FOREIGN ACCENT CONVERSION WITH NEURAL ACOUSTIC MODELING

A Dissertation

by

GUANLONG ZHAO

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	Ricardo Gutierrez-Osuna
Committee Members,	Yoonsuck Choe
	Ruihong Huang
	Jyotsna Vaid
Head of Department,	Scott Schaefer

August 2020

Major Subject: Computer Science

ABSTRACT

Foreign accent conversion (FAC) aims to generate a synthetic voice that has the voice identity of a given non-native speaker (NNS), but the pronunciation patterns (i.e., accent) of a native speaker (NS). This synthetic voice is often referred to as "Golden Speaker" in the computer-assisted pronunciation training literature. Prior FAC algorithms do not fully remove mispronunciations in the original non-native speech or fully capture the voice quality of the non-native speaker. More importantly, most prior methods require a reference utterance from a native speaker at synthesis time, thus limiting the application scope of FAC in pronunciation training. This dissertation aims to address these issues by proposing solutions to three interrelated problems:

- Reducing mispronunciation in the accent converted speech
- Improving the voice similarity between the accent conversions and the NNS
- Removing the need for an NS reference utterance at synthesis time

To address the first problem, I propose an approach that matches frames from the native reference speaker and non-native speakers based on their phonetic similarity. To generate accent conversions, I then use the paired frames to train a Gaussian Mixture Model (GMM) that converts the native reference utterance to match the voice identity of the non-native speaker. The algorithm outperforms earlier methods that match frames based on Dynamic Time Warping or acoustic similarity, improving ratings of acoustic quality and native accent while retaining the voice identity of the non-native speaker. I

also show that this approach can be applied to non-parallel training data and achieve comparable performance.

To address the second problem, I develop a sequence-to-sequence speech synthesizer that maps speech embeddings (e.g., phonetic posteriorgrams) from the non-native speaker into the corresponding spectrograms. At inference time, I drive the synthesizer with a speech embedding from an NS reference utterance. The proposed system produces speech that sounds clearer, and more natural and similar to the non-native speaker compared with the model presented in the first work, while significantly reducing the perceived accentedness compared with non-native utterances.

To address the third and final problem, I present a reference-free FAC system. First, I generate a synthetic golden speaker for the non-native speaker using the method proposed in the second work. Then, I train a pronunciation-correction model that maps the non-native speaker utterance into the synthetic golden speaker utterance. Both objective and subjective evaluations show that the reference-free FAC model generates speech that resembles the non-native speaker's voice while being significantly less accented.

In the process of conducting this research, I also took a leading role in collecting, curating, and releasing a non-native speech corpus named L2-ARCTIC, which is the first open-source corpus of its kind and provides valuable resources for the speech community. I include descriptions of the curation process, data analysis, and applications of the corpus in this dissertation.

DEDICATION

To my parents

ACKNOWLEDGEMENTS

First and foremost, I would like to express my sincere gratitude to my advisor Dr. Ricardo Gutierrez-Osuna. He is an enthusiastic and resourceful scholar, and he taught me how to perform proper research and become a better researcher. This work would never be possible without his guidance and support. I also much appreciate the encouragement and feedback from my dissertation committee members Dr. Yoonsuck Choe, Dr. Ruihong Huang, and Dr. Jyotsna Vaid.

I would like to thank my wonderful collaborators at the Iowa State University, Dr. John Levis, Dr. Evgeny Chukharev-Hudilainen, Dr. Sinem Sonsaat, Alif Silpachai, Ivana Lučić Rehman, and Dr. Taylor Anne Barriuso, for their help with data collection, annotation, and interpretation.

I treasure the fellowship and friendship with my amazing colleagues Sandesh, Jin, Avinash, Tian, Difan, Genna, Tianlong, Akhil, Purvesh, Roger, María, Adam, Chris, Dennis, Shaojin, Anurag, Sudip, and Nitin. Thank you so much for creating a collaborative, supportive, and joyful work environment. It is my honor to have the chance to work with so many talented people.

Thanks also go to my support network for their emotional support. I would like to offer my special thanks to my buddies Zelun, Ruohuang, and Han, for keeping my morale high throughout this Ph.D. program.

I wish to acknowledge the help provided by 362 anonymous human participants from Amazon Mechanical Turk. They contributed valuable data to this research.

This manuscript was written during the hardship of the COVID-19 global pandemic, and I am genuinely grateful for everyone who fought in this tough battle and kept our lives safe and healthy.

I would like to thank my parents, Mr. Benyuan Zhao and Mrs. Yun Tao, for their unconditional love and support. I also would like to thank my mysterious significant other, who remained kamikakushi (a Japanese phrase for “hide by god”/ “spirited away”) during the five years, which helped me devote my full energy to this dissertation research. Lastly, I would like to thank myself for having the courage and devotion to finish this long journey.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

The dissertation committee members are Dr. Ricardo Gutierrez-Osuna (chair), Dr. Yoonsuck Choe, Dr. Ruihong Huang, and Dr. Jyotsna Vaid. Other people who contributed to this dissertation research include Shaojin Ding, Christopher Liberatore, Dr. Sinem Sonsaat, Alif Silpachai, Ivana Lučić Rehman, Dr. Evgeny Chukharev-Hudilainen, Dr. Taylor Anne Barriuso, and Dr. John Levis.

Funding Sources

This dissertation was supported by two National Science Foundation research grants (1619212 and 1623750), one research grant from Facebook Inc., one research grant from Qatar National Research Fund (NPRP8-293-2-124), and a teaching assistantship from the Department of Computer Science & Engineering. I also received two travel grants from the Office of Graduate and Professional Studies and the Department of Computer Science & Engineering, which supported my trips to ICASSP 2017 (New Orleans, Louisiana) and Interspeech 2019 (Graz, Austria). The early results of this dissertation were presented at these conferences. The opinions expressed herein are solely the author's and do not necessarily reflect the views of the funding sources.

TABLE OF CONTENTS

Page

LIST OF FIGURES

Page

LIST OF TABLES

Page

1. INTRODUCTION

Adult learners of a second language (L2) often speak with a foreign accent. This is a result of multiple social and linguistic factors, which include the age of L2 learning, length of residence in an L2-speaking country, gender, education level, and the learner's native language's (L1) transfer effect [1-3]. Although a foreign accent does not necessarily reduce the comprehensibility or intelligibility of the non-native speech [4], by improving pronunciation, L2 learners interacting with native speakers (e.g., immigrants, foreign employees) have much to gain in the workplace, career opportunities, social life, and education [5-9].

In-person pronunciation coaching is an effective way to improve a learner's pronunciation but is often expensive and inaccessible for most learners. As such, listening and repeating after a pre-recorded native teacher's reference speech has been a widely adopted and affordable alternative. Students can identify potential mispronunciations in their production by comparing their speech with a teacher's utterance that has the same linguistic content, and then repeat after the teacher's speech to resolve these issues. Several studies [10, 11] have suggested that having a suitable native speaker to imitate – a so-called *golden speaker* can be beneficial in pronunciation training. Felps et al. [12] suggested that each learner's *golden speaker* should be their voice, resynthesized to have a native accent, and that this synthetic voice could be created by ***foreign accent conversion*** (FAC).

Formally, foreign accent conversion aims to create an artificial voice that has the voice identity of a non-native speaker but the pronunciation characteristics (e.g., prosody, segmentals) of a native reference speaker. Multiple solutions have been proposed for accent conversion, including voice morphing [12-15], acoustic-similarity based frame pairing [16], and articulatory synthesis [17-20]. Although they succeed in generating accent-reduced syntheses, they have various limitations. The voice morphing methods cannot preserve the non-native speaker's voice identity, resulting in a voice that sounds like a "third-speaker" who is neither the non-native nor the native speaker. The acoustic-similarity pairing method creates a lookup table between the native and non-native speech frames by minimizing their acoustic distance, which is measured by the Euclidean distance in the Mel-Frequency Cepstral Coefficient (MFCC) feature space. The pairing method then uses the resulting frame pairs to train a voice conversion model [21] that maps the spectral features from a native reference utterance to match the non-native speaker's identity. This method can synthesize speech that resembles the non-native speaker's voice, but it retains segmental mispronunciations that are introduced in the frame pairing process. Finally, articulatory synthesis methods need specialized apparatus to collect articulation data and often involve challenging recording conditions[1]. Therefore, they are not practical for daily use or frequent training. Moreover, to correct the mispronunciations in a non-native *test* utterance, all these methods require a pre-recorded native reference utterance,

[1] For example, recording the electromagnetic articulography (EMA) [22] requires placing sensor coils on the tongue and other articulators to measure their position and movement over time during speech production. Recording the real-time magnetic resonance imaging (MRI) of the articulations [23] requires expensive MRI machines, and they tend to produce a loud background noise, which is both uncomfortable for the participants and makes the audio recordings noisy.

which significantly limits their applications in real-world scenarios. *Therefore, this dissertation intends to resolve the issues mentioned above with previous foreign accent conversion methods.*

The key to FAC is to map the raw speech signal into an intermediate feature space that separates the linguistic and phonetic information from voice identity, such that we can perform pronunciation modification in that feature space without interfering with the voice identity. Following this idea, speech embeddings produced by an acoustic model (AM) in an automated speech recognizer (ASR) become an ideal candidate for such a feature space. A speech embedding is the output of a selected layer (generally one of the last few layers) in a neural network-based AM. AMs that are trained on a large corpus with many native speakers generate speech embeddings that are speaker-independent while representing the linguistic and phonetic information.

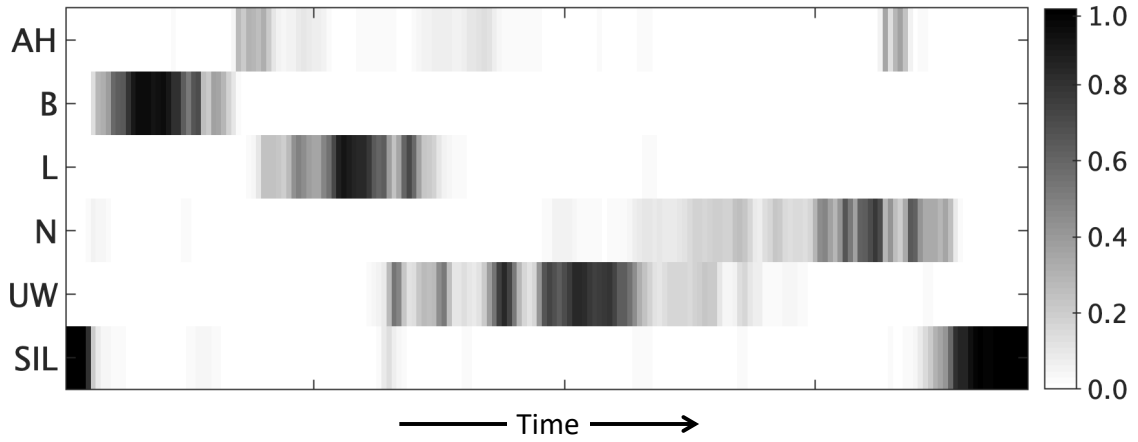

Figure 1.1: PPG of a spoken word *balloon*, whose pronunciation is "B AH L UW N" in the ARPAbet phoneme set. "SIL" means silence. An American English speaker uttered this word.

One representative speech embedding is the phonetic posteriorgram (PPG) [24]. Figure 1.1 illustrates an example of the PPG of a spoken word. A PPG is computed by segmenting speech into frames and calculating the posterior probability that each frame belongs to a set of pre-defined phonetic units (e.g., phonemes or triphones/senones). Generally, the PPG is the output of the final softmax layer in the acoustic model. Alternatively, one can also use the hidden layer outputs (typically referred to as the bottleneck features) of the acoustic model as the speech embeddings.

In the first part of this dissertation (Chapters 3, 4, and 5), I propose three novel FAC systems that utilize speech embeddings to address the limitations faced by prior FAC methods. In the second part of this dissertation (Chapter 6), I describe the FAC corpus that my collaborators and I collected and released during this dissertation research, which is the first open-source corpus for the accent conversion task.

In the first work (Chapter 3), I address the residual mispronunciation issue of previous frame pairing-based FAC methods [16]. Instead of performing frame-pairing using Dynamic Time Warping (DTW) or acoustic similarity, the proposed method uses the phonetic similarity between PPGs to measure pronunciation differences between native and non-native speech frames. Performing frame pairing in the PPG space can reduce mismatches between speech frames and improve the nativeness and acoustic quality in the converted speech.

In the second work (Chapter 4), I use a state-of-the-art speech synthesizer to improve the voice similarity between the accent converted speech and the original non-native speech. Prior FAC methods need to borrow the excitation signal from the native reference

utterance, and then use a signal processing-based vocoder to combine the excitation signal with the converted spectral features to generate the audio waveform. Therefore, the output speech is diluted with voice identity cues from the native reference speaker and thus does not fully capture the voice individuality of the non-native speaker. In this work, I propose an end-to-end speech synthesizer that directly maps PPGs from the non-native speaker to their corresponding audio waveform. Then, I drive the speech synthesizer with speaker-independent PPGs from a native reference utterance to produce the accent conversion. The end-to-end speech synthesizer is constructed with a sequence-to-sequence conversion model that converts between PPGs and mel-spectrograms, and a neural vocoder that directly recovers the speech waveform from mel-spectrograms. Therefore, one no longer needs the reference utterance's excitation signal to generate the audio, improving the voice identity of the accent conversion.

In the third work (Chapter 5), I propose a two-step solution to eliminate the requirement of the native reference speech at *inference*/*test time* (i.e., runtime synthesis). In the first step, I train the speech synthesizer proposed in Chapter 4 on a dataset of utterances from the non-native speaker. Then, I generate a synthetic golden speaker by driving the speech synthesizer with speech embeddings from *training* native reference utterances. The resulting synthetic golden speaker has the voice identity of the non-native speaker (provided by the speech synthesizer) and the pronunciation and prosody of the native reference speaker. In the second step, I train a sequence-to-sequence [25] pronunciation correction model to *directly* map non-native speech to the synthetic golden speaker's speech. Sequence-to-sequence models have shown promising results across multiple domains [26-

28], and they can exploit the context-dependent nature of pronunciation errors in non-native speech. The outputs of the sequence-to-sequence model contain the linguistic content of the input non-native speech but with the synthetic golden speaker's native pronunciation patterns. In the testing phase, foreign accent conversion can be produced by passing the non-native speech through the pronunciation correction model.

The fourth and last part of this dissertation (Chapter 6) focuses on providing resources for foreign accent conversion. With the assistance from my collaborators, I curated an open-source non-native English speech corpus, which includes high-quality recordings as well as annotations on segmental mispronunciations from a diverse group of non-native English speakers with six different first languages (Hindi, Korean, Mandarin, Spanish, Vietnamese, and Arabic). Since there was no existing corpus like this, having such a corpus would positively promote future research on FAC methods for computer-assisted pronunciation training.

In summary, this dissertation research consists of <u>four main objectives</u>:

(1) **PPG frame pairing**: Develop an accent conversion system using a frame-pairing method based on phonetic similarity to improve the nativeness of the speech syntheses

(2) **Accent conversion using the sequence-to-sequence model**: Develop an accent conversion model using state-of-the-art sequence-to-sequence speech synthesizers for better speaker individuality

(3) **Reference-free accent conversion**: Develop an accent conversion algorithm that does not need a reference utterance at synthesis time

(4) **Speech corpus**: Build and release a high-quality and diverse non-native English speech corpus

This dissertation research has the following major contributions. Objective (1) improves the nativeness of accent conversion and eliminates the requirement of using parallel training data. Objectives (2) and (3) improve voice similarity and acoustic quality significantly. More importantly, they eliminate the need for native reference utterances at the synthesis time, and thus achieve end-to-end accent conversion. In essence, the non-native accent in the input speech signal is reduced directly with a single system. Objective (4) provides valuable resources for the future development of accent conversion algorithms.

The works presented in this dissertation were submitted to or published in top-tier peer-reviewed venues. Initial results from Objective (1) were published at the *2018 International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*; a thorough examination of the proposed method was published by the *IEEE/ACM Transactions on Audio, Speech, and Language Processing (TASLP)*. Results from Objective (2) were published at *Interspeech 2019*. A detailed description of the findings in Objective (3) is being submitted to *TASLP*. A paper introducing the corpus built in Objective (4) was published at *Interspeech 2018*, and the corpus is openly accessible online.

The rest of this dissertation is organized as follows. Chapter 2 summarizes the necessary background knowledge for this work. Chapters 3, 4, 5, and 6 introduce the works from the four main objectives, respectively. Chapter 7 concludes this dissertation work and discusses possible future research directions. I include Appendix A to list the related publications, Appendix B to document an open-source foreign accent conversion

tool developed during this dissertation research, and Appendix C to cover model-building strategies I adopt in this work.

2. BACKGROUND

The speech signal, which emerges from a speaker's mouth, nose, and cheeks, is a longitudinal pressure wave that is formed of compressions and rarefactions of air molecules [29]. Microphones capture and convert the fluctuating air pressure into electrical signals, which are then quantized into discrete values for further digital signal processing.

The primary information carried by a speech signal is linguistic (e.g., the words spoken). This information is then convolved with the speaker's voice identity, regional accent/dialect, and emotions to produce a speech waveform. Speech modification techniques parameterize different attributes of the speech signal and manipulate them individually. In this dissertation, I focus on the modification of a speaker's pronunciation patterns (or accent, for short). More specifically, I seek to modify a non-native speaker's utterance to match a native teacher's accent. With this in mind, I first review related concepts in non-native speech. Second, I introduce the fundamentals of speech signal analysis and synthesis, such as common speech parameterizations and vocoders. Lastly, I summarize the basic ideas of the encoder-decoder (sequence-to-sequence) paradigm, which we use extensively in this dissertation work.

2.1. Non-native accents

Moyer [30] defines an accent as "a set of dynamic segmental and suprasegmental habits that convey linguistic meanings along with social and situational affiliations." This definition applies to both native and non-native speakers. Language learners who start to learn a second language (L2) can rarely acquire native-like accents after a certain age –

the so-called “critical period” [31, 32]. A foreign accent can be viewed as the systematic deviation from the standard norm of a spoken language. The deviations can be observed in the substitution, deletion, or insertion of phones, differences in intonation, syllable/word/sentence stress, or even the choice of vocabulary and syntax. All these deviations emphasize that the salience of a foreign accent can be reflected at multiple levels.

The factors that affect the degree of foreign accents have been a long-standing research question in the linguistic and language acquisition community [1]. Scovel [32] and Lenneberg [33] suggest that passing beyond the critical period (suspected of running before the age of puberty), the brain loses its plasticity, and some neurofunctional reorganizations occur during the development of the brain. As a result, it becomes harder for adult second language learners to distinguish between new phones, making it more difficult for them to produce the correct pronunciations. Prior research works give various endpoints of the critical period. For example, Long [34] suggested the age of six years old, while Scovel [35] suggested the age of 12 years, and Patkowski [36] extended the endpoint to 15 years old.

Alternatively, Oyama, Flege, and Bialystok, among others [37-39], suggest that age-related changes in the degree of foreign accent are results of the nature and the extent of the interaction between a language learner's native language and L2 systems. According to this line of reasoning, age is an index of the state of development of the native language system. The more fully developed the native language system is when the L2 acquisition happens, the more strongly the native language will influence the L2. Consequently, the

differences in the phonetic inventory between the learner's primary and the second languages have a strong influence on a learner's foreign accent [40-42].

As an example, I discuss how the phonological differences between Chinese (Pu Tong Hua) and English manifest into the common characteristics of Chinese-accented English. English has around 15 vowels [43], while Chinese only has around five vowels [44]. English vowels such as /æ/, /ɑu/, and /ɛə/ do not exist in the Chinese vowel set. Therefore, a Chinese English learner has to learn these new sounds without the luxury of a reference from their native language. Even when a vowel does exist in both languages, the vowel's place and manner of articulation might be different. One classic example is the Chinese vowel /ɪ/ and English vowels /ɪː/ and /ɪ/. In Chinese, the long vowel /ɪː/ and short vowel /ɪ/ do not form minimal pairs, which means that the duration difference of the /ɪ/ sound in Chinese does not change the meaning of a word. Therefore, it is common for Chinese learners to mix the long /ɪː/ and the short /ɪ/ when they speak English – causing mistakes such as substituting /ʃɪːp/ (sheep) to /ʃɪp/ (ship). Chinese and English have about the same number of consonants. However, some English consonants do not exist in Chinese. For example, Chinese English speakers find it hard to produce dental fricatives /θ/ (as in "theta") and /ð/ (as in "thee") since they do not exist in their native phonology. Therefore, they often substitute them with similar-sounding consonants such as /z/ and /s/. The Chinese /ʃ/ and /r/ have different realizations in terms of place and manner of articulation compared with their English counterparts. Therefore, it is not surprising to find mispronunciations by Chinese English learners when they utter words like "English," "pronunciation," "rose," or "rise."

The phonology of a language also dictates what kinds of consonants clusters (phonotactics) are allowed. In Chinese, morphemes are generally made up of a consonant plus a vowel with no consonants cluster and usually end with a vowel. As a result, Chinese speakers commonly insert a vowel after an ending consonant (e.g., pronouncing words "book" and "bed" as /bukə/ and /bedə/). Chinese and English also have significant prosodic differences. For example, Chinese is a tonal language, while English is an intonation language. Besides, Chinese is syllable-timed, meaning that the syllables are roughly the same duration, whereas English is stress-timed (durations between stressed consecutive syllables are equal). Thus, the intonation and rhythm in Chinese-accented English differ significantly from that of native English.

The influence of a speaker's native language in their production of the second language is such a reliable indicator of their mother tongue such that it can be identified with high accuracy. For example, Behravan [45] used an i-vector framework [46] to distinguish between seven non-native English accents (Hindi, Russian, Korean, Japanese, Thai, Cantonese, and Vietnamese) and achieved 74% overall detection accuracy.

2.2. Speech signal analysis and synthesis

In this subsection, I briefly review the source-filter model of speech production, the speech parameterizations (i.e., features) used in this work, as well as the speech vocoders I adopt for speech modification. The vocoding process consists of converting audio waveform into speech features and resynthesizing these features back to the waveform, and a vocoder is a set of algorithms and tools that perform the vocoding process.

2.2.1. Source-filter model for speech production

The source-filter model [47] is a well-known theory of speech production. The model argues that speech occurs when a *source* excitation signal passing through the larynx (or at some point along the length of the vocal tract) is modified by the vocal tract acting as a *filter*. More specifically, the vibration of the vocal cords produces a complex periodic wave, and the spectrum of this wave contains energy at the fundamental frequency of laryngeal vibration and multiples of the fundamental frequency – harmonics. The vocal tract acts as a filter to accentuate and attenuate the source signals at particular frequencies when they pass through the vocal tract. When the vocal tract configuration changes through articulations, the resonance characteristics of the vocal tract also change, resulting in different speech sounds. From a signal processing point of view, using the theory of linear time invariant (LTI) systems, the overall process can be modeled in the z-domain as $Y(z) = U(z)V(z)R(z)$, where $Y(z)$ is the speech signal, $U(z)$ is the glottal source, and $V(z)$ and $R(z)$ are the transfer functions of the vocal tract and lips.

2.2.2. Mel-spectrogram

Speech processing tasks are generally carried out in the frequency domain. To do so, it is typical to convert the time domain speech waveform into power spectra[2] using the Short-Time Fourier Transform (STFT); see Figure 2.1 for an illustration.

[2] The English plural "spectrums" is not preferred in speech processing.

