furthermore by extracting tokens from sentences selected from a variety of specific conditions we discover that VQ-VAE tokens are temporally highly linguistic context dependent, but not speaker context dependent. Together these two results are promising for future speech systems as they suggest that readily available audio exemplars can be used to alter aspects of speech such as pronunciation without resorting to expensive hand-transcribed labels such as phonetic transcriptions. We further observe that within our pipeline speaking rate and duration of silences can affect downstream reconstruction intelligibility. In future work we plan to investigate neural concatenative synthesis cross-lingually, make tokens less context dependent without sacrificing reconstruction quality, and remove the reliance of our system on pretrained forced aligners and instead use word-level alignments obtained in an unsupervised fashion.

9. Acknowledgements

This work was partially supported by the EPSRC Centre for Doctoral Training in Data Science, funded by the UK Engineering and Physical Sciences Research Council (grant EP/L016427/1) and University of Edinburgh.

10. References

[1] W.-N. Hsu, Y. Zhang, and J. Glass, "Unsupervised learning of disentangled and interpretable representations from sequential data," *arXiv preprint arXiv:1709.07902*, 2017.

[2] A. v. d. Oord, O. Vinyals, and K. Kavukcuoglu, "Neural discrete representation learning," *arXiv preprint arXiv:1711.00937*, 2017.

[3] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.

[4] Y.-A. Chung and J. Glass, "Generative pre-training for speech with autoregressive predictive coding," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 3497–3501.

[5] A. Baevski, H. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *arXiv preprint arXiv:2006.11477*, 2020.

[6] T. Hayashi and S. Watanabe, "Discretalk: Text-to-speech as a machine translation problem," *arXiv preprint arXiv:2005.05525*, 2020.

[7] A. Polyak, Y. Adi, J. Copet, E. Kharitonov, K. Lakhotia, W.-N. Hsu, A. Mohamed, and E. Dupoux, "Speech resynthesis from discrete disentangled self-supervised representations," *arXiv preprint arXiv:2104.00355*, 2021.

[8] A. Perquin, E. Cooper, and J. Yamagishi, "An investigation of the relation between grapheme embeddings and pronunciation for tacotron-based systems," 2021.

[9] A. Tjandra, S. Sakti, and S. Nakamura, "Transformer vq-vae for unsupervised unit discovery and speech synthesis: Zerospeech 2020 challenge," 2020.

[10] J. Yamagishi, C. Veaux, K. MacDonald *et al.*, "CSTR VCTK Corpus: English Multi-Speaker Corpus for CSTR Voice Cloning Toolkit (version 0.92)," 2019.

[11] N. Kalchbrenner, E. Elsen, K. Simonyan, S. Noury, N. Casagrande, E. Lockhart, F. Stimberg, A. Oord, S. Dieleman, and K. Kavukcuoglu, "Efficient neural audio synthesis," in *International Conference on Machine Learning*. PMLR, 2018, pp. 2410–2419.

[12] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[13] A. J. Hunt and A. W. Black, "Unit selection in a concatenative speech synthesis system using a large speech database," in *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, vol. 1. IEEE, 1996, pp. 373–376.

[14] M. McAuliffe, M. Socolof, S. Mihuc, M. Wagner, and M. Sonderegger, "Montreal forced aligner: Trainable text-speech alignment using kaldi." in *Interspeech*, vol. 2017, 2017, pp. 498–502.

INDEX