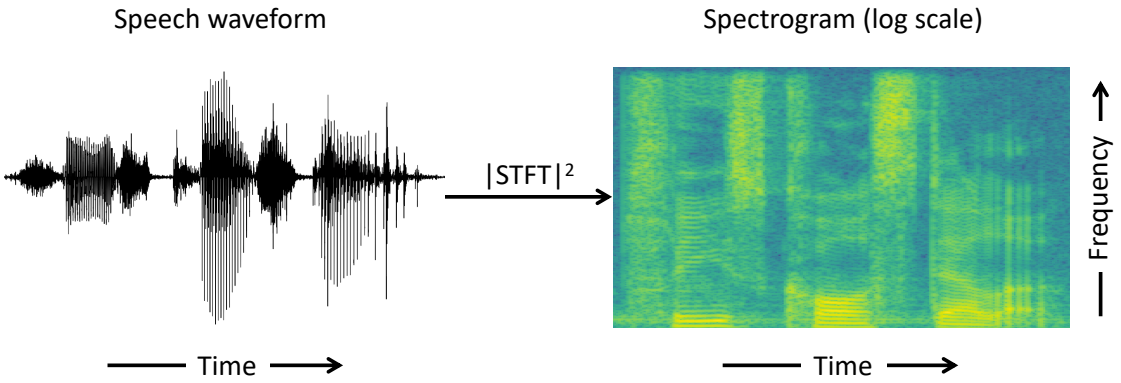

Figure 2.1: Converting speech waveform to power spectra. A spectrogram is a short-time spectrum plotted over time.

The mel-spectrogram is computed by passing the power spectra through mel filter-banks (Figure 2.2), which are the overlapping triangular filters uniformly spaced in the mel scale frequency[3]. The mel scale has a frequency resolution that is similar to the human auditory system, and therefore it is useful for speech synthesis and recognition tasks [49]. Recently, mel-spectrograms have become increasingly popular in speech synthesis due to advances in neural vocoders, which can directly synthesize high-quality speech waveforms from them. I introduce neural vocoders in Section 2.2.7.

[3] A popular formula ([48], p. 150) for the conversion between the mel scale frequency m and the linear scale frequency f is $m = 2595 \log_{10}(1 + f/700)$.

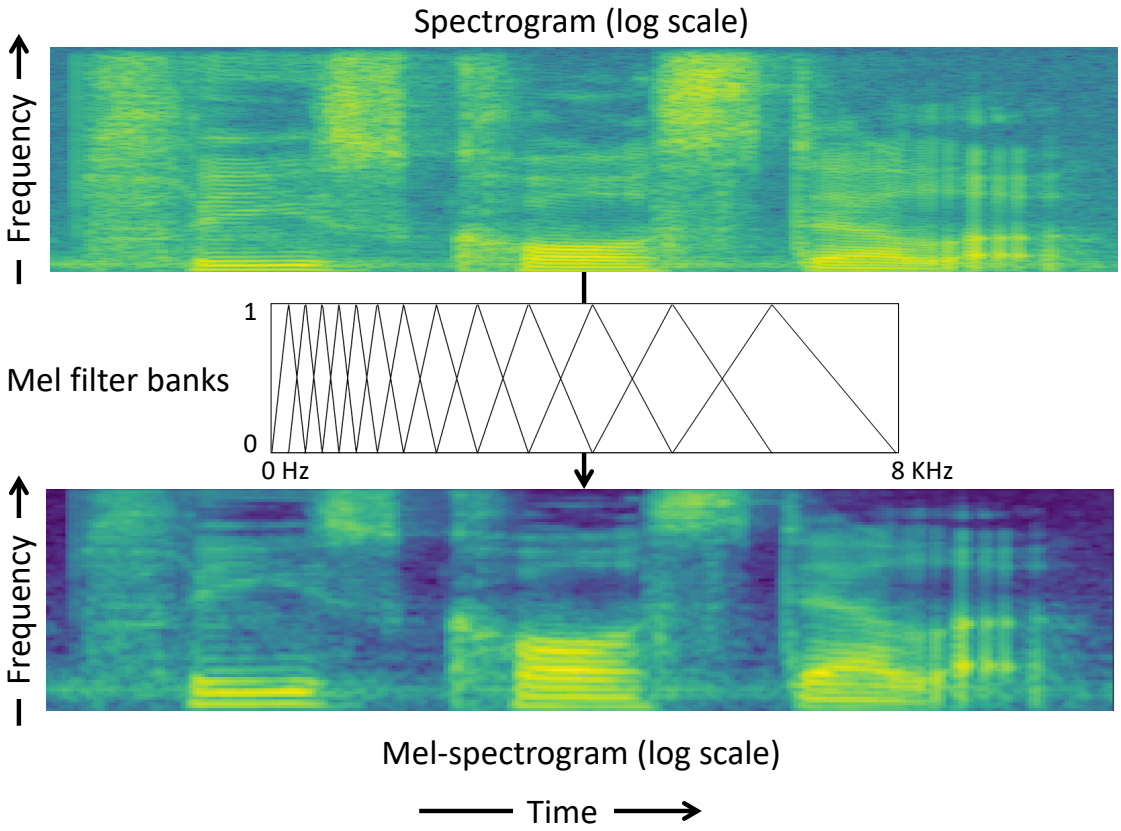

Figure 2.2: Convert power spectra to mel-spectrogram (8 kHz cut-off). The mel-spectrogram was produced by passing the power spectra through 80 mel filter banks. For visualization purposes, we only plotted 13 such triangular filter banks in the figure.

2.2.3. Mel Frequency Cepstral Coefficients (MFCCs)

The cepstral coefficients are calculated by taking the discrete cosine transform (DCT) of the log power spectrum of the speech. The *source* and *filter* components of the speech signal can then be separated by "liftering" (low-pass filtering in the cepstral domain). The most commonly used cepstral coefficients – Mel Frequency Cepstral Coefficients (MFCCs) [50] are obtained by (1) calculating the mel filter bank energies (compressing the STFT spectra through the mel filter banks; the same process as in computing

the mel-spectrogram); (2) taking the logarithm of the output mel filter bank energies; and (3) taking the DCT of the log mel filter bank energies. See Figure 2.3 for an illustration of this process. Due to its relation with the human auditory system, MFCC has become the *de facto* representation for speech recognition [51].

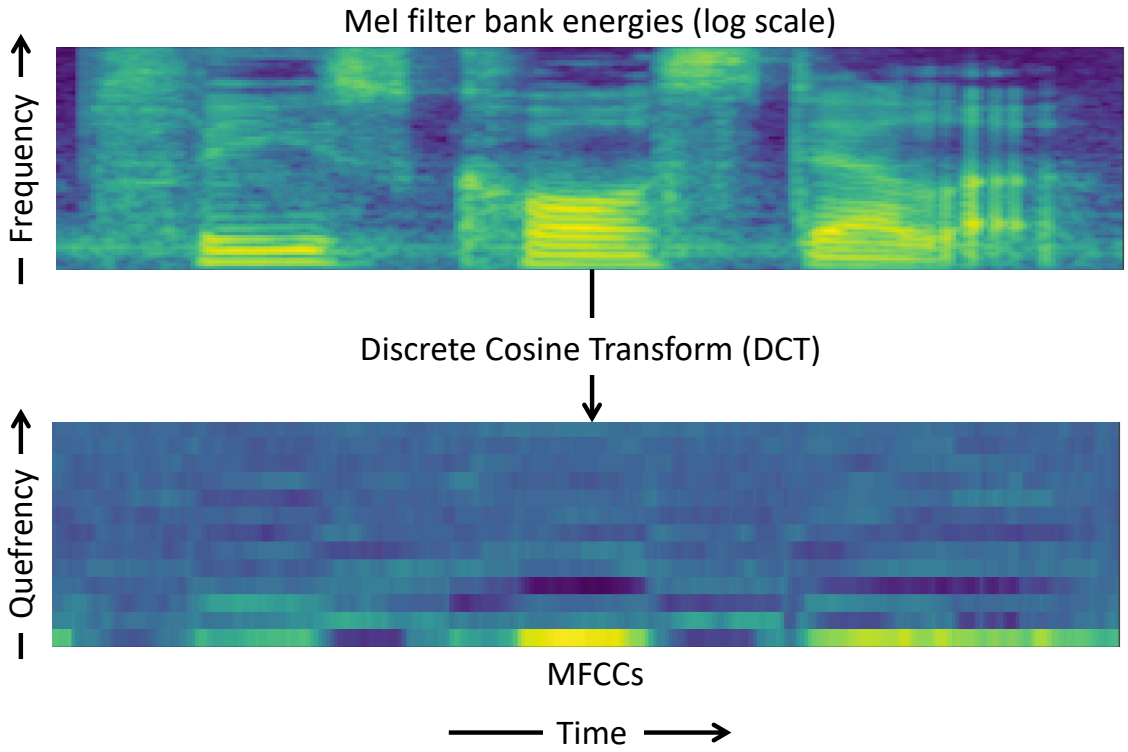

Figure 2.3: Compute MFCCs from mel filter bank energies.

2.2.4. Mel-Cepstral Coefficients (MCCs/MCEPs)

In another widely used variant of mel-cepstral analysis, instead of using MFCCs, we compute the M-th order Mel-Cepstral Coefficients (MCCs/MCEPs) $c_\alpha(m)$ to model the spectrum $H(e^{j\omega})$ of the speech signal,

$$H(z) = \exp \sum_{m=0}^{M} c_{\alpha}(m)\tilde{z}^{-m}\,, \tag{2.1}$$

where the first order all-pass transfer function is expressed as,

$$\tilde{z}^{-1} = \frac{z^{-1} - \alpha}{1 - \alpha z^{-1}}, \qquad |\alpha| < 1. \tag{2.2}$$

The variable α controls the phase characteristic of the all-pass transfer function. The actual value of α is empirically determined based on the sampling rate of the speech signal. Common α values include 0.554 (48 kHz), 0.544 (44.1 kHz), 0.42 (16 kHz), 0.35 (10 kHz), and 0.31 (8 kHz).

MCEPs can be estimated by minimizing a cost function based on the unbiased log spectrum estimation method [49] using the Newton-Raphson method. Figure 2.4 shows an example of the MCEP feature. MCEP is one of the most common spectral representations used in statistical parametric speech synthesis, as shown by its performance in prior works on speech synthesizers [21, 52-54].

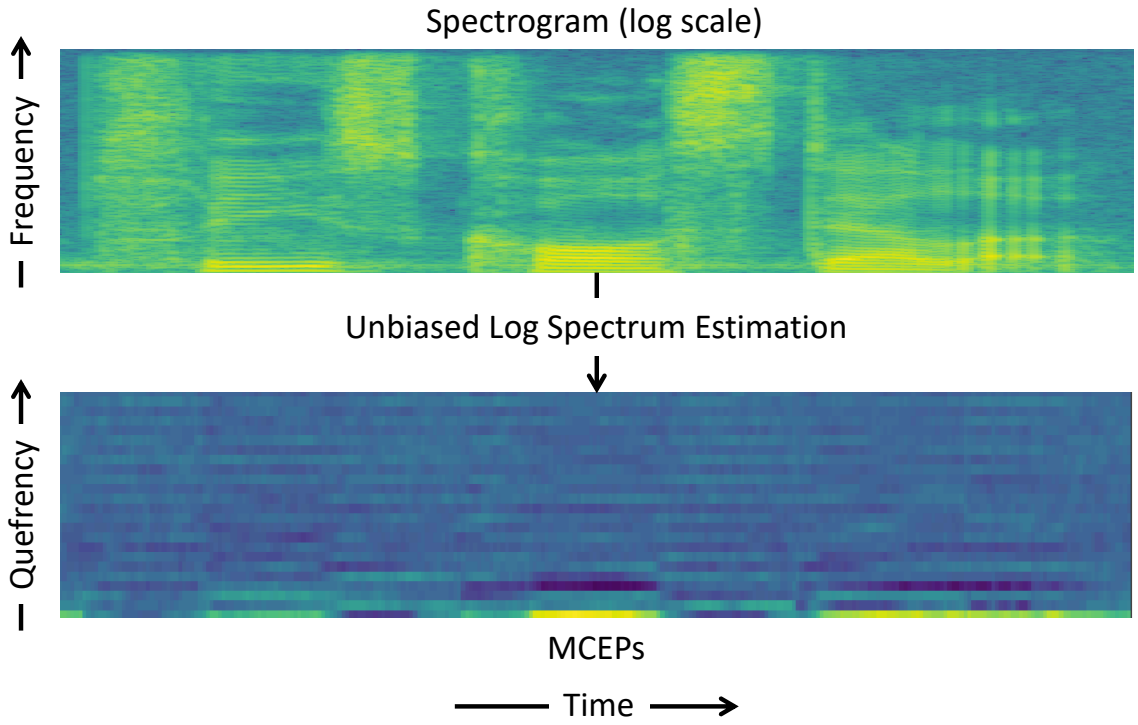

Figure 2.4: Compute MCEPs from the log spectra with unbiased log spectrum estimation [49].

2.2.5. STRAIGHT vocoder

STRAIGHT (Speech Transformation and Representation using Adaptive Interpolation of weiGHTed spectrum) is a vocoder that uses bilinear interpolation over the time-frequency representation of the speech signal to estimate the spectrogram [55]. STRAIGHT analysis decomposes the speech signal into three independent components (Figure 2.5): (1) a spectrogram that is decoupled (as much as possible) from the fundamental frequency and the harmonics, (2) a one-dimensional fundamental frequency (F_0) signal, and (3) an aperiodicity signal, which is the spectrogram of the nondeterministic excitation signal (e.g., noise). This model allows independent modification of these three

components without any significant decrease in the naturalness and the acoustic quality of the synthesis. Due to the naturalness of the synthesis and the flexibility of the model, the STRAIGHT vocoder has gained popularity in applications such as voice conversion [21, 56] and parametric text-to-speech synthesis [53]. In the first proposed foreign accent conversion system, I extract MCEPs from the STRAIGHT spectrogram.

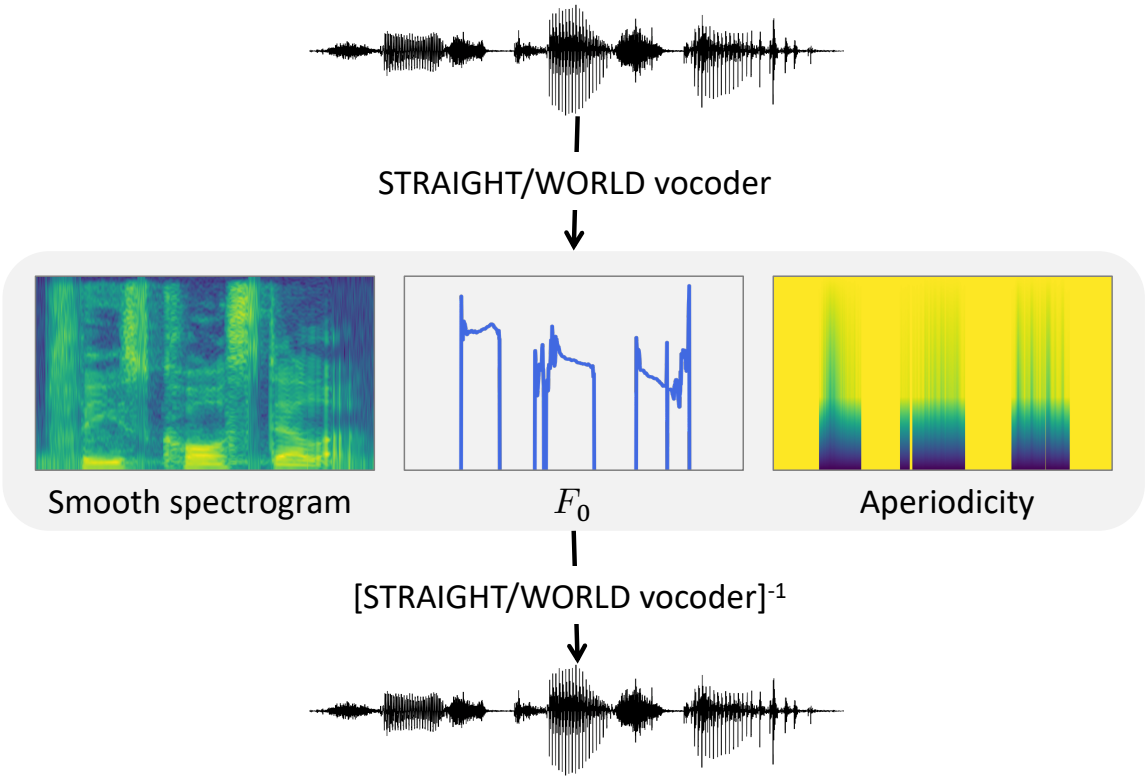

Figure 2.5: Speech analysis and synthesis using conventional vocoders (STRAIGHT or WORLD).

2.2.6. WORLD vocoder

The WORLD vocoder [57] follows the same design philosophy as the STRAIGHT vocoder; it also decomposes speech waveform into a smoothed spectral envelope, an F_0

trajectory, and an aperiodicity signal (Figure 2.5). However, the WORLD vocoder uses more accurate algorithms for estimating the three vocoder features – the DIO algorithm [58] for estimating the F_0, the CheapTrick algorithm [59] for computing the spectral envelope, and the PLATINUM algorithm [60] for extracting the aperiodicity. Since the WORLD vocoder produces significantly higher-quality speech with a much better real-time factor compared with the STRAIGHT vocoder [61], recent works on speech modification and synthesis tend to choose the WORLD vocoder. In this dissertation, when applicable[4], I use the WORLD vocoder to extract the spectrogram, F_0, and aperiodicity.

2.2.7. Neural vocoder

The neural vocoder is inspired by neural network-based Text-To-Speech (TTS) systems such as the WaveNet model [62]. The inputs to a neural TTS system are the linguistic features extracted from the text input, and the outputs are speech waveforms. To build a neural vocoder, the input to the neural TTS model is replaced with acoustic features. These acoustic features can be the output of a conventional vocoder (e.g., STRAIGHT, WORLD) or raw features like mel-spectrogram. There are different variations of neural vocoders; a few notable works include WaveNet [63], WaveGlow [64], FFTNet [65], and LPCNet [66].

The main advantage of a neural vocoder is that it can generate speech with audio quality that is comparable to natural speech. On the downside, the neural vocoder needs to be trained with a relatively large amount of speech data, and the training process can be

[4] Some proposed works were performed before the WORLD vocoder was released.

slow. The first neural vocoder, the WaveNet vocoder, suffers from slow inference speed due to its autoregressive nature, but this can be resolved with alternative implementations, such as Parallel-WaveNet [67] and WaveGlow. An interesting property of a neural vocoder is that even if the vocoder is trained on only one speaker, it can generally synthesize speech for another speaker from the same gender. Recent research [68] has proposed speaker-independent neural vocoders.

2.3. Sequence-to-sequence models

A sequence-to-sequence (seq2seq) model is a neural network model that can directly transform an input sequence from one domain to an output sequence from another domain. Figure 2.6 provides a high-level illustration of a seq2seq model. Most commonly, a seq2seq model contains an encoder that consumes the input sequence and generates a hidden representation, and a decoder that reads the hidden representation and predicts the output sequence in an autoregressive manner. The encoder and decoder usually consist of Recurrent Neural Network (RNN) cells, such as the Long Short-Term Memory (LSTM) [69] units or Gated Recurrent Unit (GRU) [70].

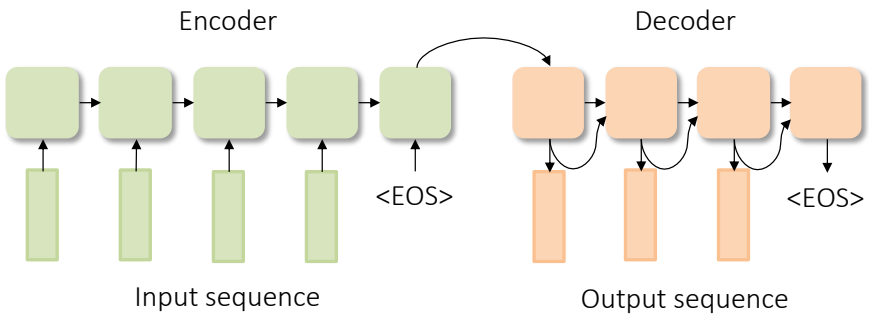

Figure 2.6: High-level illustration of a vanilla sequence-to-sequence model. <EOS> represents the end of the sequence.

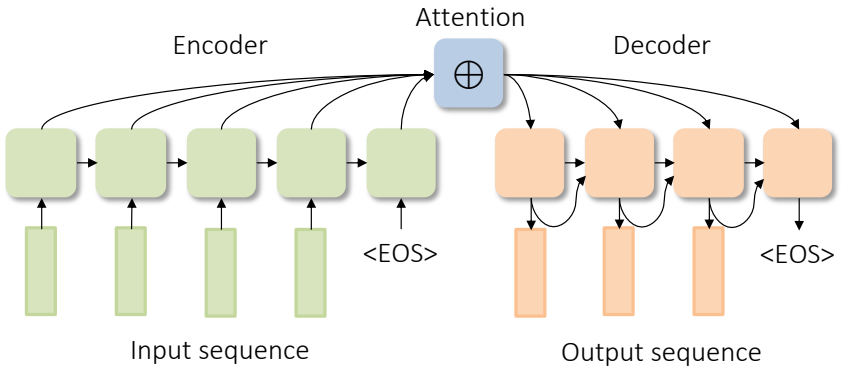

Figure 2.7: Conceptual illustration of a sequence-to-sequence model with the attention mechanism. <EOS> represents the end of the sequence. ⊕ represents the weighted sum operation.

A major improvement to this "vanilla" seq2seq model is the introduction of an attention mechanism [71]. As the name suggests, the essential idea of the attention mechanism is to let the model decide what parts of the input sequence contain useful information when making the predictions. Instead of using the last frame output from the encoder as the hidden representation, the attention mechanism computes a weighted sum of all the outputs from the encoder at each decoding time step and then uses the weighted attention-context vector as the hidden representation to predict the output features. See Figure 2.7 for a conceptual illustration of the attention mechanism. The attention weights are generally produced by learnable layers (i.e., parameters) of the seq2seq model.

Seq2seq models obtained early success in the machine translation task [28]. Later, they were adopted by the speech recognition community to produce end-to-end automatic speech recognition [71, 72]. Seq2seq models have also been adopted in the speech synthesis community to develop end-to-end TTS synthesizers [73, 74]. Due to the sequential

and monotonic nature of the speech signal, the seq2seq model and the attention mechanism are especially suitable for speech-related tasks, and I use this type of model extensively in my dissertation.

3. USING PHONETIC POSTERIORGRAM BASED FRAME PAIRING FOR SEGMENTAL ACCENT CONVERSION*

3.1. Overview

Accent conversion (AC) aims to transform non-native utterances to sound as if the speaker had a native accent. This can be achieved by mapping source speech spectra from a native speaker into the acoustic space of the target non-native speaker. In prior work, we proposed an AC approach that matches frames between the two speakers based on their acoustic similarity after compensating for differences in vocal tract length. In this chapter, we propose a new approach that matches frames between the two speakers based on their phonetic (rather than acoustic) similarity. Namely, we map frames from the two speakers into a phonetic posteriorgram using speaker-independent acoustic models trained on native speech. We thoroughly evaluate the approach on a speech corpus containing multiple native and non-native speakers. The proposed algorithm outperforms the prior approach, improving ratings of acoustic quality (22% increase in mean opinion score) and native accent (69% preference) while retaining the voice quality of the non-native speaker. Fur-

ther, we show that the approach can be used in the reverse conversion direction, i.e., generating speech with a native speaker's voice quality and a non-native accent. Finally, we show that this approach can be applied to non-parallel training data, achieving the same accent conversion performance.

3.2. Introduction

Learners who acquire a second language (L2) after a "critical period" [33] usually speak with a non-native accent. Having a non-native accent can often reduce the speaker's intelligibility [4] and may also lead to discriminatory attitudes [75, 76]. Therefore, non-native speakers have much to gain by improving their pronunciation. Several studies [10, 11] have shown that having a suitable native (L1) speaker to imitate – a so-called "golden speaker" with similar voice characteristics as the learner but with a native accent, can be beneficial in pronunciation training. Based on these findings, Felps et al. [12] suggested that such a "golden speaker" could be created by resynthesizing the non-native speaker's own voice with a native accent borrowed from a native reference speaker.

Traditional voice-conversion (VC) methods [21, 77-79] cannot be used for this purpose since VC cannot decouple the speaker's voice quality from her or his accent, i.e., VC assumes that accent is part of the speaker's identity. In this work, we distinguish two concepts: *voice quality,* which focuses on the physical characteristics of the speaker's voice (e.g., vocal tract and glottal configuration, pitch range), and *speaker identity*, a combination of *voice quality* and other speaker characteristics (e.g., accent, speaking rate, intonation, word choice).

To address the accent-and-voice-quality entanglement issue of traditional VC methods, Aryal and Gutierrez-Osuna [16] proposed a modified VC method where source frames (i.e., from the native reference speaker) and target frames (i.e., from the non-native speaker) were paired based on their acoustic similarity. In a first step, the authors applied vocal-tract length normalization (VTLN) to the source speech, so it matched the target speaker's vocal-tract length. Then, they paired each frame in the source corpus with the closest frame in the target corpus, and vice versa. Though VTLN did improve frame pairing compared to time alignment (i.e., the conventional approach in VC), vocal-tract length is just one of the potentially many differences between speakers, and it is too coarse to account for differences in pronunciation.

To address this issue, we present an approach that matches source and target frames based on their phonetic content. Leveraging advances in acoustic modeling [80], we extract phonetic information from phonetic posteriorgrams (PPGs) [24]. Namely, we compute the posteriorgram for each source and target speech frame through a speaker-independent acoustic model trained on a large corpus of native speech. Then, we use the symmetric Kullback-Leibler (KL) divergence [81] in posteriorgram space to match source and target frames. The result is a set of source-target frames that are paired based on their phonetic similarity, with which we train a Gaussian Mixture Model (GMM) to model the joint distribution of source and target Mel-Cepstral Coefficients (MCEPs). In a final step, we map source MCEPs into target MCEPs using maximum likelihood estimation of spectral parameter trajectories considering the global variance [21] of the target speaker. Our implementation is based on a conventional GMM spectral mapping method to ensure a

fair comparison with the prior study [16], but our proposed frame matching method can be combined with any spectral mapping methods (e.g., neural networks, frequency warping) that take frame pairs as input.

Our approach differs from prior works on accent conversion, which modify speech features that carry accent information, such as prosody, formants, spectral envelopes, or articulatory gestures [12, 13, 15, 82]. Instead, we use a VC technique to capture the voice quality of the (target) non-native speaker while preserving the (source) native speaker's pronunciation characteristics – both segmental and prosodic. Unlike VC methods, however, we avoid the issue of time aligning source and target utterances, which is problematic when the target speaker is non-native. Our approach is related to that of Xie et al. [83], who used speaker-adaptive acoustic models to generate posteriorgrams for VC. Their method groups all target speaker training data into phonetic clusters in the posteriorgram space using symmetric KL divergence and K-means clustering. Then, each frame of the source speaker's corpus is mapped to the centroid of the closest target phonetic cluster. The final converted speech is generated from those closest cluster centroids using the maximum probability trajectory generation algorithm. In contrast with their frame clustering approach, we use PPGs to produce frame pairs between source and target speakers, and then we train a GMM using those frame pairs. A second major difference with their approach is that we use speaker-independent acoustic models trained on native speech to ensure that the PPGs only reflect native pronunciations, whereas their approach uses speaker-adaptive training, which would introduce non-native pronunciations into the

acoustic models. Initial findings from this work were presented at the International Conference on Acoustics, Speech, and Signal Processing (ICASSP) in 2018 [84]. That earlier conference paper presented preliminary listening test results that verified the effectiveness of the PPG-based frame-matching method. The present manuscript describes our method in detail and significantly expands the perceptual studies and data analyses, including an experimental comparison of the proposed method on parallel and non-parallel data.

3.3. Literature review

3.3.1. Algorithms for accent conversion

Foreign and non-native accents occur when speech deviates from the expected acoustic (e.g., formants) and prosodic (e.g., intonation, duration, and rate) norms of a language [12]. Therefore, prior work has focused on modifying certain speech characteristics to alter the perceived accent. In early work, Yan, et al. [85] used a voice-morphing software to change the trajectories of formants, pitch, and duration to convert between three different English dialects (British, Australian, and General American English). The authors found that prosodic modifications produced noticeable differences on perceived accent, although not as significant as those produced by modifying formant trajectories. In the approach of Felps et al. [12], the spectral envelope of the non-native speech was replaced with that of the native speaker's, which had been normalized to the non-native speaker's vocal tract length with a piecewise linear warping function. Their results showed that the segmental correction was able to significantly reduce the foreign accentedness of the modified utterances. More recently, Jügler, et al. [86] used PSOLA to correct the prosody of non-native German speech spoken by native French speakers. Prosodic (duration

and pitch) corrections were performed at the syllable level, and the results showed a moderate but significant reduction in accentedness of the corrected speech.

A couple of studies also tried to blend native and non-native spectra to control the accent. Huckvale and Yanagisawa [15] blended the spectral envelope of non-native Japanese speech produced by an English Text-To-Speech (TTS) with its native counterpart through voice morphing to reduce the accent. Aryal, et al. [13] decomposed the cepstrum into spectral slope and spectral detail, and then generated accent conversions by combining the spectral slope of the non-native speaker with a morph of the spectral detail of the native speaker. Though these spectra-blending methods can reduce non-native accents, they also tend to produce syntheses that are perceived as a "third speaker," one who is different from either the source (native) or target (non-native) speaker. To tackle this problem, Aryal and Gutierrez-Osuna [16] adapted VC techniques to perform accent conversion. The authors used vocal-tract-length normalization (VTLN) before pairing acoustic frames between source (native) and target (non-native) speaker, then built a GMM using those frame pairs to perform VC. This method was able to reduce non-native accent significantly, while retaining the non-native speaker's voice quality; however, it required a relatively large set of parallel recordings from the two speakers, and VTLN only accounted for a subset of the speaker characteristics.

An alternative to using acoustic methods is to operate in the articulatory domain. Along these lines, Felps, et al. [82] used an articulatory synthesizer based on unit-selection to replace mispronounced non-native diphones with those from the non-native corpus that matched the articulatory configuration of a reference utterance from a native speaker.

Later, Aryal and Gutierrez-Osuna used GMMs [17] and DNNs [87] to build an articulatory synthesizer (i.e., a mapping from articulatory gestures into acoustics) for the non-native speaker, then drove the GMM/DNN with articulatory gestures from a native speaker. Methods based on articulatory data generate syntheses that sound more like the non-native (target) speaker than acoustic methods, since they effectively decouple linguistic information (e.g., articulatory gestures from a native [source] speaker) from voice quality (captured by the articulatory-to-acoustic synthesizer of the non-native speaker). However, articulatory methods are expensive and require specialized equipment to collect articulatory data, so they are impractical for pronunciation training.

3.3.2. Connection between accent and voice conversion

Accent conversion is closely related to the problem of voice conversion [88]. Voice conversion transforms a source speaker's speech into that of a (known) target speaker. The conversion aims to match the voice characteristics of the target speaker, which may include vocal tract configuration, glottal characteristics, pitch range, pronunciation, and speaking rate. Ideally, the only information retained from the source speech is its linguistic content, i.e., the words that were uttered. Popular methods for voice conversion include joint-density GMMs [21], frequency warping [89, 90], DNNs [91, 92], and sparse coding [79, 93-95]. Accent conversion modifies speech at a finer level of granularity, and seeks to combine the linguistic content and pronunciation of the source speaker with the voice quality of the target speaker. Therefore, accent conversion is a more challenging problem than voice conversion in the sense that, first, there is no ground truth for the output voice, and second, accent conversion needs to split the speech into voice

quality (converted) and accent (preserved), whereas voice conversion jointly converts both.

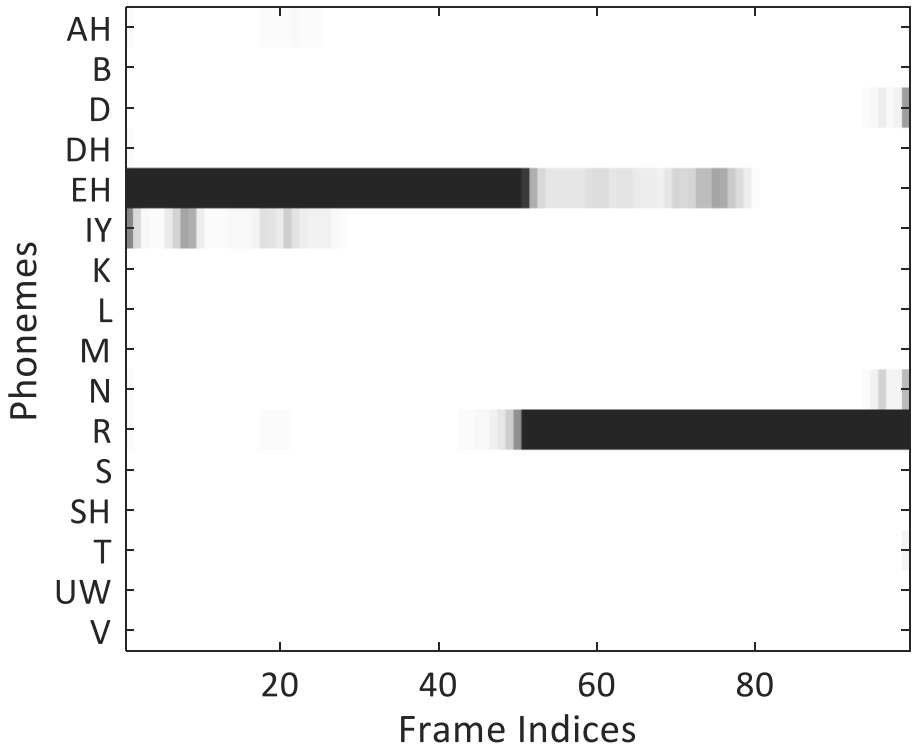

Figure 3.1: PPG for the word "air," whose phonetic transcription in ARPAbet is "EH R." For visualization purposes, we used a subset of the ARPABET phoneme set and omitted phonemes that had small values.

3.4. Method

3.4.1. Phonetic posteriorgrams

At its core, our proposed method relies on Phonetic Posteriorgrams (PPGs) to measure the similarity of speech frames across speakers. A phonetic posteriorgram is computed by segmenting speech into frames and computing the posterior probability that each frame belongs to a set of pre-defined phonetic units. As an example, Figure 3.1 shows the

PPG of the spoken word "air." In practice, it is advisable to include context when computing the PPG by concatenating each speech frame with its neighboring right and left frames. Moreover, phoneme labels are too coarse to describe the variety of speech sounds. Therefore, the dimensions in a phonetic posteriorgram are often associated with triphones, as we will see next.

Generally, the phonetic posteriorgram is computed from the acoustic model in an automatic speech recognizer (ASR). The acoustic model in ASR acts as a sequential classifier: given an input acoustic feature vector, the acoustic model assigns how likely it is that the vector belongs to each of a set of states/senones. In recent years, acoustic models based on DNNs have yielded state-of-the-art speech recognition accuracy [80]. The most advanced ASR systems can achieve Word Error Rates that are comparable to or better than expert human transcribers on specific tasks [96].

In this work, we compute phonetic posteriorgrams using a p-norm DNN [97] as the acoustic model. The input layer accepts a feature frame accompanied by its left and right neighbors; then the input is de-correlated by a fixed linear transformation [98]. The de-correlated features are then passed through N hidden layers, each employing the p-norm non-linearity $y = \|\boldsymbol{x}\|_p = \left(\sum_i |x_i|^p\right)^{1/p}$, where y is one output dimension of a hidden layer and $\boldsymbol{x}$ represents a group of hidden neurons of that layer. Therefore, the number of output dimensions of each hidden layer is smaller than the number of hidden neurons. The output of the p-norm layer is then processed by a normalization layer to limit its standard deviation to one [97]. The output of the final hidden layer is fed into a softmax layer that produces more output nodes than the desired number of senones using a technique

called “mixing-up” [97]. “Mixing-up” operates as follows. About halfway through training, the dimension of the softmax layer is increased by letting each output senone’s probability be a sum over potentially multiple “mixture components.” The mixture components are distributed using a power rule, proportional to the senone class priors. The neural network then “group-sums” the output of the softmax layer according to the group assignment defined in the “mixing-up” step, resulting in the final output nodes that correspond to individual senones. Figure 3.2 shows the overall structure of the p-norm deep neural network that we use in this work.

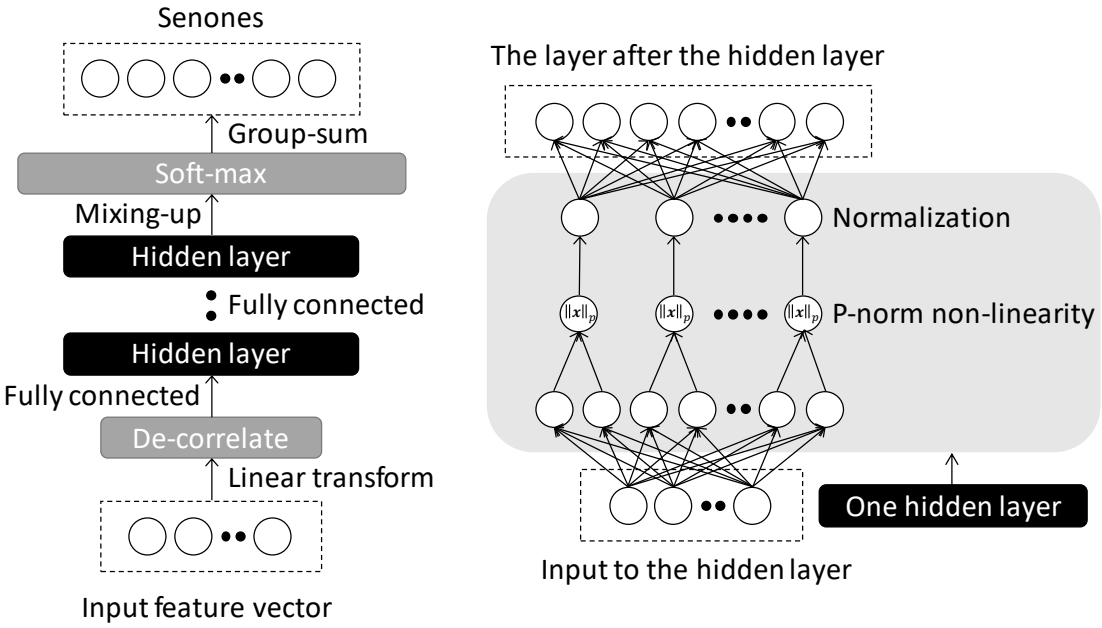

Figure 3.2: P-norm deep neural network structure for acoustic modeling.

During training, inputs to the p-norm DNN consist of stacked MFCC frames $\boldsymbol{X}$, whereas target outputs $\boldsymbol{Y}$ are senone labels obtained from force-alignment using an existing GMM-HMM speech recognizer. The training objective is the sum (across all frames of training data) of the log-probability of $\boldsymbol{Y}$ given $\boldsymbol{X}$: $\sum_i \log p(\boldsymbol{Y}_i|\boldsymbol{X}_i)$. After the DNN is fine-tuned using Stochastic Gradient Descent [99], we compute the posterior probability of observing senone l given the speech frame $\boldsymbol{x}$ by doing a complete forward propagation,

$$p(l|\boldsymbol{x}) = \sum_{g \in G} \frac{\exp(x'_g)}{\sum_k \exp(x'_k)}, \tag{3.1}$$

where x'_k is the output of the hidden layer that precedes the softmax layer, and G is the set of softmax outputs that are grouped into senone l during the "mixing-up" procedure. A PPG frame of $\boldsymbol{x}$ is constructed by forming a vector from all possible values of $p(l|\boldsymbol{x})$, see eq. (3.2).

3.4.2. Frame pairing

Conventional *voice* conversion methods use time alignment to pair frames from source and target utterances. As such, a VC model trained from time-aligned frame pairs will retain the non-native speaker's accent. Instead, to perform *accent* conversion, the pairing must be based on the phonetic similarity between source and target frames. In this way, each native speech frame is associated with its most similar non-native counterpart in terms of pronunciation. If we train a spectral conversion model between these frame pairs, the pronunciation from the native speech data will be preserved and the spectral envelope of the native speaker will be modified to match the non-native speaker's voice quality.

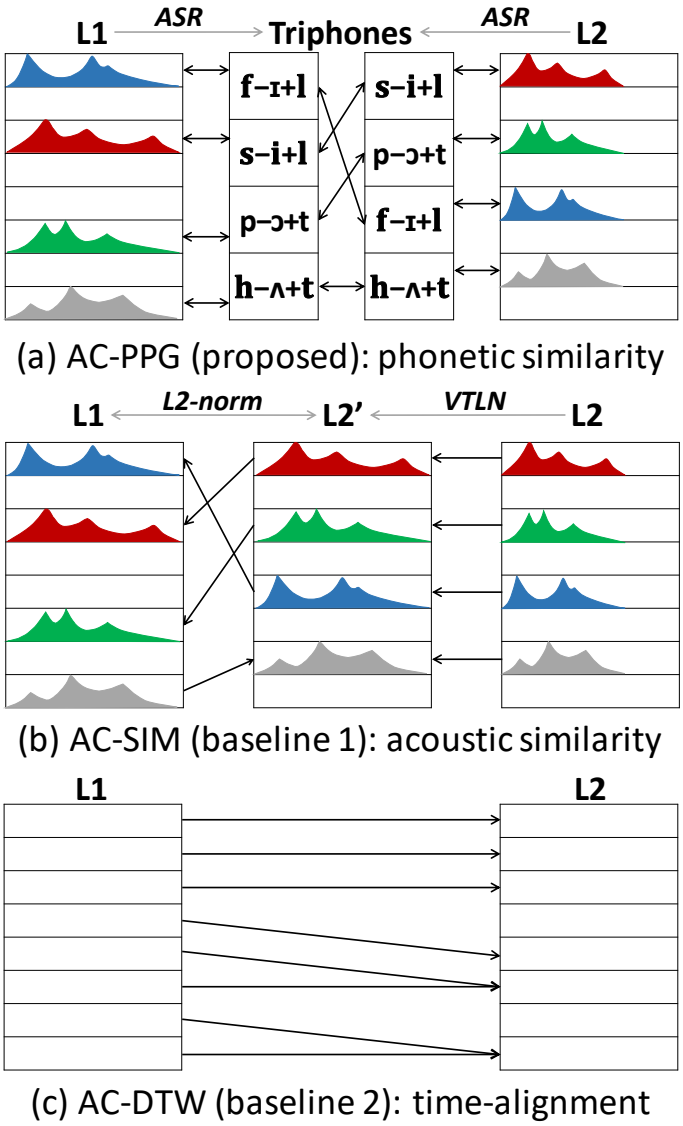

Figure 3.3: L1: native, L2: non-native. (a) AC-PPG: proposed AC algorithm that uses phonetic similarity. (b) AC-SIM: Baseline 1 that uses acoustic similarity through VTLN to pair frames [16]. (c) AC-DTW: Baseline 2; native and non-native frames are time-aligned following their ordering in the data.

3.4.2.1. Frame pairing based on phonetic similarity (AC-PPG)

We use PPGs to pair frames between the native and non-native speaker. Our rationale is straightforward: if an ASR trained on native speech determines that a non-native

speech segment $\boldsymbol{y}$ is close to the native speech production of a particular phoneme (or triphone, in our case), then it is reasonable to pair $\boldsymbol{y}$ with a native speech segment $\boldsymbol{x}$ with the same phonetic label; see Figure 3.3 (a). Specifically, our approach works as follows. In a first step, we compute PPG frames for speech frames from the two speakers,

$$\mathcal{L}_{\boldsymbol{x_i}} = [p(l_1|\boldsymbol{x_i}), p(l_2|\boldsymbol{x_i}), \dots,\ p(l_V|\boldsymbol{x_i})], \tag{3.2}$$

where $\boldsymbol{x_i}$ is the acoustic feature vector of the i-th speech frame; $V = \{l_1,\ l_2, \dots,\ l_V\}$ is the predefined senone set; $P(l_j|\boldsymbol{x_i})$ is the conditional probability that the speech frame belongs to senone l_j given $\boldsymbol{x_i}$; $\sum_j P(l_j|\boldsymbol{x_i}) = 1$.

Given posterior feature vectors $\mathcal{L}_{\boldsymbol{x_i}}$ and $\mathcal{L}_{\boldsymbol{x_j}}$, we calculate their distance using the symmetric KL divergence,

$$D\big(\mathcal{L}_{\boldsymbol{x_i}}, \mathcal{L}_{\boldsymbol{x_j}}\big) = \big(\mathcal{L}_{\boldsymbol{x_i}} - \mathcal{L}_{\boldsymbol{x_j}}\big) \cdot \big(\log \mathcal{L}_{\boldsymbol{x_i}} - \log \mathcal{L}_{\boldsymbol{x_j}}\big). \tag{3.3}$$

The symmetric KL divergence is commonly used to compute the similarity between distributions, and here, each frame of the PPG functions like a distribution. For each source (i.e., native) frame $\boldsymbol{x_i}$ we find its closest target (i.e., non-native) frame $\boldsymbol{y_i^*}$,

$$\boldsymbol{y_i^*} = \underset{\forall \boldsymbol{y}}{\operatorname{argmin}}\, D(\mathcal{L}_{\boldsymbol{x_i}}, \mathcal{L}_{\boldsymbol{y}})\,. \tag{3.4}$$

Likewise, for each non-native frame $\boldsymbol{y_i}$ we find its closest native frame $\boldsymbol{x_i^*}$,

$$\boldsymbol{x_i^*} = \underset{\forall \boldsymbol{x}}{\operatorname{argmin}}\, D(\mathcal{L}_{\boldsymbol{x}}, \mathcal{L}_{\boldsymbol{y_i}})\,. \tag{3.5}$$

Each frame pairing process only involves two speakers – the given native and non-native speakers. The frame pairing does not constrain the search space. Therefore, it is possible to pair multiple frames from one speaker with the same frame from the other

speaker. In this case, we duplicate that frame multiple times. The resulting frame pairs are used to train a Gaussian Mixture Model (GMM).

3.4.2.2. Baseline methods for frame pairing

We compared the proposed PPG-based method against two baseline techniques for frame pairing: the acoustic similarity method of Aryal and Gutierrez-Osuna [16], and dynamic time warping.

Baseline 1 *(AC-SIM)*. Following [16], we measured acoustic similarity as the inverse of the L2-norm between native and non-native speaker frames, after normalizing the native speaker to match the vocal tract length of the non-native speaker; see Figure 3.3 (b).

In a first step, we learn a VTLN transform to reduce physiological differences in vocal tract between the two speakers. For this purpose, we time-align parallel training utterances of the two speakers, each utterance represented as a sequence of MFCCs. Following Panchapagesan and Alwan [100], we then learn a linear transform between the MFCCs of both speakers using ridge regression:

$$T^* = \underset{T}{\operatorname{argmin}} \|\boldsymbol{x} - T\boldsymbol{y}\|^2 + \lambda \|T\|^2, \tag{3.6}$$

where $\boldsymbol{x}$ and $\boldsymbol{y}$ are vectors of MFCCs from the native and non-native speakers, respectively, and T^* is the VTLN transform. Next, for each native vector $\boldsymbol{x_i}$ we find its closest non-native vector $\boldsymbol{y_j^*}$ as:

$$\boldsymbol{y_j^*} = \underset{\forall \boldsymbol{y}}{\operatorname{argmin}} \|\boldsymbol{x_i} - T^*\boldsymbol{y}\|^2 \,. \tag{3.7}$$

We repeat the process for each non-native vector $\boldsymbol{y_i}$ to find its closest match $\boldsymbol{x_j^*}$:

$$\boldsymbol{x}_{\boldsymbol{j}}^{*} = \underset{\forall \boldsymbol{x}}{\operatorname{argmin}} \|\boldsymbol{x} - T^{*}\boldsymbol{y}_{\boldsymbol{i}}\|^{2} . \tag{3.8}$$

The above process results in a lookup table where each native and non-native frame in the database is paired with the closest one from the other speaker.

Baseline 2 *(AC-DTW).* As our second baseline method, we use Dynamic Time Warping (DTW) [101] to time-align native and non-native frames, as illustrated in Figure 3.3 (c).

We note that baselines 1 and 2 need parallel data for training, whereas the proposed method can operate on non-parallel data, as we shall see in the experiments.

3.4.2.3. Spectral conversion

To ensure a fair comparison between the three frame-pairing methods, we use a common spectral conversion technique to map a native source speaker's spectral features to match a non-native target speaker's voice quality. Following Toda et al. [21], we use a GMM to model the joint distribution of source and target frame pairs, and then use maximum likelihood parameter generation (MLPG) with global variance (GV) [102] to generate the converted speech for a given source utterance. Specifically, we use $2D$-dimensional acoustic features, $X_t = [x_t^T, \Delta x_t^T]^T$ from the source speaker, and $Y_t = [y_t^T, \Delta y_t^T]^T$ from the target speaker, consisting of D-dimensional static and dynamic features, where $(\cdot)^T$ denotes the transpose. Given the paired source and target features, we train a GMM to model the joint probability density $p(X, Y|\Theta)$ where Θ denotes model parameters, estimated using Expectation-Maximization (EM):

$$\Theta = \mathrm{EM}\left(\underset{\Theta}{\operatorname{argmax}}\, p(X, Y|\Theta)\right). \tag{3.9}$$

When converting source static and dynamic feature vectors $X = [X_1^T, X_2^T, \dots, X_T^T]^T$ to the target static feature vectors $y = [y_1^T, y_2^T, \dots, y_T^T]^T$ – after the GMM is trained, we maximize the function below with respect to y,

$$\hat{y} = \underset{y}{\operatorname{argmax}} \log\{p(Y|X,\Theta)^{\omega} p(v(y)|\theta_v)\}, \qquad Y = Wy, \tag{3.10}$$

where $p(Y|X,\Theta)$ denotes the conditional probability density function (PDF) on the target static and dynamic feature vectors, and $p(v(y)|\theta_v)$ represents the likelihood of a PDF on the global variance of the target feature vectors, which is represented as a separate GMM (one mixture) and trained using the EM algorithm as well. W is a matrix that appends dynamic features to the static features, and ω adjusts the relative importance between the two distributions and is set as the ratio of number of dimensions between vectors $v(y)$ and Y $(= 1/2T)$. We use a GMM instead of a DNN in this study to focus on low-resource accent conversion scenarios – in real pronunciation training applications, we generally have a limited amount of data from the non-native speakers.

3.4.2.4. Pitch scaling

Previous studies [12, 15, 85] have shown that prosody modification is an essential part of accent conversion, and the pitch contour contains identity-related information. Since pitch modification is not the focus of this study, we follow the standard procedure [21] and use the pitch trajectory from the source (native) speaker, which captures native intonation patterns, then normalize it to match the pitch range of the target (non-native) speaker using mean and variance normalization in the $\log F_0$ space.

3.5. Experimental setup

3.5.1. DNN acoustic model for extracting PPG

To train the DNN acoustic model, we used Kaldi's Librispeech recipe[5]. The model is a p-norm DNN (p=2), as introduced in the method section, with five hidden layers. We extracted 13-dim MFCC vectors with a 7-frame context, passed the concatenated 91-dim (13×7) MFCCs through a Linear Discriminant Analysis (LDA) to generate a 40-dim input feature vector, then concatenated nine frames of such 40-dim LDA features as the final input to the DNN. The 360-dim (40×9) input features were de-correlated using a fixed linear transform. All hidden layers had 5,000 hidden neurons and output 500 activations because each p-norm non-linearity was computed over ten hidden neurons. Every hidden layer was fully-connected with the previous layer. Right after the last hidden layer was a softmax layer of 14,000 nodes; those nodes were then "group-summed" to produce the final output across senones (5,816 dimensions, which were obtained from state-tying on a phonetic decision tree built from the transcripts of the training data; see [103] for more details on how the decision tree was constructed). The DNN acoustic model was trained on Librispeech's [104] training set, a speech recognition corpus that contains 960 hours of native English speech, the majority being American English. In the following experiments, the Librispeech corpus was used solely for building the acoustic model.

[5] https://github.com/kaldi-asr/kaldi/tree/master/egs/librispeech

3.5.2. Speech corpus for accent conversion

For the native speech synthesis corpus, we used two speakers from the CMU ARCTIC dataset [105]: BDL and CLB. Those recordings have a sampling rate of 16 kHz. For the non-native (L2) English speech synthesis corpus, we used five non-native speakers from the L2-ARCTIC corpus[6] [106]: two native Hindi speakers, two native Korean speakers; and one native Arabic speaker. Each non-native speaker produced the full ARCTIC dataset (~1100 utterances; around one hour of speech). The speech was recorded in a quiet room at 44.1 kHz. For the following experiments, we down-sampled all the non-native speech data to 16 kHz using sox[7]. The speaker demographic information is summarized in Table 3.1. For the non-native speakers, their English proficiency level was measured in their TOEFL iBT scores[8] [109].

Table 3.1: Demographic information of the speakers.

Speaker	*Gender*	*Native Language*	*English Proficiency*
BDL	M	English	Native
CLB	F	English	Native
RRBI	M	Hindi	91
TNI	F	Hindi	99
HKK	M	Korean	114
YKWK	M	Korean	N/A
ABA	M	Arabic	94-101

[6] https://psi.engr.tamu.edu/l2-arctic-corpus/
[7] http://sox.sourceforge.net/Main/HomePage
[8] Speaker ABA only reported his IELTS [107] score (7.0). We converted it to a TOEFL iBT score following [108].

3.5.3. System configurations

In what follows, we will refer to the proposed frame-pairing algorithm, baseline 1 (acoustic similarity), and baseline 2 (dynamic time warping) as **AC-PPG**, **AC-SIM**, and **AC-DTW**, respectively.

We used the TANDEM-STRAIGHT vocoder[9] [112] to decompose speech into aperiodicity (AP), F_0, and a 513-dim spectral envelope. Then, we computed 25-dim MFCCs[10] from the spectral envelopes to learn the VTLN transform and pair frames using acoustic similarity (AC-SIM); see Section 3.4.2.2. AC-DTW also used those MFCCs (excluding MFCC_0) to time-align a source speaker to a target speaker. AC-PPG used the 5816-dim PPGs extracted by the acoustic model to perform frame pairing.

We also computed 25-dim MCEPs from the spectral envelopes as the acoustic feature (excluding MCEP_0 since it is energy) to train the spectral conversion models (GMMs) and convert speech from the native speaker to the non-native speaker. MCEPs from the two speakers were frame paired using the three methods (AC-PPG, AC-SIM, AC-DTW) before being fed to the GMMs. Following Aryal and Gutierrez-Osuna [16], all GMMs had 128 mixture components with diagonal covariance matrices. Input features to the GMM include delta features, and therefore the joint feature vectors had 96 dimensions. Once we converted the native speaker's MCEPs to the non-native speaker's space, we reconstructed

---

[9] We used the NDF F_0 extractor [110] instead of the default F_0 extractor that comes with TANDEM-STRAIGHT, because based on our experience and a prior study [111], the NDF F_0 extractor is more robust than the TANDEM-STRAIGHT default.

[10] We only used those MFCCs to generate the frame pairing lookup tables in **AC-SIM** and **AC-DTW** and discarded in other tasks

the spectrogram from the converted MCEPs ($MCEP_0$ being copied from the native speaker), and combined it with the native speaker's AP and normalized F_0 to synthesize speech using the TANDEM-STRAIGHT vocoder. The conversion pipeline is illustrated in Figure 3.4.

All experiments were conducted on a desktop running Windows 10 with an Intel Core i7-7700K CPU @ 4.2GHz, 16GB of memory, and an NVidia GTX 1070 GPU. Most of the algorithms were implemented and run on Matlab v9.3, except for the acoustic model and PPGs, which were computed using Kaldi on Ubuntu 16.04.

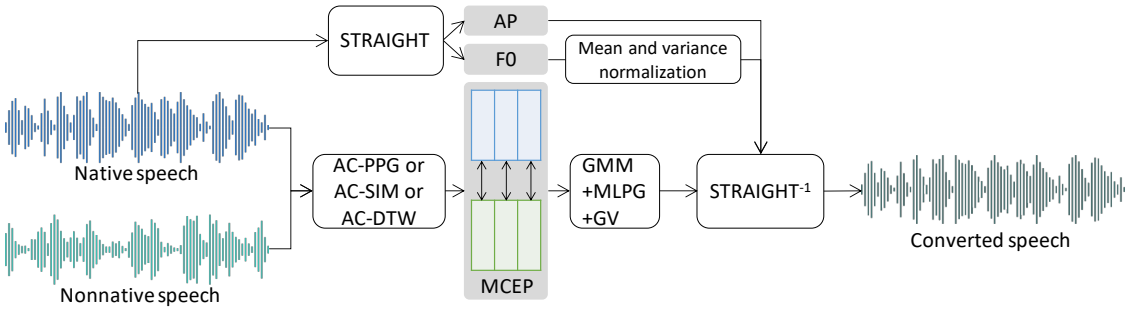

Figure 3.4: Accent conversion workflow; frame pairing can be AC-PPG, AC-SIM (baseline 1), or AC-DTW (baseline 2).

3.6. Results

We conducted three sets of perceptual listening studies to evaluate different properties of the proposed frame-pairing algorithm. In the first experiment, we compared the approach against the two baseline systems by its ability to reduce perceived accents while matching the voice quality of the non-native speakers. In the second experiment, we evaluated whether the approach could also be used for the reverse purpose, i.e., to impart a

non-native accent to a native speaker's voice. In the third and final experiment, we evaluated the approach to perform accent conversion using non-parallel speech corpora.

We recruited anonymous human participants from Amazon's Mechanical Turk platform[11] for our listening tests. Following Buchholz and Latorre [113], all listening tests included calibration trials designed to be easy to judge, and we used the participants' responses on those calibration trials to detect cheating behaviors. We excluded data from participants whose responses were below chance level on those calibration questions. All participants' calibration responses were excluded from the final analyses. In addition, and following [82], all human subjects passed a screening test that consisted of identifying various American English accents. We compensated participants for their time at an hourly rate of eight USD. In all experiments, the reference native and non-native English speech were resynthesized from their MCEPs using TANDEM-STRAIGHT to keep their acoustic quality comparable with the converted speech, which went through the same vocoder compression. When selecting testing samples, we always randomly draw from the available pools, i.e., we did not cherry-pick the audio clips. All test trials were randomly presented. For any listening tests that required pairwise comparisons, the presentation order within an utterance pair was counterbalanced. Unless otherwise noted, we used paired-sample t-tests for the analyses.

[11] https://www.mturk.com/

3.6.1. Experiment 1: Comparing AC-PPG against baselines

In this experiment, we considered five native to non-native speaker pairings for accent conversion: BDL to RRBI, BDL to HKK, BDL to YKWK, BDL to ABA, and CLB to TNI. For each speaker pair, we used 100 parallel utterances for training and 50 utterances for testing; there was no overlap between the two sets. We performed accent conversion on all 50 test utterances using models trained on each of the three frame-pairing algorithms, i.e., AC-PPG, AC-SIM, and AC-DTW.

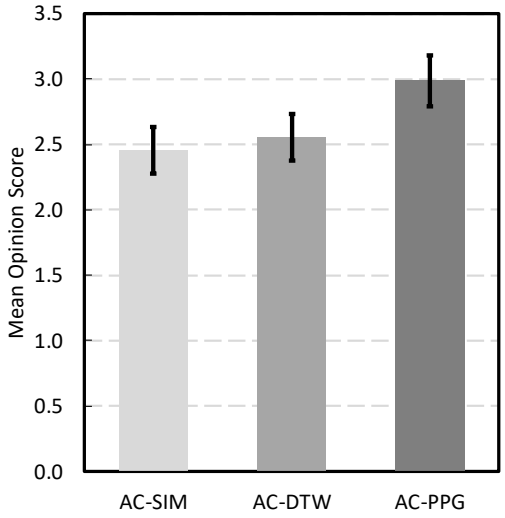

Figure 3.5: Mean Opinion Scores for the proposed method (AC-PPG) and the two baseline methods (AC-SIM, AC-DTW); the error bars show 95% confidence intervals.

Acoustic quality. We used a standard five-point (1-Bad, 2-Poor, 3-Fair, 4-Good, 5-Excellent) Mean Opinion Score (MOS) to rate the acoustic quality of the synthesized speech. Thirty listeners rated 150 test samples: 50 per system, 10 utterances per conversion direction. Results are shown in Figure 3.5. The proposed method (AC-PPG) received a

MOS rating of 2.99, which was significantly higher than either baseline: AC-SIM (2.45 MOS, 22% relative improvement; $t(29) = 15.61, p \ll 0.001$; one-tail) and AC-DTW (2.55 MOS, 17% relative improvement; $t(29) = 12.04, p \ll 0.001$; one-tail). These results suggest that the proposed algorithm can boost the acoustic quality of the converted speech using exactly the same training data without even having to modify the GMM training and spectral conversion methods.

Voice quality. Following our prior work [95], we used a voice similarity score (VSS) ranging from -7 (definitely different speakers) to +7 (definitely same speaker) to assess the speaker's voice quality. Twenty-six participants rated 150 utterance pairs: 50 pairs per system (25 AC-L1 and 25 AC-L2 pairs, each pair contained one AC and one L1 [native]/L2 [non-native] utterance), and ten pairs per conversion direction. Following Felps et al. [12], we played utterances in reverse to prevent the accent from interfering with the perception of voice quality. In each trial, listeners first answered whether both utterances were produced by the same speaker (+1) or different speakers (-1), and then rated their confidence level on a 7-point scale (1-Not at all confident, 7-Extremely confident). The VSS was then compiled by multiplying the response from the first question with the confidence rating. Results are summarized in Figure 3.6.

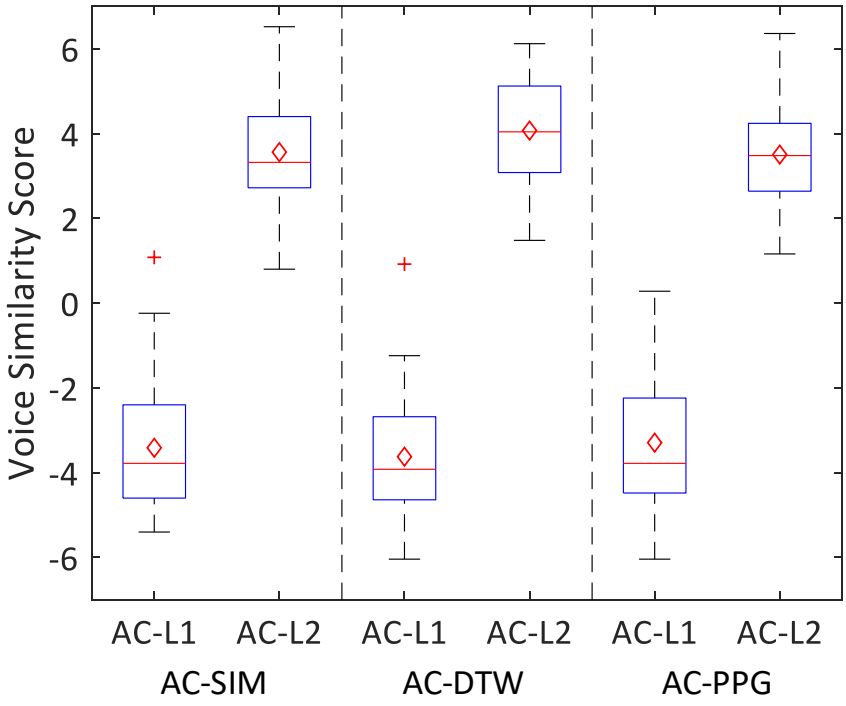

Figure 3.6: Voice quality results; AC-L1: VSS between AC and native (L1) speaker; AC-L2: VSS between AC and non-native (L2) speaker; the middle bars in the boxes show the median values and diamond markers (◊) show the mean values, the plus signs (+) indicate outliers, those notations apply to all boxplots in this chapter.

Overall, the three systems have similar VSS, and AC-L1 pairs received an average VSS between -3.29 to -3.62, indicating that listeners were "*confident*" that the AC utterances had a different voice quality from those of the native speaker. Likewise, AC-L2 pairs received an average VSS between 3.50 to 4.07, indicating that listeners were "*confident*" that the same speaker produced the AC and L2 utterances. When analyzing the AC-L1 pairs, we found no significant differences in VSS between AC-PPG and either baseline (AC-PPG:AC-SIM $t(25) = 1.13, p = 0.27$; AC-PPG:AC-DTW, $t(25) = 1.95, p = 0.06$; two-tail). These results suggest that the three methods are equivalent in terms of producing

speech that is different from the native speaker. When analyzing AC-L2 pairs, we found no significant difference between AC-PPG and AC-SIM ($t(25) = 0.42, p = 0.68$, two-tail), suggesting that the new accent conversion algorithm did not sacrifice the speaker's voice quality. However, AC-DTW achieved a higher VSS (4.07) than AC-PPG (3.50); one-tail t-test ($t(25) = 3.59, p \ll 0.05$). One possible explanation for this result is that listeners still picked up subtle cues of non-native accent in the AC-DTW speech samples, and used it to rate voice quality. Because AC-DTW only performs voice conversion, it retains some of the non-native speaker's accent. This residual non-native accent may have led listeners to rate samples from AC-DTW as more similar to the non-native speech, even though the recordings were played backwards. This explanation is consistent with prior studies [114, 115] showing that, even when speech is played backwards, native English speakers can still detect non-native English accents.

Non-native accentedness. We used a preference test to determine if AC-PPG does indeed make the converted speech sound more native-like. Thirty native English speakers rated 150 utterance pairs: 50 pairs for each comparison: AC-PPG vs. AC-SIM, AC-PPG vs. AC-DTW, and AC-PPG vs. L2 (i.e., original utterances from the non-native speaker), ten pairs of utterances per conversion direction, each utterance pair was from the same sentence. Listeners were asked to choose the most native-like (least foreign) utterance from each pair, and then rate their confidence level using a seven-point scale (1-Not confident at all, 7-Very confident). Aryal and Gutierrez [16] had previously established that AC-SIM outperforms AC-DTW and L2 in this task; therefore, we omitted those comparisons in this study.

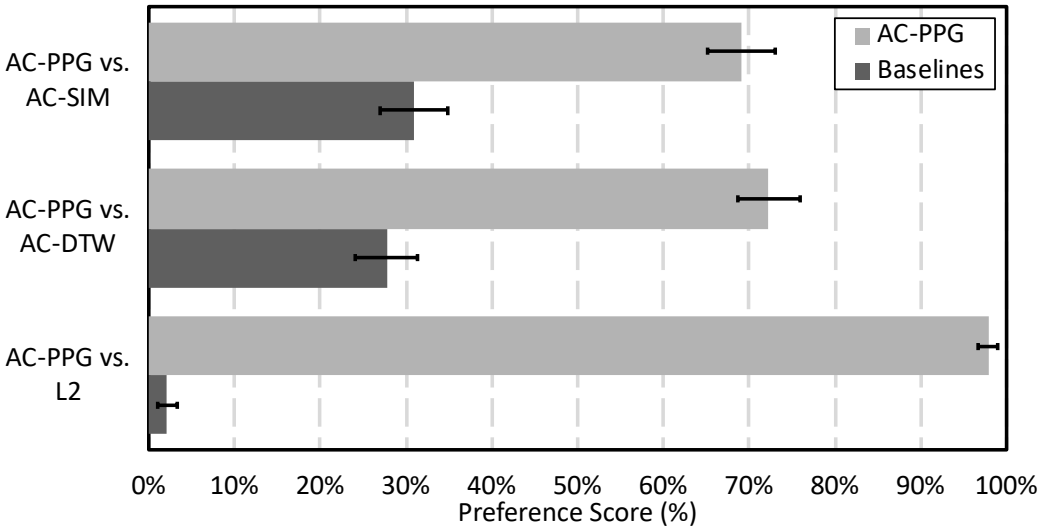

Figure 3.7: Accent preference score with 95% confidence interval.

In a first analysis, we sought to determine if a particular system was preferred as "less-accented" and compared the *preference ratings* from the participants. Results are summarized in Figure 3.7. On average, listeners were very confident (mean: 98%, STD: 3%) that the AC-PPG conversions were more native-like than the original non-native utterances. More importantly, listeners were positive that AC-PPG outperformed both AC-SIM (mean: 69%, STD: 11%) and AC-DTW (mean: 72%, STD: 10%). All the above preference scores are statistically significant ($p \ll 0.001$; one-tail) compared with chance levels (50%). Since preference tests sometimes are too coarse and will mask out nuances in raters' attitudes, we further used the *confidence ratings* to compute a more detailed measurement – the cumulative confidence score (CCS) [116]. The CCS for each system in each comparison pair was computed as follows. We treat each response as if it were assigning

a number of points to a system; for example, if a listener preferred the AC-PPG system and was "somewhat confident" (rated as three), then the AC-PPG system would receive three points. We then computed the average CCS that listeners allocated to each system. Therefore, the highest score a system can get is 350 points (7×50), within a comparison pair. Results are summarized in Figure 3.8. As shown, all comparison pairs have the same trend as in the preference test, with AC-PPG performing significantly better than both baselines. All differences in CCS were statistically significant ($p \ll 0.001$, one-tail).

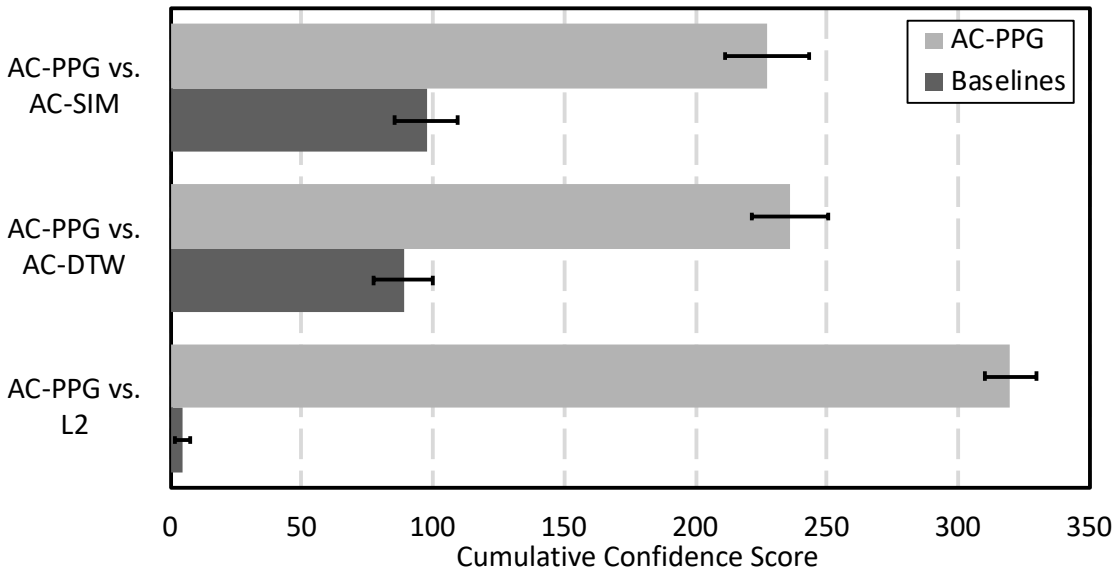

Figure 3.8: Cumulative confidence score for accentedness with 95% confidence interval.

3.6.2. Experiment 2: Native to non-native conversion

In a second experiment, we evaluated whether AC-PPG can perform the accent conversion task in the opposite direction – creating a voice that has the native speaker's

voice quality but speaking with a non-native accent. Prior work [117] has tackled this problem from a Text-To-Speech perspective, so we wanted to determine if it could also be achieved through accent conversion. Accordingly, for this experiment we performed accent conversion in five directions that were from non-native to native English speakers, i.e., RRBI to BDL, HKK to BDL, YKWK to BDL, ABA to BDL, and TNI to CLB. The training and testing data for all speakers were identical to those used for *Experiment 1*.

In an initial listening test, we recruited 20 subjects to rate the non-native English accent of the converted speech using a nine-point Likert-scale rating test [4], where 1 corresponded to "no accent" and 9 to "very strong accent." For each conversion direction, we randomly picked five utterances, and we made sure that the final 25 (5×5) utterances for evaluation were from different elicitation sentences. To provide a reference, we also included the same set of sentences that were uttered by the native and non-native speakers in the test. Therefore, all listeners rated 75 (25×3) sentences. Given that our native speakers (BDL and CLB) spoke American English, before the test, we instructed listeners to consider that "*A 'foreign accent' is defined as an accent that is different from the General American English accent*." We also provided two samples of American accent English that were produced by native speakers not used in this study. All listeners were geographically located in the United States and all but one listener self-reported to be native English speakers. The only listener whose native language is not English is a native Italian speaker who also speaks English and French, and since this participant passed our American accent pretest, we did not exclude this participant's responses. Results are summarized in Figure 3.9. On average, listeners rated the native speech to be 1.4 points (closer to "no accent")

and the non-native speech to be 6.4 points (closer to "very strong accent"). The accent-converted speech had an average rating of 6.2 points (closer to "very strong accent"), which was similar to the ratings of the non-native speech. No significant difference was found between accentedness ratings of non-native and AC speech ($t(19) = 0.82, p = 0.42$, two-tail). Therefore, this experiment indicates that our accent conversion approach was able to impart the non-native accent of the non-native speaker to utterances from a native speaker.

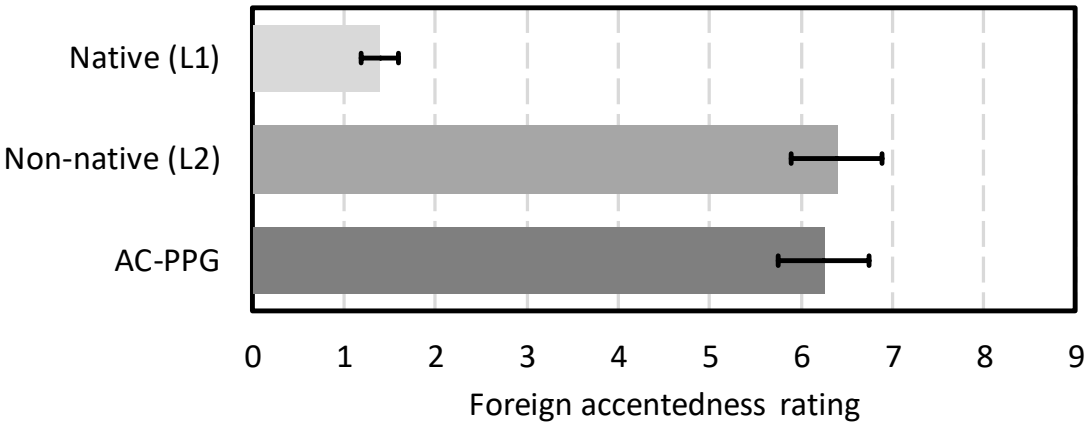

Figure 3.9: Foreign accentedness ratings for L1 (native English), L2 (non-native English), and AC speech; the error bars show 95% confidence intervals.

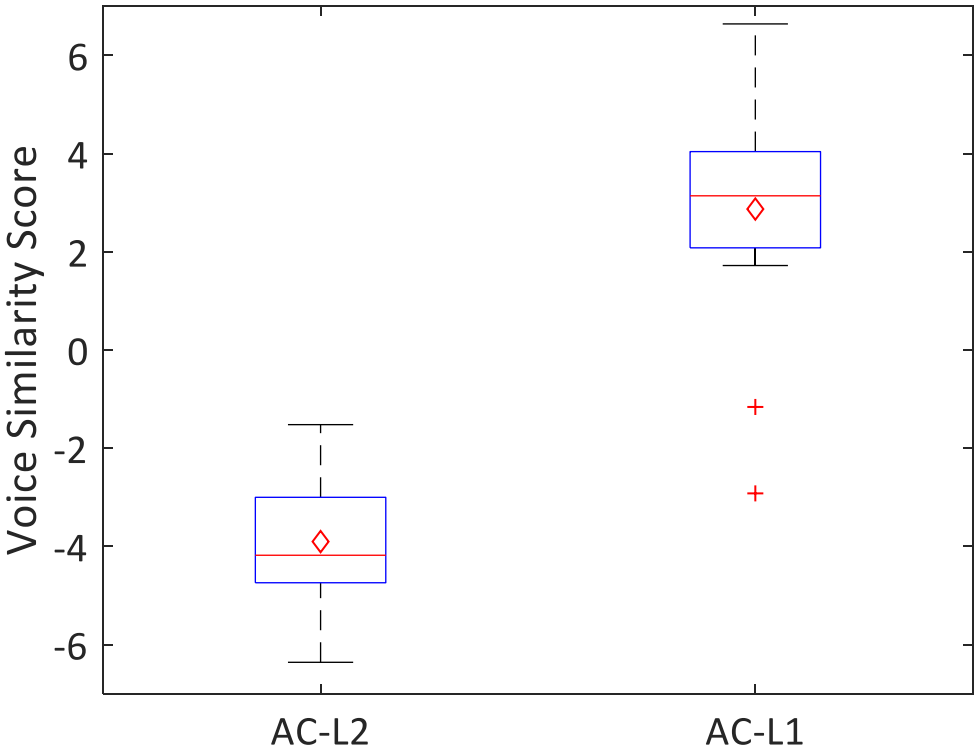

Figure 3.10: Voice similarity score for AC-L1 and AC-L2 comparisons.

In a second listening test, we focused on evaluating whether the converted speech retained the voice quality of the original (native) speaker. Accordingly, we used the same VSS test as in *Experiment 1* to produce voice similarity scores between AC sentences and the original native/non-native sentences. Twenty listeners rated 50 utterance pairs, among which 25 were AC-L1, and the rest were AC-L2 pairs. As before, we randomized all presentation order and played the recordings in reverse. Results are summarized in Figure 3.10. Listeners were "confident" that AC utterances had the voice quality of the native speakers (mean AC-L1 VSS score 2.87), and was different from the non-native speaker (mean AC-L2 VSS score -3.90) despite that they share the same accent. Considering the

results from both listening tests in this experiment, we can conclude that AC-PPG is able to impart a non-native accent to native voices.

3.6.3. Experiment 3: AC-PPG using non-parallel training data

Our method does not impose timing constraints when pairing native and non-native speech frames: an acoustic frame from the native speaker is paired with a frame in the non-native speaker's training set by minimizing the symmetric KL divergence between their respective PPGs. Thus, in principle, our method removes the constraint that native and non-native speakers must produce the same set of utterances. This property is particularly useful for real-world applications because it allows more flexibility when recording training sentences. Therefore, in a third and final experiment we evaluated the AC performance by comparing two variants of our method:

- **AC-PPG-P**: the same system used in *Experiment 1*, i.e., using parallel sentences as the training data;
- **AC-PPG-NP**: a system that used non-parallel sentences. For this purpose, we randomly selected 100 native training utterances that were different from those in the non-native training or non-native test sentences. As a result, the native and non-native speakers never uttered any common sentence. All other configurations for this system were the same as AC-PPG-P.

The AC directions and test sentences were the same as those used in *Experiment 1*. For each system, we generated accent converted sentences from all 50 testing samples for evaluation.

In a first listening study, we used a preference test to determine which system yielded better acoustic quality. Twenty participants rated 50 utterance pairs – one from AC-PPG-P and the other from AC-PPG-NP, both utterances having the same linguistic content. We randomly selected 10 utterance pairs from each AC direction. For each pair, participants were asked to pick the utterance that has the best acoustic quality. The test allowed them to choose "no preference" as their response. Results are summarized in Figure 3.11. The majority of the votes (40.3%) reflected no difference between the acoustic quality of the two systems ("no preference"), and both systems received a similar percentage of votes (29.7% for AC-PPG-P; 30.0% for AC-PPG-NP). We found no significant difference in terms of acoustic quality between using parallel or non-parallel data ($t(19) = 0.11, p = 0.91$, two-tail).

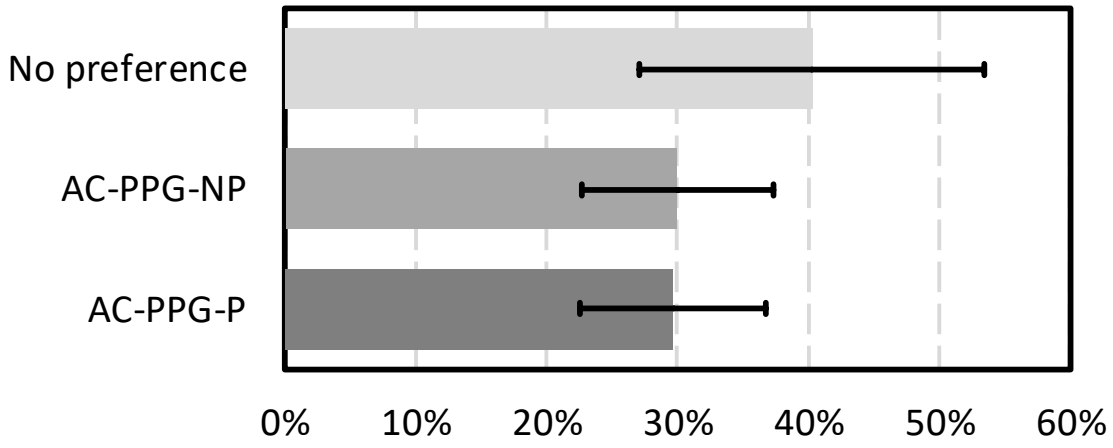

Figure 3.11: Preference scores for comparing the acoustic quality of AC-PPG-P and AC-PPG-NP; the error bars display the 95% confidence intervals.

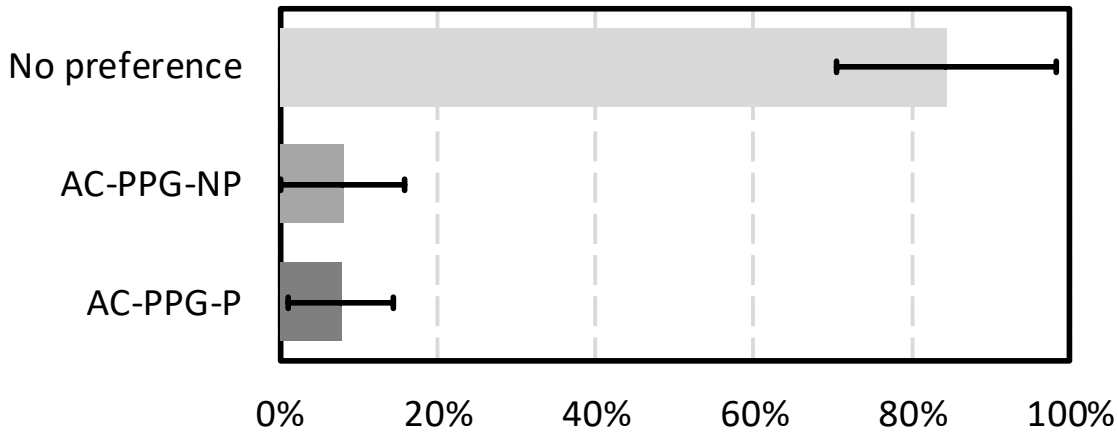

Figure 3.12: Preference scores for comparing foreign accentedness of AC-PPG-P and AC-PPG-NP; the error bars display the 95% confidence intervals.

In a second listening test, we investigated whether using non-parallel data would affect the non-native ratings of the converted speech. The experimental protocol was the same as the one we used in the acoustic quality experiment, except that in this case, for each AC-PPG-P and AC-PPG-NP utterance pair, we asked participants to select the one that had the "least foreign accent." Twenty participants rated 50 utterance pairs, 10 pairs for each AC direction. Results are summarized in Figure 3.12. The vast majority of the votes (84.3%) indicated that there was no difference between the two systems. Furthermore, a t-test on the preference scores for AC-PPG-P (mean 7.7%) and AC-PPG-NP (mean 8.0%) revealed no significant differences ($t(19) = 0.17, p = 0.86$, two-tail).

Finally, we asked 21 listeners to rate AC-PPG-NP sentences in terms of voice quality. Each listener rated 50 converted utterances, where we randomly selected 10 utterances from all 5 conversion directions. The VSS scores are summarized in Figure 3.13. The average AC-L1 VSS is -2.64 (std: 1.52), and 3.07 (std: 1.48) for AC-L2. Using a two-

tail independent samples t-test assuming unequal variances[12], we found no significant difference between the average VSS for the AC-PPG system in Figure 3.6 and those in Figure 3.13. For AC-L1, the test gave $t(43) = 1.44, p = 0.16$. For AC-L2, the test yielded $t(40) = 1.06, p = 0.29$. Thus, this experiment verified that using non-parallel data still allows our frame-pairing technique to preserve the non-native speaker's voice quality in the converted speech.

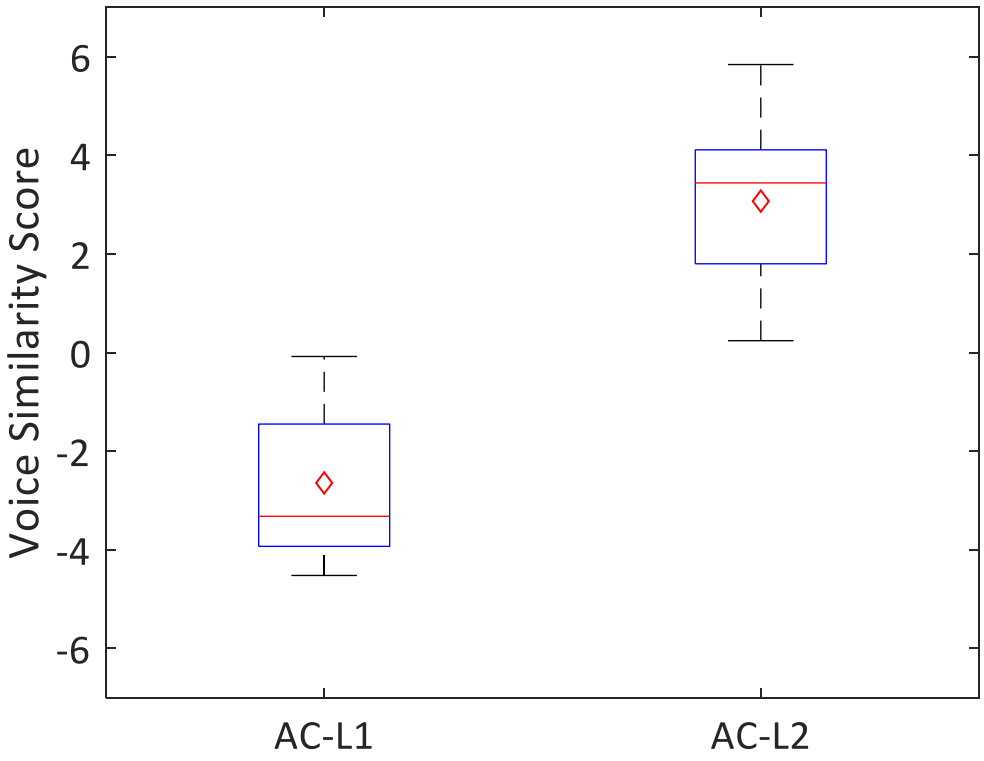

Figure 3.13: Voice similarity scores for AC-PPG-NP

[12] The two groups we are comparing have 26 (AC-PPG in Figure 3.6) and 21 (AC-PPG-NP) subjects respectively, therefore, it is not reasonable to assume that they have the same variance.

3.7. Discussion

In prior work [16], Aryal and Gutierrez-Osuna had shown that paring speech frames based on acoustic similarity (i.e., the AC-SIM baseline in our study), and then using the resulting frame pairs to train a voice conversion model could be used to create a voice that captured a native speaker's pronunciation and a non-native speaker's voice quality. Their method was able to achieve a significantly better accentedness rating compared with pairing frames using DTW, though the results were based on a single pair of speakers. During our internal evaluations (results not shown) with multiple pairs of speakers and several sets of non-native accents, we found that the speech generated by AC-SIM still contained noticeable mispronunciations. Since AC-SIM normalizes the vocal tract length difference between native and non-native speakers, we hypothesized that there remains a lot of other unattended speaker-dependent (SD) information in the VTLN-transformed acoustic feature space, which makes the resulting frame pairing not ideal. PPGs, on the other hand, are produced by speaker-independent (SI) acoustic models built for ASR. As a result, the most dominant information in PPGs is linguistic information. These analyses reinforced our intuition to use AC-PPG to eliminate the effects of SD cues in the frame pairing process.

The listening tests in *Experiment 1* show that the proposed frame pairing method can significantly reduce the non-native accent ratings compared with two baselines. In terms of *voice similarity* between the non-native speaker and the converted speech, AC-PPG performs as well as AC-SIM. Although the speech generated by AC-DTW was rated

more similar to the non-native speaker than AC-PPG, we suspected that it is hard to decouple the influence of *accent* and *voice quality* on the perceived *speaker identity* (refer to the introduction of this chapter for the difference between *voice quality* and *speaker identity*). Listeners may have used the remaining foreign accent in the AC-DTW utterances to select the *speaker identity* of the utterances instead of their *voice quality*. Therefore, an interesting future direction would be to design a new perceptual experiment protocol that can better decouple *voice quality* and *accent* in spoken sentences, compared with the current solution of playing audio in reverse.

Another interesting observation from *Experiment 1* is that despite using the same spectral conversion model as the two baseline systems, AC-PPG can significantly boost the acoustic quality of the synthesis. When comparing the speech syntheses from AC-PPG with the others, we did notice that there were fewer noises and artifacts. One possible explanation for this is that AC-PPG pairs frames with similar phonetic context. Therefore, frame pairs have similar spectral structures, making the statistical regression model for spectra estimation less likely to introduce odd shapes in the predicted spectral envelopes. Consequently, better spectral predictions lead to better synthesis quality. Future work could investigate if this property of AC-PPG generalizes to other statistical conversion models that take frame pairs as training input (e.g., deep neural networks [78, 92], direct waveform modification [77]).

Experiments 2 and *3* investigated other interesting aspects of the proposed frame-pairing method. *Experiment 2* verified that AC-PPG could also work in the opposite conversion direction – creating an artificial voice that has a native speaker's voice quality

while speaking in a foreign accent. This artificial voice can be useful for generating materials for perceptual studies. For example, it can map speech from speakers that have different accents to the same voice quality, therefore removing the impact of voice quality when comparing differences in accents. *Experiment 3* verified that we could use a non-parallel dataset to achieve the same accent conversion performance (measured in acoustic quality, accentedness, and voice quality) using AC-PPG. One possible reason why we could use non-parallel training data is that AC-PPG looks at a fine-grained context (95 ms in the current implementation)[13], and this context size is comparable with the duration of a vowel [118] or consonant [119] segment in American English. Therefore, as long as the two sets of training data from native and non-native speakers have a balanced phonetic distribution, the approach is indifferent to the actual word-level prompts. The non-parallel data constraint is much more relaxed than the widely used parallel constraint, making the proposed method applicable to real-world scenarios, where parallel data are scarce or tedious to obtain.

AC-PPG can run efficiently with careful optimization and GPU-based parallelization. In our experiments, it generally took no more than two minutes to compute the pairing between 100 training utterances (~5 minutes of speech) from the native and non-native speakers. Further reductions in computation time may be achieved via dimensionality reduction and clustering.

---

[13] Each frame of PPG feature looks at a larger context than the analysis window (25 ms), because the input to the acoustic model consists of nine frames of adjacent LDA feature, and each frame was computed from seven consecutive MFCC feature vectors (25 ms). Therefore, the total context for a frame of PPG feature is 9+7-1=15 consecutive analysis windows, which converts to 95 ms under a 5 ms window shift.

At present, our ratings of acoustic quality are on the low end of what state-of-the-art voice conversion systems can achieve [120]. This is largely due to the choice of voice conversion system used, i.e., a conventional GMM-based spectral conversion system as a case study, which was needed to ensure a fair comparison with our previous work [16]. Fortunately, our frame-pairing approach can be combined with other spectral conversion methods to produce higher quality speech synthesis. For example, instead of converting speech frame-by-frame, we could perform the conversion over a larger context (e.g., sequence to sequence conversion [28].) Using a larger conversion context is likely to increase the acoustic quality [26, 27]. More importantly, mispronunciations often occur at the segment level, which is beyond the scope of frame-level conversion, and contextual information has to be taken into consideration to accurately correct segmental pronunciation errors in accent conversion.

Another line of ongoing work in our group is to relax the non-parallel data constraint further to allow the use of cross-lingual training data. In preliminary experiments (not shown here), we successfully performed accent conversion using utterances recorded in the target speaker's native language to capture their voice quality[14].

3.8. Conclusion

We have proposed a new frame-pairing method based on the phonetic similarity between acoustic frames. To measure phonetic similarity, we map source and target

[14] In these preliminary experiments, we used native Brazilian Portuguese speakers from the SID dataset [121] as the target speakers. Since Portuguese share some phonological similarities with English [122], we used the acoustic model used in this study directly to produce the PPGs from native Portuguese speech. For future work and more general cases (e.g., languages from the Sino-Tibetan family), we have to include senones from the target speaker's native tongues in the acoustic modeling process.

frames into a phonetic posteriorgram space using speaker-independent acoustic models trained on a native English corpus. Through a series of perceptual studies, we have shown that merely changing the frame pairing method can lead to significant improvement in acoustic quality and "nativeness," while keeping the voice quality of the non-native speaker. Our results also show that the approach works well across multiple non-native speakers with different native tongues. Additionally, the proposed algorithm does not need parallel data for training, which is ideal for real-world applications. Our approach only requires 5-10 minutes of speech data from the non-native speaker, making it practical for pronunciation training in realistic settings [123]. The implementation of the proposed system can be found at https://github.com/guanlongzhao/ppg-gmm.

4. FOREIGN ACCENT CONVERSION BY SYNTHESIZING SPEECH FROM PHONETIC POSTERIORGRAMS*

4.1. Overview

Methods for foreign accent conversion (FAC) aim to generate speech that sounds similar to a given non-native speaker but with the accent of a native speaker. Conventional FAC methods borrow excitation information (F_0 and aperiodicity; produced by a conventional vocoder) from a reference (i.e., native) utterance during synthesis time. As such, the generated speech retains some aspects of the voice quality of the native speaker. We present a framework for FAC that eliminates the need for conventional vocoders (e.g., STRAIGHT, WORLD) and therefore the need to use the native speaker's excitation. Our approach uses an acoustic model trained on a native speech corpus to extract speaker-independent phonetic posteriorgrams (PPGs), and then train a speech synthesizer to map PPGs from the non-native speaker into the corresponding spectral features, which in turn are converted into the audio waveform using a high-quality neural vocoder. At runtime, we drive the synthesizer with the PPG extracted from a native reference utterance. Listening tests show that the proposed system produces speech that sounds more clear, natural, and similar to the non-native speaker compared with a baseline system, while significantly reducing the perceived foreign accent of non-native utterances.

4.2. Introduction

Foreign accent conversion [12, 16, 18] aims to create a new voice that has the voice quality[15] of a given non-native speaker and the pronunciation patterns (e.g., prosody, segmentals) of a native speaker. This can be achieved by combining accent-related cues from a native utterance with the voice quality of the non-native speaker. FAC has potential application in computer-assisted pronunciation training [10, 12, 123], where it could be used as a model voice to imitate.

The main challenge in FAC is to divide the speech signal into accent-related cues and voice quality. Multiple solutions have been proposed, including voice morphing [12-15], frame pairing [16, 84], and articulatory synthesis [17-20]. These approaches can reduce the accent of non-native utterances, but have various limitations. Voice morphing often generates voices that sound like a "third" speaker, one who is different from either speaker. Frame-pairing methods can synthesize speech that resembles the non-native speaker's voice but the syntheses retain some aspects of the native speaker's voice quality; this is because excitation information from the native speaker is used to synthesize the speech. Finally, articulatory synthesis needs specialized apparatus to collect articulation data, so they are not practical for real-world applications.

In this work, we propose to perform FAC in a speaker-independent phonetically-rich speech embedding: a phonetic posteriorgram (PPG) [24]. A PPG is defined as the

[15] In the context of FAC, we use voice quality to refer solely to the organic aspects of a speaker's voice, e.g., pitch range, vocal tract dimensions.

posterior probability that each speech frame belongs to a set of pre-defined phonetic units (phonemes or triphones/senones), which retain the linguistic and phonetic information of the utterance. Our approach works as follows. In a first step, we generate a PPG for the non-native speaker using a speaker-independent acoustic model that is trained on a large corpus of native speech. Then, we construct a sequence-to-sequence speech synthesizer that captures the voice quality of the non-native speaker. The synthesizer takes a PPG sequence from the non-native speaker as the input and produces the corresponding mel-spectrogram sequence as the output. Finally, we train a neural vocoder, WaveGlow [64], to convert the mel-spectrogram into a raw speech signal. During testing, we feed the synthesizer with a PPG sequence from a native utterance. The resulting output contains the native speaker's pronunciation patterns and the non-native speaker's voice quality. The overall workflow of the proposed system is shown in Figure 4.1.

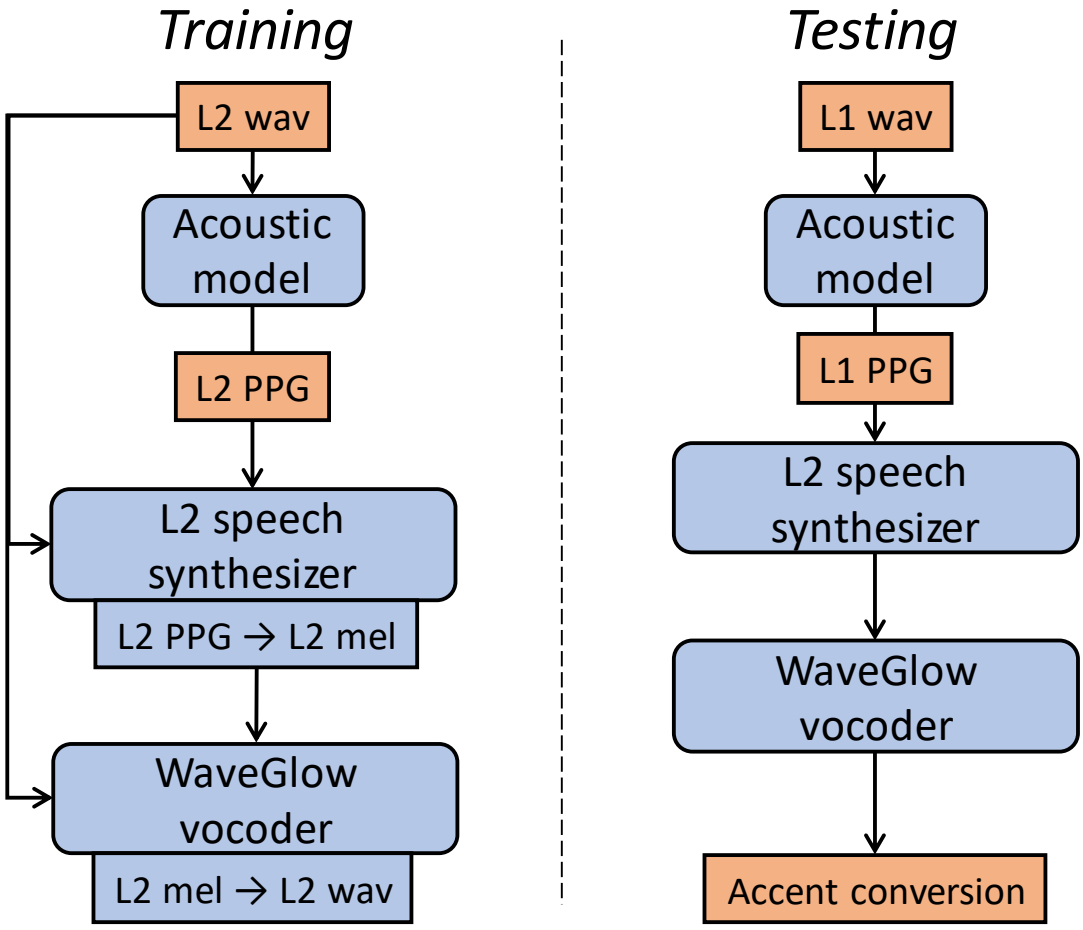

Figure 4.1: Overall workflow of the proposed system. L1: native, L2: non-native.

The proposed system has three advantages. First, it eliminates the need to borrow excitation information from the native reference speech, which prevents aspects of the native speaker's voice quality from leaking into the synthesized speech. Second, our system does not require any training data from the native reference speaker. Thus, we have the flexibility to use any reference voices during testing. Third, our system captures contextual information by means of a sequence-to-sequence model, which has shown state-of-the-art performance on multiple tasks [25, 28, 73], helping produce better audio quality.

4.3. Related work

Early attempts at accent conversion used voice morphing [12-15] to control the degree of accent by blending spectral components from the native and non-native speakers. In [86, 124], the authors used PSOLA to modify the duration and pitch patterns of accented speech. Aryal and Gutierrez-Osuna [16] adapted voice conversion (VC) techniques, replacing Dynamic Time Warping (DTW) with a technique that matched source and target frames based on their MFCC similarity after vocal tract length normalization. Later, Zhao et al. [84] used PPG similarity instead of MFCC similarity to pair acoustic frames.

PPGs have been applied to many tasks, e.g., neural-network-based speech recognition [80, 125], spoken term detection [24], mispronunciation detection [126], and personalized TTS [127]. PPGs have also gained much recent attention for VC. Xie et al. [83] divided PPGs from a target speaker into clusters and then mapped PPGs from a source speaker into the closest cluster of the target speaker. Sun et al. [128] used PPGs for many-to-one voice conversion. Miyoshi et al. [26] extended the PPG-based VC framework to include a mapping between source and target PPGs using LSTMs; they obtained better speech individuality ratings but worse audio quality than a baseline that did not include the PPG mapping process. Zhang et al. [25] concatenated bottleneck features and mel-spectrograms from a source speaker, then used a sequence-to-sequence model to convert the source mel-spectrograms into those of the target speaker, and finally recovered the speech waveform using a WaveNet [62] vocoder. Their model required parallel recordings and needed to train a new model for each speaker pair. They then applied text supervision

[129] to resolve some of the mispronunciations and artifacts in the converted speech. Recently, Zhou et al. [130] adopted bilingual PPG for cross-lingual voice conversion.

4.4. Method

Our system is composed of three major components; a speaker independent acoustic model (AM) that extracts PPGs, a speech synthesizer for the non-native speaker that converts PPGs into mel-spectrograms, and a WaveGlow vocoder to generate speech waveform from the mel-spectrograms in real-time.

4.4.1. Acoustic modeling and PPG extraction

We use a DNN with multiple hidden layers and the p-norm non-linearity as the AM. We train the AM on a native speech corpus [104] by minimizing the cross-entropy between outputs and senone labels obtained from a pre-trained GMM-HMM forced aligner. Training on native speech is critical for our task because the native and non-native frames have to be matched in a native phonetic space. For more details about the AM, please refer to [97].

4.4.2. PPG-to-Mel-spectrogram conversion

We convert PPGs from the non-native speaker into their corresponding mel-spectrograms using a modified Tacotron 2 model [74]. The original Tacotron 2 model (shown in Figure 4.2) takes a one-hot vector representation of characters and passes it to an encoder LSTM that converts it into a hidden representation, which is then passed to a decoder LSTM with a location-sensitive attention mechanism [71] that predicts the mel-spectrogram.

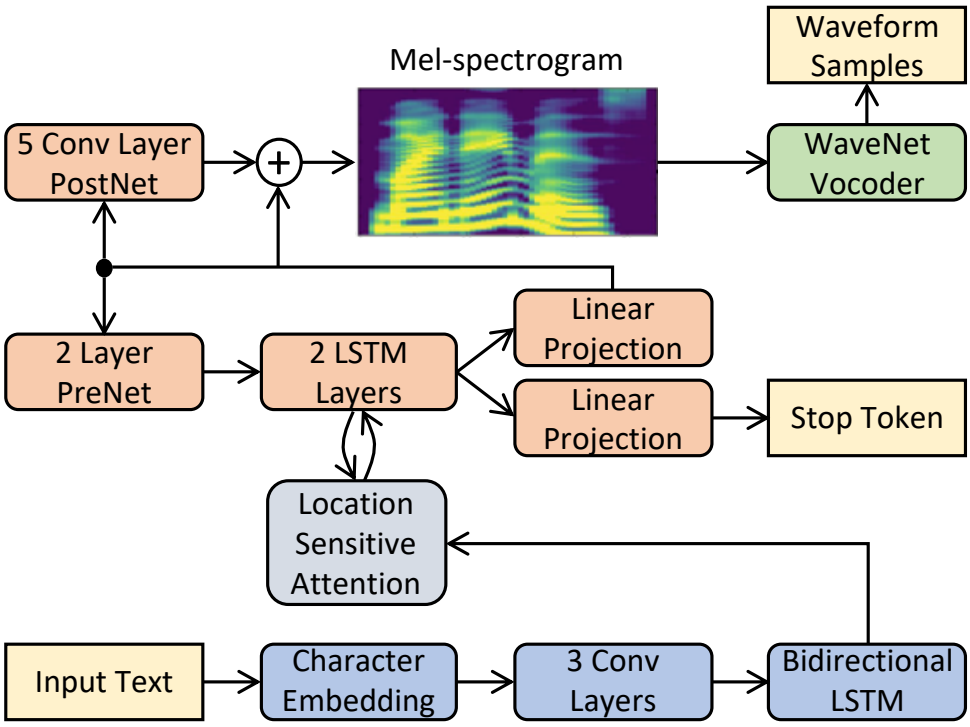

Figure 4.2: The original Tacotron 2 model architecture. Characters (represented by one-hot vectors) are passed to an encoder Bi-LSTM and a decoder LSTM with a location-sensitive attention mechanism to predict the mel-spectrogram. The speech waveform is generated by a WaveNet vocoder. A stop token is also predicted to determine when to stop the prediction.

To improve model performance, the character embedding is passed through multiple convolution layers before being fed to the encoder LSTM. The decoder appends a PreNet (two fully connected layers with the ReLU activation; Figure 4.3 (a)) before passing the predicted mel-spectrogram to the attention and decoder LSTM to extract structural information. It also applies a PostNet (five 1-D convolutional layers with the tanh activation; Figure 4.3 (b)) after the decoder to predict spectral details and add them to the raw prediction.

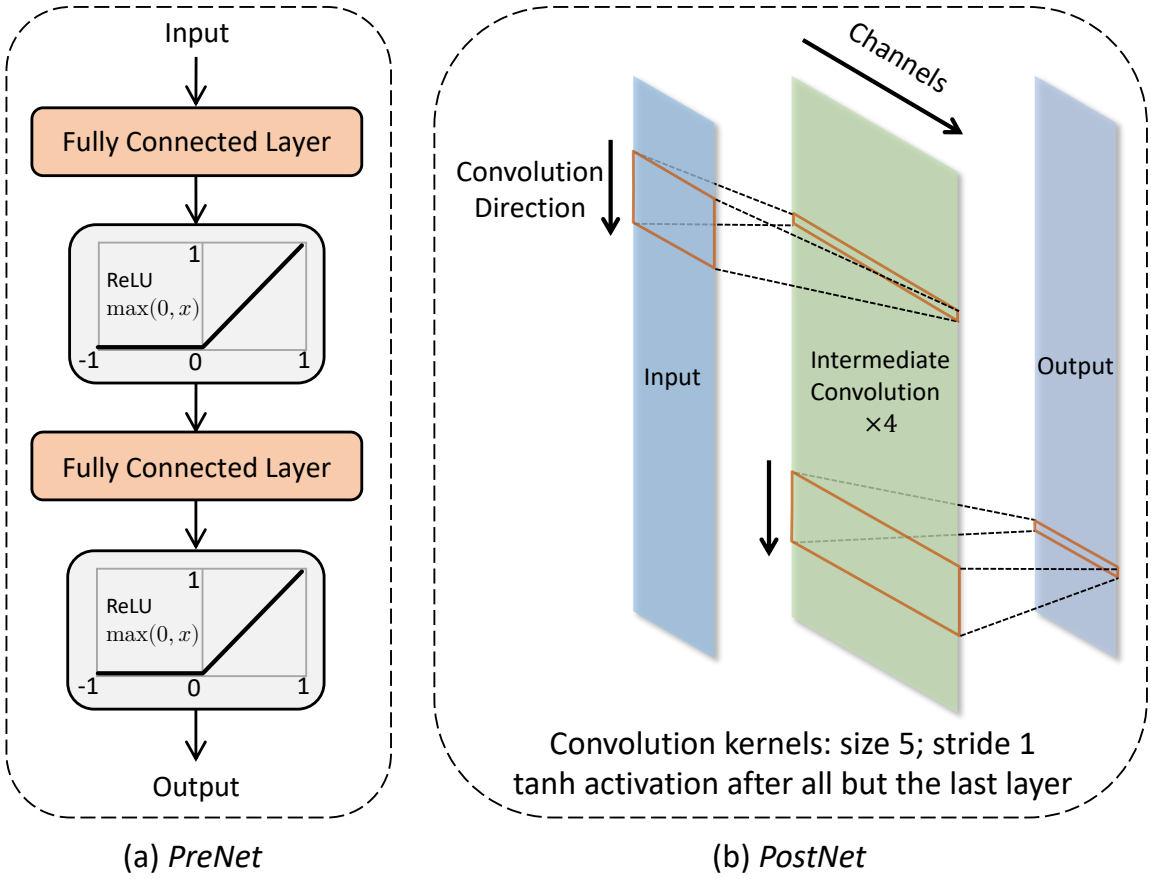

Figure 4.3: (a) PreNet: Two filly connected layers with the ReLU activation. (b) Post-Net: Five 1-D convolutional layers; kernel size 5, stride 1; tanh activation after all but the last layer. When the input is the mel-spectrogram, the convolution kernels move along the time axis one frame at a time, convolving five consecutive frames.

In this work, we replace the character-embedding layer with a PPG-embedding network (PPG PreNet; same model building block as Figure 4.3 (a)), which contains two fully connected hidden layers with the ReLU nonlinearity. This PPG-embedding network is similar to the PreNet in Tacotron 2 and transforms the original high-dimensional input PPGs into lower dimensional bottleneck features. This step is essential for the model to converge. The PPG-to-Mel conversion model is illustrated in Figure 4.4.

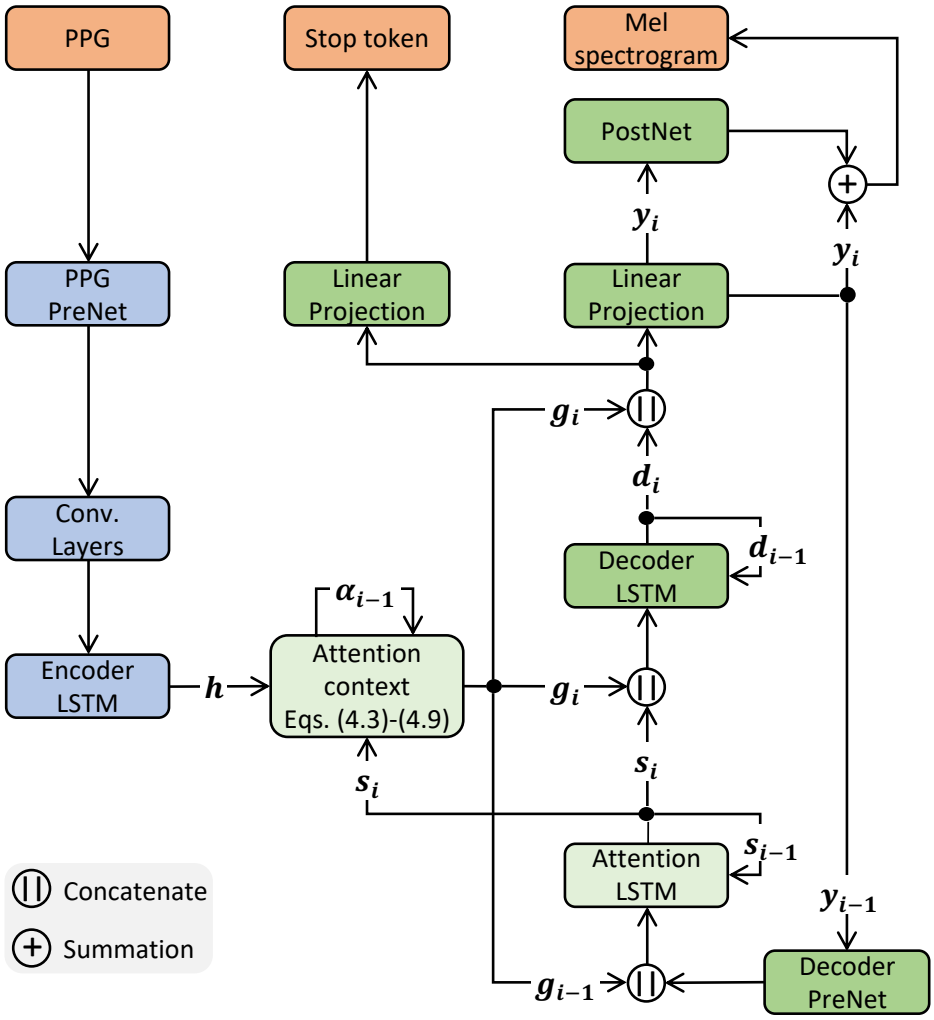

Figure 4.4: PPG-to-Mel conversion model.

The original Tacotron 2 was designed to accept character sequences as input, which are significantly shorter than our PPG sequences. For example, each sentence in our speech corpus [106] contains an average of 41 characters, whereas the PPG sequence has a few hundred frames. Therefore, the original Tacotron 2 attention mechanism would be confused by such long input sequences and cause misalignment between the PPG and

acoustic sequences, as pointed out in [25]. As a result, the inference would be ill-conditioned and would generate non-intelligible speech. One solution to this issue is to train the PPG-to-Mel model with shorter PPG sequences. For example, one could use word segments instead of sentences. However, this solution has several issues. First, to obtain accurate word boundaries, we need to perform forced alignment on the training sentences, which requires access to the transcription. Second, and more importantly, training with short segments and performing inference with significant longer input sequences leads to model failure, as observed in [71].

We resolve this issue by adding a locality constraint to the attention mechanism. Speech signals have a strong temporal-continuity and progressive nature. To capture the phonetic context, we only need to look at the PPGs in a small local window. Inspired by this, at each decoding step during training we constrain the attention mechanism to look at a window in the hidden state sequence, instead of the full sequence. We formally define this constraint as follows. We suggest the audience to also refer to Figure 4.4 when reading the following paragraphs.

Let d_i be the output of the decoder LSTM at time step i, y_i be the predicted acoustic features (output after applying a linear projection on d_i), and $h = [h_1, ..., h_T]$ be the full sequence of hidden states from the encoder. Applying the location-sensitive attention mechanism, we have,

$$d_i = \text{DecoderLSTM}(d_{i-1}, s_i, g_i), \tag{4.1}$$

where s_i is the hidden state of the attention LSTM at the i-th time step, g_i is the attention context,

$$s_i = \mathrm{AttentionLSTM}\big(s_{i-1}, g_{i-1}, \mathrm{PreNet}(y_{i-1})\big), \tag{4.2}$$

$$g_i = \sum\nolimits_{j=1}^{T} \alpha_i^j h_j \,, \tag{4.3}$$

and,

$$\alpha_i = \mathrm{Attend}(s_i, \alpha_{i-1}, h) = [\alpha_i^1, \ldots \alpha_i^T], \tag{4.4}$$

$$\alpha_i^j = \frac{\exp(e_{ij})}{\sum_{j=1} \exp(e_{ij})}, \tag{4.5}$$

are the attention weights. The attention scores e_{ij} are computed as follows,

$$e_{ij} = v^T \tanh\big(W s_i + V h_j + U f_i^j + b\big) \,, \tag{4.6}$$

$$f_i = F * \alpha_{i-1} = [f_i^1, \ldots, f_i^T], F \in R^{k \times r}, \tag{4.7}$$

where v, W, V, U, b are learnable parameters of the attention module. F contains k 1-D learnable kernels with r-dims, and $f_i^j \in R^k$ is the result of convolving α_{i-1} at position j with F.

Now, to enforce the locality constraint, we only consider the hidden representation within a fixed window centered on the current frame, i.e., let,

$$\tilde{h} = [0, \ldots, 0, h_{i-w}, \ldots, h_{i+w}, 0, \ldots, 0], \tag{4.8}$$

where w is the window size, and let,

$$\alpha_i = \mathrm{Attend}\big(s_i, \alpha_{i-1}, \tilde{h}\big). \tag{4.9}$$

The loss function for training the PPG-to-Mel model is,

$$\begin{aligned} L = \alpha \| G_{mel} - P_{Decoder} \|_2 + \beta \| G_{mel} - P_{PostNet} \|_2 \\ + \gamma \mathrm{CE}(G_{stop}, P_{stop}), \end{aligned} \tag{4.10}$$

where G_{mel} is the ground-truth mel-spectrogram; $P_{Decoder}$ and $P_{PostNet}$ are the predicted mel-spectrograms from the decoder (after linear projection) and PostNet, respectively; G_{stop} is the ground-truth stop token, and P_{stop} is the predicted stop token value; $\mathrm{CE}(\cdot)$ is the cross-entropy loss; α, β, γ control the relative importance of each loss term.

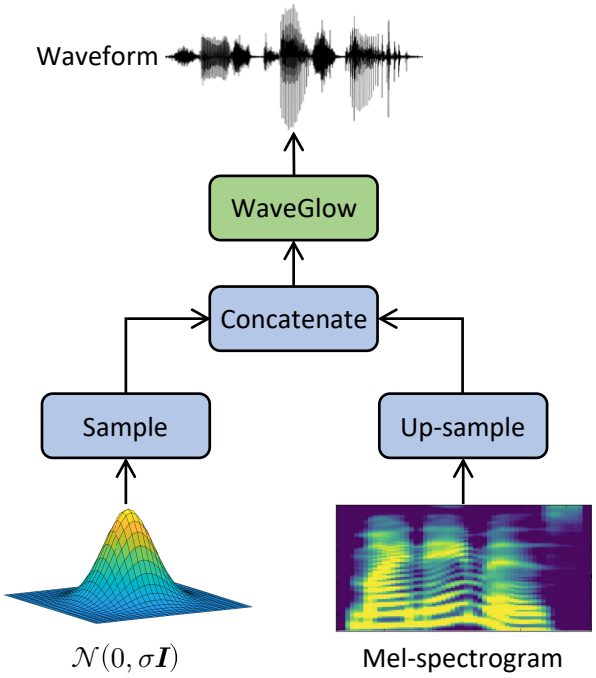

Figure 4.5: The WaveGlow vocoder. Random samples from a zero-mean spherical Gaussian (with variance σ) are concatenated with the up-sampled (matching the speech sampling rate) mel-spectrogram to predict the audio samples. In the plot, we use a 2D normal distribution for visualization; in practice, the vocoder may generate more than two samples at a time, e.g., the implementation we use produces eight audio samples at each step.

4.4.3. Mel-spectrogram to speech

We use a WaveGlow vocoder (Figure 4.5) to convert the output of the speech synthesizer back into a speech waveform. WaveGlow is a flow-based [131] network capable

of generating high-quality speech from mel-spectrograms (comparable to WaveNet). It takes samples from a zero-mean spherical Gaussian (with variance σ) with the same number of dimensions as the desired output and passes those samples through a series of layers that transform the simple distribution to one that has the desired distribution. In the case of training a vocoder, we use WaveGlow to model the distribution of audio samples conditioned on a mel-spectrogram. During inference, random samples from the zero-mean spherical Gaussian are concatenated with the up-sampled (matching the speech sampling rate) mel-spectrogram to predict the audio samples. WaveGlow can achieve real-time inference speed using only a single neural network, whereas WaveNet takes a long time to synthesize an utterance due to its auto-regressive nature. For more details about the WaveGlow vocoder, we refer readers to [64].

4.5. Experimental setup

We used the Librispeech corpus [104] to train the AM. It contains 960 hours of native English speech, most of which from North America. The AM has five hidden layers and an output layer with 5816 senones. We trained the PPG-to-Mel and WaveGlow models on two non-native speakers, YKWK (native male Korean speaker) and ZHAA (native female Arabic speaker) from the publicly-available L2-ARCTIC corpus [106]. We applied noise reduction on the original L2-ARCTIC recordings using Audacity [132] to remove ambient background noise. For the native reference speech, we used two North American speakers, BDL (M) and CLB (F) from the ARCTIC corpus [105]. Each speaker in L2-ARCTIC and ARCTIC recorded the same set of 1132 sentences, or about an hour of

speech. For each L2-ARCTIC speaker, we used the first 1032 sentences for model training, the next 50 sentences for validation, and the remaining 50 sentences for testing. All audio signals were sampled at 16 kHz. We used 80 filter banks to extract mel-spectrograms with a 10ms shift and a 64ms window. The PPG was also extracted with a 10ms shift.

Table 4.1: The model details of the PPG-to-Mel synthesizer.

Module	***Parameters***
PPG PreNet	Two fully connected (FC) layers; 600 ReLU units 0.5 dropout rate [134]
Conv. Layers	Three 1-D convolution layers (kernel size 5) batch normalization [135] after each layer
Encoder LSTM	One-layer Bi-LSTM; 300 cells in each direction
Decoder PreNet	Two FC layers; 300 ReLU units; 0.5 dropout rate
Attention LSTM	One-layer LSTM; 300 cells; 0.1 dropout rate
Attention	v in eq. (4.6) has 150 dims; eq. (4.7), $k = 32$, $r = 31$
Decoder LSTM	One-layer LSTM; 300 cells; 0.1 dropout rate
PostNet	Five 1-D conv. layers; 512 channels; kernel size 5

The PPG-to-Mel model parameters are summarized in Table 4.1. We used a batch size of 6 and a learning rate of 1×10^{-4}. α, β, γ were empirically set to 1.0, 1.0, and 0.005, respectively. The window size w of the locality constraint of the attention mechanism was set to 20. We trained the model until the validation loss reached a plateau (~8h). For the WaveGlow models, we set σ to 0.701 during training and 0.6 during testing, as suggested by [64]. The batch size was 3 and the learning rate was 1×10^{-4}. The models were trained until convergence (~one day). All models were trained on a single Nvidia

GTX 1070 GPU. The AM was trained with Kaldi, and the other models were implemented in PyTorch and trained with the Adam optimizer [133]. For more details and audio samples, please refer to https://github.com/guanlongzhao/fac-via-ppg.

We compared our proposed system against a baseline from [84] that worked as follows. First, we computed the PPG for each native and non-native frame. Then, we used the symmetric KL divergence in the PPG space to pair the closest native and non-native frames. In a final step, we extracted Mel-Cepstral Coefficients (MCEPs) from the frame pairs to train a joint-density GMM (JD-GMM) spectral conversion as described in [21]. We then converted the native MCEPs using the JD-GMM to match the non-native speaker's voice quality. Finally, we used the STRAIGHT vocoder [112] to synthesize speech from the converted MCEPs combined with the native speaker's aperiodicity (AP) and F_0 (normalized to the non-native speaker's pitch range). We used the same 1032-utterance training set for the baseline system. The GMM contained 128 mixtures and full covariance matrices. We used 24-dim MCEPs (excluding MCEP_0) and the Δ features. All features were extracted by STRAIGHT with a 10ms shift and 25ms window. For each system, we generated accent conversion for speaker pairs BDL-YKWK and CLB-ZHAA.

4.6. Results

We conducted three listening tests to compare the performance of the systems: a Mean Opinion Score (MOS) test of audio quality and naturalness, a voice similarity test, and an accentedness test. All experiments were conducted on Amazon Mechanical Turk, and all participants resided in the U.S. For each test, 25 utterances per speaker pair (50 in

total) from each system were randomly selected. The presentation order of the samples was randomized in all experiments.

Table 4.2: MOS results with 95% confidence intervals.

Conversion	*Rating Type*	*Baseline*	*Proposed*
BDL-YKWK	Audio quality	3.23±0.11	**3.48±0.12**
	Naturalness	3.18±0.15	**3.59±0.15**
CLB-ZHAA	Audio quality	2.86±0.15	**3.58±0.14**
	Naturalness	2.66±0.13	**3.32±0.20**
ALL PAIRS	Audio quality	3.04±0.10	**3.53±0.09**
	Naturalness	2.92±0.12	**3.46±0.13**

Table 4.3: MOS ratings for original recordings.

Real speech	*Rating type*	*Rating*
ARCTIC	Audio quality	4.40±0.08
	Naturalness	3.54±0.11
L2-ARCTIC	Audio quality	3.98±0.09
	Naturalness	3.50±0.08

The MOS test rated the audio quality and naturalness of audio samples on a five-point scale (1-bad, 2-poor, 3-fair, 4-good, 5-excellent). Audio quality and naturalness MOS described how clear and human-like the speech was, respectively. The two measures were obtained from non-overlapping groups of listeners to avoid bias. Each audio sample received at least 17 ratings. Listeners also rated the same set of original ARCTIC and L2-ARCTIC recordings as a reference. Results are summarized in Table 4.2 and Table 4.3. It should be noted that in [136], we established that the baseline system's audio quality MOS

is around 0.4 higher than a conventional JD-GMM system that uses DTW for frame pairing. Therefore, our baseline is a stronger system than the conventional JD-GMM. In all cases, our system outperformed the baseline significantly in both audio quality and naturalness. Although the two systems have lower audio quality MOS than the original recordings, there is *no* significant difference between the proposed system and either the ARCTIC ($p = 0.35$) or L2-ARCTIC ($p = 0.54$) recordings on the naturalness MOS, using a two-tail two-sample t-test.

In the voice similarity test, listeners were provided with three utterances, the original non-native utterance and syntheses from the two systems, and were asked to choose which of the two syntheses sounded more like the non-native speaker. Participants were also asked to rate their confidence level on a 7-point scale (1-not at all confident, 7-extremely confident) when making a choice. Participants were instructed to ignore accent when performing the task. Presentation order of samples from the two systems was counter-balanced in each trial, and 17 participants rated the audio samples. Results are presented in Table 4.4. In 72.47% of the cases, listeners preferred the proposed system with a 3.4 confidence level (above "somewhat confident"), whereas in the remaining 27.53% of the cases, listeners chose the baseline with a much lower confidence level (1.05, or "not at all confident.")

Table 4.4: Voice similarity test results.

Measure	***Baseline***	***Proposed***
Preference	27.53$\pm$5.00%	**72.47$\pm$5.00%**
Confidence	1.05$\pm$0.21	**3.40$\pm$0.32**

In the accentedness test, participants were asked to rate the degree of foreign accent in a nine-point scale (1-no foreign accent, 9-very strong foreign accent), which is commonly used in the pronunciation literature [4]. Each audio sample was rated by 18 individuals. Results are summarized in Table 4.5. Original utterances from ARCTIC speakers were rated as "no foreign accent" (1.20), whereas original utterances from the L2-ARCTIC speakers were rated as heavily accented (7.17). Both the baseline (2.94) and proposed (3.93) systems reduced the foreign accent significantly compared with the L2-ARCTIC speech but were rated more accented than the native speech. Surprisingly, speech generated from our system was rated as more accented than that of the baseline system; see the discussion section for a potential explanation of this result.

Table 4.5: Accentedness ratings.

Baseline	***Proposed***	***ARCTIC***	***L2-ARCTIC***
2.94±0.30	3.93±0.30	**1.20±0.04**	7.17±0.17

4.7. Discussion and conclusion

The proposed accent-conversion system produces speech with better quality than the baseline system because it uses a state-of-the-art sequence-to-sequence model (a modified Tacotron 2) to convert PPGs into mel-spectrograms, and then utilizes a neural vocoder to generate audio directly from the mel-spectrogram. This process takes advantage of the temporal-dependent nature of speech signals and avoids the use of conventional signal-processing based vocoders, which generally degrade the synthesis quality. We have

also proposed an easy-to-implement locality constraint on the attention mechanism to make the PPG-to-Mel model trainable on utterance-level samples. Note that our MOS ratings are lower than those in the original Tacotron 2 and WaveGlow paper, largely because their systems were trained with 24× more data. One future direction for improving the MOS ratings of the proposed system is training the PPG-to-Mel and WaveGlow models jointly.

In contrast with the baseline, which borrows excitation information (F_0, AP) from the native speaker, our system generates the non-native speaker's excitation directly from the synthesized mel-spectrogram. This prevents the voice quality of the native speaker from "leaking" into the synthesis, making it more similar to the voice quality of the non-native speaker.

Our system extracts native pronunciation patterns from the native PPG sequence, and therefore makes the synthesized speech significantly less accented than the non-native speech. The slight increase in accentedness rating compared to the baseline system could be the result of two factors. First, the AM inevitably produces recognition errors when extracting the PPG and these errors will be reflected as mispronunciations in the synthesis. Second, the proposed model does not explicitly model stress and intonation patterns; as such, we find that some of the synthesis results have unexpected intonations. Therefore, in future work we plan to incorporate intonation information into the modeling process; one possible solution is to condition the PPG sequence on a normalized F_0 contour when training and testing the PPG-to-Mel model.

Currently, the PPG-to-Mel and WaveGlow models need at least one hour of speech from the non-native speaker. This requirement may be relaxed by following the transfer-learning paradigm from multi-speaker TTS [137]. The ultimate goal of accent conversion is to eliminate the need for a reference utterance at synthesis time, i.e., to take a non-native utterance and automatically reduce its accent. This may be accomplished by learning a sequence-to-sequence mapping from the non-native speaker's PPG sequence to a native PPG sequence, and then driving the PPG-to-Mel synthesizer with this accent-reduced PPG sequence.

5. REFERENCE-FREE FOREIGN ACCENT CONVERSION*

5.1. Overview

Foreign accent conversion (FAC) is the problem of generating a synthetic voice that has the voice identity of a second-language (L2) learner and the pronunciation patterns (or accent) of a native (L1) speaker. This synthetic voice has been referred to as a "golden speaker" in the pronunciation-training literature. FAC is generally achieved by building a voice-conversion model that maps utterances from a source (L1) speaker onto the target (L2) speaker. As such, FAC requires that a reference utterance from the L1 speaker be available at synthesis time. This greatly restricts its application scope since the model can only transform utterances that were prerecorded by the L1 speaker. In this work, we propose a "reference-free" FAC system that eliminates the need for reference L1 utterances at synthesis time, and transforms L2 utterances directly. The system is trained in two steps. In a first step, a conventional FAC procedure is used to create a golden speaker using utterances from a reference L1 speaker (which are then discarded) and the L2 speaker. In a second step, a pronunciation-correction model is trained to convert L2 utterances to the golden-speaker utterances obtained in the first step. At synthesis time, the system is presented with a novel L2 utterance, and directly transforms it into its golden-speaker counterpart. Our results show that the system reduces the foreign accent in novel L2 utterances, achieving a 9% (absolute) reduction in word-error-rate of an American English automatic

* This chapter is being submitted to the IEEE/ACM Transactions on Audio, Speech, and Language Processing.

speech recognizer and a 19% (relative) reduction in perceptual ratings of non-native accentedness obtained through listening tests. Over 73% of the listeners also rated golden-speaker utterances and the original L2 utterances as having the same voice identity.

5.2. Introduction

Foreign accent conversion (FAC) [12] aims to create a synthetic voice that has the voice identity (or timbre) of a non-native speaker but the pronunciation patterns (or accent) of a native speaker. In the context of computer-assisted pronunciation training [10, 12, 138, 139], this synthetic voice is often referred to as a "golden speaker" for the non-native speaker –a second-language (L2) learner. The rationale is that the golden speaker is a better target for the L2 learner to imitate than an arbitrary native speaker, because the only difference between the golden speaker and the L2 learner's own speech is the accent, which makes mispronunciations more salient. In addition to pronunciation training, FAC finds applications in movie dubbing [140], personalized Text-To-Speech (TTS) synthesis [127, 141], and improving speech recognition performance [142].

The main challenge in FAC is that one does not have ground-truth data for the desired golden speaker, since, in general, the L2 learner is unable to produce speech with a native accent. Therefore, it is not feasible to apply conventional voice-conversion techniques to the FAC problem. Previous solutions work around this issue by requiring a reference utterance from a native (L1) speaker at synthesis time. But this limits the types of pronunciation practice that FAC techniques can provide, e.g., the L2 learner can only practice sentences that have already been prerecorded by the reference L1 speaker.

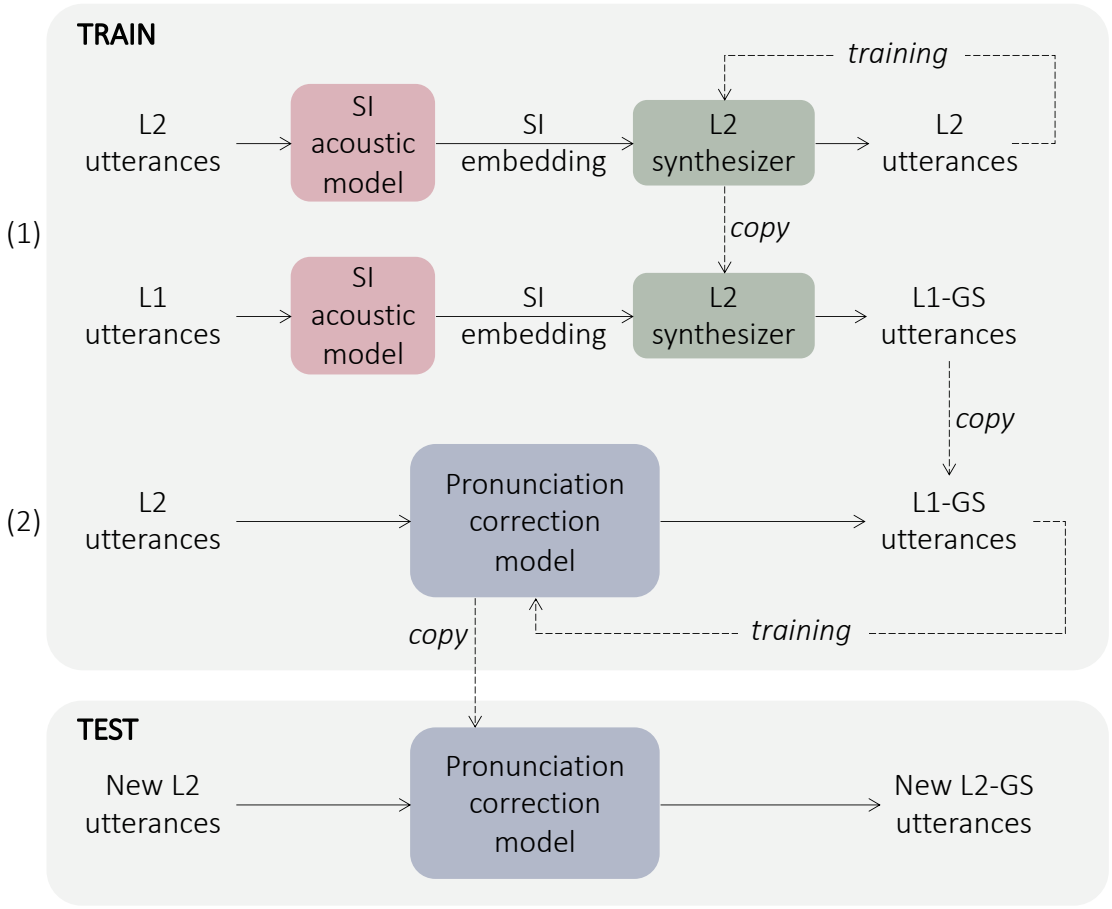

Figure 5.1: Overall workflow of the proposed system. L1: native; L2: non-native; GS: golden speaker; SI: speaker independent. In steps 1, we use a conventional FAC procedure to generate a set of golden-speaker utterances (L1-GS), which serve as targets for step 2. In step 2, we train a pronunciation-correction model that converts L2 utterances into the L1-GS utterances obtained earlier. In the testing stage, a new L2 utterance is processed by the pronunciation-correction model to create its "accent-free" counterpart (L2-GS).

To address this issue, we propose a new FAC system that does not require a reference L1 utterance at inference time. We refer to this type of FAC system as *reference-free*. The proposed system is illustrated in Figure 5.1. Assume that we have a training set of parallel utterances from the L1 and L2 speaker. The training pipeline consists of two

steps. In step one, we build a sequence-to-sequence (seq2seq) speech synthesizer [74] that converts speech embeddings from the L2 utterances into their corresponding mel-spectrograms. The speech embeddings are extracted using an acoustic model trained on a large corpus of native speech, so they are speaker-independent and contain only linguistic information [83, 136]. We then extract speech embeddings from the L1 utterances, and use them to drive the L2 synthesizer. This results in a set of golden-speaker utterances that have the voice identity of the L2 speaker (since they are generated from the L2 synthesizer) and the pronunciation patterns of the L1 speaker (since the input is an L1 utterance). We refer to these golden-speaker utterances as L1-GS utterances, since they are obtained using L1 utterances as a reference. The L1 utterances can be discarded at this point. In the second (and key) step, we train a seq2seq pronunciation-correction model that maps the L2 utterances into the L1-GS utterances obtained in the first step. During inference time, we feed a new L2 utterance to the pronunciation-correction model, which then modifies it to generate its "accent free" counterpart; we refer to the latter as an L2-GS utterance since it is generated directly from the L2 utterance (i.e., in a reference-free fashion).

The pronunciation-correction model is based on a state-of-the-art seq2seq voice conversion framework proposed by Zhang et al. [129], which we use as a baseline. Their system consists of an encoder to extract hidden representations of the input features (e.g., mel-spectra), an attention mechanism to learn the alignment between the input and output sequences, a decoder to predict the output mel-spectrograms, and multi-task phoneme classifiers to help stabilize the training process. During our internal evaluation of the baseline system, we found that it had difficulty converting between an L2 and an L1 speaker

because L2 utterances tend to have a significant amount of disfluency and hesitations, which makes it hard for the attention mechanism to properly align input and output sequences. To address this issue, our system includes a forward-and-backward decoding technique [143, 144] in the pronunciation-correction model to help the attention mechanism and decoder to fully utilize the information in the input data. The rationale is that, by forcing the decoder to compute the attention alignments from both the forward and backward directions during training, we can make the decoder incorporate useful contextual information from both the past and future when producing the alignment. Throughout this study, we use a high-quality WaveGlow [64] real-time neural vocoder to convert mel-spectrograms to speech waveform.

The chapter is organized as follows. Section 5.3 reviews prior approaches on foreign accent conversion as well as related work in seq2seq voice conversion. Section 5.4 describes the proposed reference-free accent conversion system. Sections 5.5 and 5.6 presents the objective and subjective evaluation results and an in-depth discussion of these results. Lastly, we summarize the findings of this work in Section 5.7 and point out future research directions. We include two Appendices that provide additional technical details on model implementation.

5.3. Related work

FAC is related to the more general problem of voice conversion (VC) [88]. In VC, one seeks to transform a source speaker's speech into that of a (known) target speaker. The conversion aims to match the voice characteristics of the target speaker, which include vocal tract configurations, glottal characteristics, pitch range, pronunciation, and speaking

rate; ideally, the only information retained from the source speech is its linguistic content, i.e., the words that were uttered. In contrast with VC, FAC seeks to combine the linguistic content *and* pronunciation characteristics of the source speaker with the voice identity of the target speaker. This is a more challenging problem than VC for two reasons. First, FAC lacks ground-truth since there are no recordings of the L2 speaker producing speech with the desired native target accent. But, more importantly, FAC requires decomposing the speech into voice identity and accent, whereas VC does not. Several techniques have been proposed to perform this decomposition, which can be grouped into articulatory and acoustic methods. The basic strategy in *articulatory methods* is to build an articulatory synthesizer for the L2 speaker, that is, a mapping from the speaker's articulatory trajectories (e.g., tongue and lip movements) to his or her acoustics features (e.g., Mel Cepstra.) Once complete, the L2 speaker's articulatory synthesizer is driven by articulatory trajectories from an L1 speaker to produce "accent-free" speech[16]. A number of techniques can be used to build the articulatory synthesizer, including unit-selection [82], GMMs [17], and DNNs [87].

Decoupling voice identity from accent in the articulatory domain is intuitive, but impractical in most cases since collecting articulatory data is expensive and requires specialized equipment[17]. In contrast, decoupling voice identity from accent in the *acoustic domain* is more practical since it only requires recording speech with a microphone, but is

[16] This process can be likened to "voice puppetry" [145], where the puppet is the articulatory synthesizer and the strings are the native speaker's articulations.
[17] Articulatory measurements can be performed via electromagnetic articulography [82], ultrasound imaging [146], palatography [147], and more recently, real-time MRI [148].

more challenging from a speech-processing standpoint. The conventional approach used in VC (pairing source and target frames via dynamic time warping; DTW) cannot be used in FAC, since it would result in a model that maps native-accented source into non-native-accented target speech. Instead, source and target frames have to be paired based on their linguistic similarity. In early work, Aryal and Gutierrez-Osuna [16] replaced DTW with a technique that matched source (L1) and target (L2) frames based on their MFCC similarity after performing vocal tract length (VTL) normalization. Then, they trained a GMM with those frame pairs to map source L1 utterances to have the target L2 speaker's identity, while retaining the native pronunciations. More recently, Zhao et al. [84] used a speaker-independent acoustic model (i.e., from an ASR system) to estimate the posterior probability that each frame belonged to a set of pre-defined phonetic units –a phonetic posteriorgram (PPG) [24]. Once a PPG had been computed for each source and target frame in the corpus, the two were paired in a many-to-many fashion based on the similarity between their respective PPGs [84, 136]. In their study, matching source and target frames based on their PPG similarity achieved better ratings on accentedness and acoustic quality than matching them based on the VTL-normalized MFCC similarity of Aryal and Gutierrez-Osuna [16].

More recently, Zhao et al. [149] have used sequence-to-sequence (seq2seq) models to perform FAC. In their approach, a seq2seq speech synthesizer is trained to convert PPGs to Mel-spectra using recordings from the L2 speaker. Then, golden-speaker utterances are generated by driving the seq2seq synthesizer with PPGs extracted from an L1 utterance, a process that reminisces articulatory-based methods (i.e., if PPGs are viewed as articulatory

information). Their method produced speech that was significantly less accented than the original L2 speech. Seq2seq models have also garnered much attention in the VC literature since, unlike prior frame-by-frame VC models [21, 77, 94, 95, 128, 150], they can convert segmental and prosody features simultaneously, leading to better conversion performance. Miyoshi et al. [26] built a seq2seq model that mapped source context posterior probabilities to the target's; they obtained better speech individuality ratings (but worse audio quality) than a baseline without the context posterior mapping process. Zhang et al. [25] concatenated bottleneck features and mel-spectrograms from a source speaker, used a seq2seq model to convert the concatenated source features into the target mel-spectrogram, and finally recovered the speech waveform with a WaveNet [62] vocoder. This model outperformed the best-performing system from the 2018 Voice Conversion Challenge [120]. Zhang et al. then applied text supervision [129] on top of [25] to resolve some of the mispronunciations and artifacts in the converted speech. More recently, they extended their framework to the non-parallel condition [151] with trainable linguistic and speaker embeddings. Other notable sequence-to-sequence VC works include [152], which proposed a novel loss term that enforced attention weight diagonalness to stabilize the seq2seq training; the Parrotron [142] system, which uses large-scale corpora and seq2seq models to normalize arbitrary speaker voices to a synthetic TTS voice; and [153], which used a fully convolutional seq2seq model instead of conventional recurrent neural networks (RNNs, e.g., LSTM) because RNNs are costly to train and difficult to optimize for parallel computing.

To the best of our knowledge, the only prior work on *reference-free* FAC is a recent study by Liu et al. [154]. Their system used a speaker encoder, a multi-speaker TTS model, and an ASR encoder. The speaker encoder and the TTS model are trained with L1 speech only, and the ASR encoder is trained on speech data from L1 speakers and the target L2 speaker. During testing, they use the speaker encoder and ASR encoder to extract speaker embeddings and linguistic representations from the input L2 testing utterance, respectively. Then, they concatenate the two and feed them to the multi-speaker TTS model, which then generates the accent-converted utterance. Their evaluations suggested that the converted speech had a near-native accent, but did not capture the voice identity of the target L2 speaker because it had to be interpolated by their multi-speaker TTS. Our proposed method avoids this problem since our pronunciation-correction module is trained on golden-speaker utterances that have been pre-generated for the L2 speaker using a conventional foreign-accent conversion framework.

5.4. Method

Our proposed approach to reference-free FAC is illustrated in Figure 5.1. The system requires a parallel corpus of utterances from the L2 speaker and a reference L1 speaker. The training process consists of two steps. In a first step, we build a speech synthesizer for the L2 speaker that converts speech embeddings into mel-spectrograms, which are then converted to speech waveforms with a WaveGlow neural vocoder[18]. We then drive the L2 synthesizer with a set of utterances from the reference L1 speaker, to produce

[18] We train speaker-dependent WaveGlow neural vocoders for L2 speakers using the official implementation provided by Prenger et al. [64].

a set of golden-speaker utterances (i.e., L2 voice identity with L1 pronunciation patterns). We refer to these as L1 golden-speaker (L1-GS) utterances, since they are obtained using L1 utterances as a reference. The L1 utterances can be discarded at this point. In a second step, we build a pronunciation-correction model that directly maps L2 utterances into their corresponding L1-GS utterances obtained in the previous step, that is, without the need for the L1 reference. Critical in this process is the generation of the speaker embeddings, which we describe first.

5.4.1. Extracting speaker-independent speech embeddings

We use an acoustic model (AM) to generate a speaker-independent (SI) speech embedding for an input (L1 or L2) utterance. Our AM is a Factorized Time Delayed Neural Network (TDNN-F) [155, 156], a feedforward neural network that utilizes time-delayed input in its hidden layers to model long term temporal dependencies. TDNN-F can achieve performance on Large Vocabulary Continuous Speech Recognition (LVCSR) tasks that is comparable to that of AMs based on recurrent structures (e.g., Bi-LSTMs), but is more efficient during training and inference due to its feedforward nature [155]. To produce an SI speech embedding, we concatenate each acoustic feature vector (40-dim MFCC) with an i-vector (100-dim) of the corresponding speaker [46] and use them as

inputs to the AM, which we then train on a large corpus from a few thousand native speakers (Librispeech [104])[19]. As part of this study, we evaluated three different speech embeddings:

- **Senone phonetic posteriorgram (Senone-PPG)**: The output from the final softmax layer of the AM, which is high dimensional (6,024 senones) and contains fine-grained information about the pronunciation pattern in the input utterance.
- **Bottleneck feature (BNF)**: The output of the layer prior to the final softmax layer of the AM. The BNF contains rich classifiable information for a phoneme recognition task, but lower dimensionality (256).
- **Monophone phonetic posteriorgram (Mono-PPG)**: The phonetic posteriorgram for monophones obtained by collapsing the senones into monophone symbols (346 monophones with work positions, e.g., word-initials, work-finals).

5.4.2. Step 1: Generating a reference-based golden-speaker (L1-GS)

In the first step, we build a speech synthesizer for the L2 speaker that converts speech embeddings into mel-spectrograms, which are then converted to speech waveforms with a WaveGlow neural vocoder. See Figure 5.2 (a) for an illustration. We then drive the L2 synthesizer with a set of utterances from the reference L1 speaker, to produce a set of L1 golden-speaker (L1-GS) utterances, as shown in Figure 5.2 (b). We use the resulting L1-GS utterances as training data in the next step (pronunciation-correction). This process

[19] The AM is trained following the Kaldi [157] "tdnn_1d" configuration of the TDNN-F model. We use the full training set (960 hours) in the Librispeech corpus for acoustic modeling. A subset (200 hours) of the training set is used to train the i-vector extractor.

mitigates the lack of ground truth data issue that previously blocked the development of reference-free systems for accent conversion.

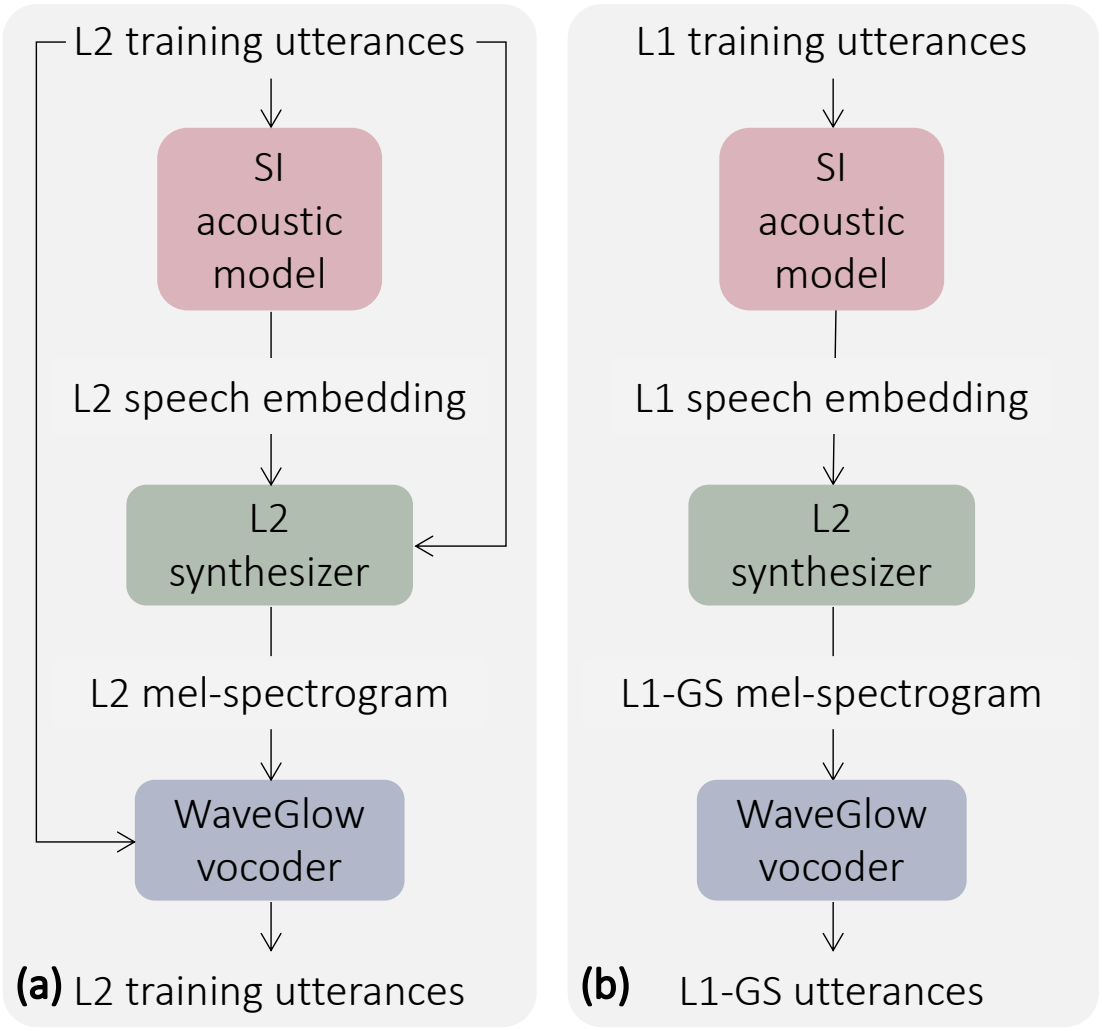

Figure 5.2 (a) Train the L2 speech synthesizer. The speech embedding extracted by the AM is converted to the mel-spectrogram, which is then synthesized to speech waveform through a WaveGlow neural vocoder. (b) Create an L1-GS that corresponds to the L2 speaker by driving the L2 speech synthesizer with training utterances from an L1 reference speaker.

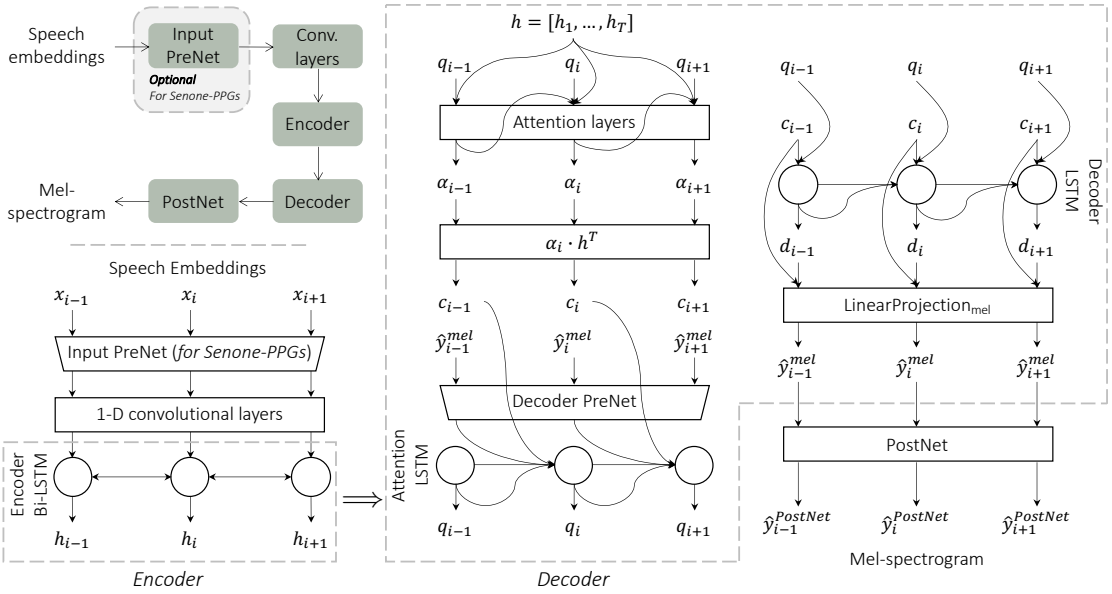

Figure 5.3: Speech embedding to mel-spectrogram synthesizer. The flowchart on the top-left highlights the overall dataflow of the model; the remainder of the figure provides model details. The speech embeddings are sequentially processed by an input PreNet (optional, for Senone-PPGs only), convolutional layers, an encoder, a decoder, and a PostNet to generate their corresponding mel-spectra. We omitted the stop token predictions in the figure for better visualization.

The synthesizer is based on a modified Tacotron2 architecture[20] [74], and is illustrated in Figure 5.3. The model follows a general encoder-decoder (or seq2seq) paradigm with an attention mechanism. Conceptually, an encoder-decoder architecture uses an encoder (usually a recurrent neural network; RNN) to "consume" input sequences and gen-

[20] To facilitate the method description and maintain consistency with prior literature, we adopt the following terminologies from Tacotron2: PreNet: Two fully connected layers with a ReLU nonlinearity; PostNet: Five stacked 1-D convolutional layers; LinearProjection: One fully connected layer. For illustrations of the PreNet and PostNet, please refer to Figure 4.3.

erate a high-level hidden representation sequence. Then, a decoder (an RNN with an attention mechanism) processes the hidden representation sequence. The attention mechanism allows the decoder to decide which parts of the hidden representation sequence contain useful information to make the predictions. At each output time step, the attention mechanism computes an attention context vector (a weighted sum of the hidden representation sequence) to summarize the contextual information. The decoder RNN reads the attention context vectors and predicts the output sequence in an autoregressive manner.

Our speech synthesizer takes the speech embeddings as input. Then, if the input speech embeddings have high dimensionality (e.g., Senone-PPGs), we reduce their dimensions through a learnable input PreNet. This step is essential for the model to converge when using high-dimensional speech embeddings as input. For speech embeddings with lower dimensionality, such as Mono-PPGs and BNFs, we skip the input PreNet. The speech embeddings are then passed through multiple 1-D convolutional layers, which model longer-term context. Next, an encoder (one Bi-LSTM) converts the convolutions into a hidden linguistic representation sequence. Finally, we pass the hidden linguistic representation sequence to the decoder, which consists of a location-sensitive attention mechanism [71] and a decoder LSTM, to predict the raw mel-spectrogram.

Formally, let $[a; b]$ represent the operation of concatenating vectors a and b, $h = [h_1, ..., h_T]$ be the full sequence of hidden linguistic representation from the encoder and $(\cdot)^T$ denote the matrix transpose. At the i-th decoding time step, applying the location-sensitive attention mechanism, the attention context vector c_i is the weighted sum of h,

$$c_i = \alpha_i \cdot h^T. \tag{5.1}$$

$$\alpha_i = \text{AttentionLayers}(q_i, \alpha_{i-1}, h) = [\alpha_i^1, ... \alpha_i^T], \tag{5.2}$$

$$q_i = \text{AttentionLSTM}(q_{i-1}, [c_{i-1};\ \text{DecoderPreNet}(\hat{y}_{i-1}^{mel})]), \tag{5.3}$$

$$\alpha_i^j = \frac{\exp(e_{ij})}{\sum_{j=1} \exp(e_{ij})}, \tag{5.4}$$

$$e_{ij} = v^T \tanh\left(W q_i + V h_j + U f_i^j + b\right), \tag{5.5}$$

$$f_i = F * \alpha_{i-1} = [f_i^1, ..., f_i^T], F \in R^{k \times r}. \tag{5.6}$$

$\alpha_i = [\alpha_i^1, ... \alpha_i^T]$ are the attention weights. q_i is the output of the attention LSTM, and $\hat{y}_{i-1}^{mel}$ is the predicted raw mel-spectrum from the previous time step. v, W, V, U, b, F are learnable parameters of the attention layers. F contains k 1-D learnable kernels with kernel size r, and $f_i^j \in R^k$ is the result of convolving α_{i-1} at position j with F.

Next, let d_i be the output of the decoder LSTM at decoding time step i, and $\hat{y}_i^{mel}$ be the new raw mel-spectrum prediction, we have,

$$d_i = \text{DecoderLSTM}(d_{i-1}, [q_i; c_i]), \tag{5.7}$$

$$\hat{y}_i^{mel} = \text{LinearProjection}_{\text{mel}}([d_i; c_i]). \tag{5.8}$$

At each time step, to determine if the decoder prediction reaches the end of an utterance, we compute a stop token using a separate trainable fully connected layer,

$$\hat{y}_i^{stop} = \begin{cases} 1\ \text{(stop)} & \text{Sigmoid}\left(\text{LinearProjection}_{\text{stop}}([d_i; c_i])\right) \geq 0.5 \\ 0\ \text{(continue)} & \text{Sigmoid}\left(\text{LinearProjection}_{\text{stop}}([d_i; c_i])\right) < 0.5 \end{cases}. \tag{5.9}$$

The original Tacotron 2 was designed to accept character sequences as input, which are significantly shorter than our speech embedding sequences. For example, each sentence in our corpus contains 41 characters on average, whereas the corresponding

speech embedding sequence has a few hundred frames. Therefore, the vanilla location-sensitive attention mechanism might fail, as pointed out in [25]. As a result, the inference would be ill-conditioned and would generate non-intelligible speech. Following a preliminary study [149] of this work, we add locality constraint to the attention mechanism. Speech signals have a strong temporal-continuity and progressive nature. To capture the phonetic context, we only need to look at the speech embeddings in a small local window. Inspired by this, at each decoding step during training, we constrain the attention mechanism to only consider the hidden linguistic representation within a fixed window centered on the current frame, i.e., let,

$$\tilde{h} = [0, \dots, 0, h_{i-w}, \dots, h_i, \dots, h_{i+w}, 0, \dots, 0], \tag{5.10}$$

where w is the window size. Consequentially, we replace eq. (5.2) with eq. (5.11),

$$\alpha_i = \text{AttentionLayers}\big(q_i, \alpha_{i-1}, \tilde{h}\big). \tag{5.11}$$

Finally, to further improve the synthesis quality, the speech synthesizer appends a PostNet after the decoder to predict residual spectral details from the raw mel-spectrum prediction, and then adds the spectral residuals to the raw mel-spectrum,

$$\hat{y}_i^{PostNet} = \hat{y}_i^{mel} + \text{PostNet}(\hat{y}_i^{mel}). \tag{5.12}$$

The loss function for training this speech synthesizer is,

$$L = w_1\big(\|Y_{mel} - \hat{Y}_{mel}^{Decoder}\|_2 + \|Y_{mel} - \hat{Y}_{mel}^{PostNet}\|_2\big) + \ w_2\text{CE}\big(Y_{stop}, \hat{Y}_{stop}\big), \tag{5.13}$$

where Y_{mel} is the ground-truth mel-spectrogram; $\hat{Y}_{mel}^{Decoder}$ and $\hat{Y}_{mel}^{PostNet}$ are the predicted mel-spectrograms from the decoder and PostNet, respectively; Y_{stop} and $\hat{Y}_{stop}$ are the

ground-truth and predicted stop token sequences; $\mathrm{CE}(\cdot)$ is the cross-entropy loss; w_1 and w_2 control the relative importance of each loss term.

The predicted mel-spectrograms are converted back to audio waveforms using a WaveGlow neural vocoder trained on the L2 utterances. We then drive the L2 synthesizer with a set of utterances from the reference L1 speaker, to produce the L1-GS utterances.

5.4.3. Step 2: Generating the reference-free golden speaker (L2-GS) via pronunciation-correction

In the second step, we train a pronunciation-correction model that converts L2 utterances to match the pronunciations (accents) of the L1-GS utterances generated in step 1. In the testing stage, a new L2 utterance is processed by the pronunciation-correction model to create its "accent-free" counterpart, to which we refer as an L2-GS utterance, since it is driven by an L2 utterance at the input. As in the previous step, we use speaker-dependent WaveGlow neural vocoder to generate audio waveform from the mel-spectrogram.

Our pronunciation-correction model is based on a state-of-the-art seq2seq VC system proposed by Zhang et al. [129]. We chose this system as a baseline since it outperformed the best system in the Voice Conversion Challenge 2018 [120]. The rationale behind using a VC system as the pronunciation-correction model is that VC can convert both the voice identity and accent to match the target speaker. In our application scenario, we treat the L2 speaker and the L1-GS as the source and target speakers in a VC task, respectively. Since the two speakers already share the same voice identity, the VC model only

needs to match the accent of the target speaker (i.e., the golden speaker). During the inference stage, we can directly input L2 speech into the pronunciation-correction model, and the output will share similar pronunciation patterns as the GS. The difficulty of this procedure is that L2 speakers tend to have disfluencies, hesitations, and inconsistent pronunciations, making the conversion much harder than converting between two native speakers, as discussed in prior literature [136]. To overcome this difficulty, we propose to use a variation of the forward-and-backward decoding technique [143, 144], in addition to the baseline pronunciation model, to achieve better pronunciation-correction performance. We first formally introduce the baseline system, and then describe the proposed improvement.

The baseline system is also based on an encoder-decoder paradigm with an attention mechanism. Figure 5.4 shows an overview of the baseline system. Unlike conventional parallel VC systems (e.g., GMM, feedforward neural networks), which need time-alignment between the source and target speakers to generate the training frame pairs, seq2seq systems use an attention mechanism to produce learnable alignments between the input and output sequences. Therefore, they can also adjust for prosodic differences (e.g., pitch, duration, and stressing) between the input and output sequences. In our application, this is crucial since prosody errors also contribute to foreign accentedness.

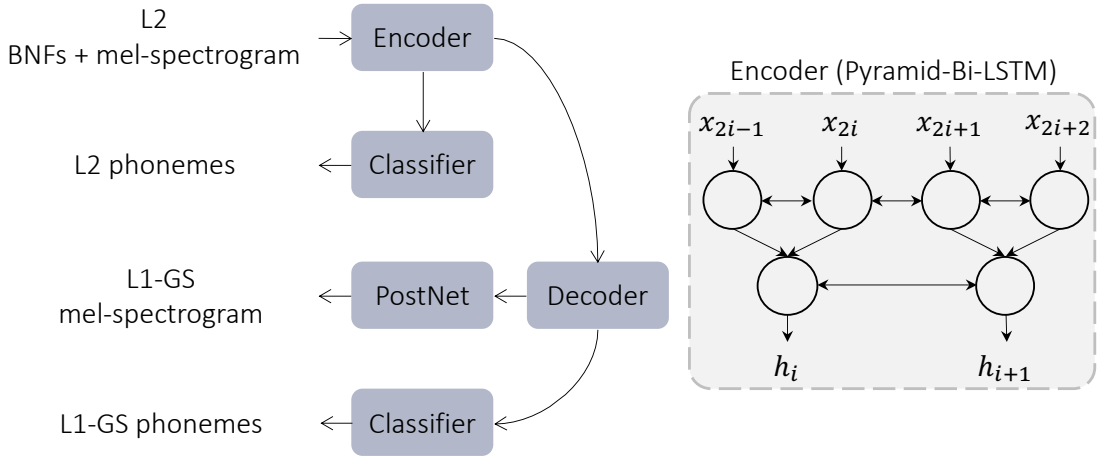

Figure 5.4: Training pipeline of the baseline pronunciation-correction model. The input feature sequence (concatenation of bottleneck features [BNFs] and mel-spectra) from the L2 speaker is converted to the L1-GS mel-spectrogram. The phoneme classifications are only applied to stabilize the model training and are discarded during testing. The encoder is constructed with a two-layer Pyramid-Bi-LSTM. The decoder has the same neural network structure as the one in Figure 5.3.

Specifically, the input $X = [x_1, \dots, x_{T_{in}}]$ to the conversion system is the concatenation of the bottleneck features[21] (i.e., BNFs, cf. Section 5.4.1) and mel-spectrogram computed from the L2 utterance. Let x_i be the i-th feature vector in the sequence. The output sequence is denoted by $Y_{mel} = [y_1^{mel}, \dots, y_{T_{out}}^{mel}]$ where y_i^{mel} is the i-th mel-spectrum of the L1-GS utterance. A two-layer Pyramid-Bi-LSTM encoder [72] with a down-sampling rate of two consumes the input sequence and produces the encoder hidden embeddings $h = \left[h_1, \dots, h_{\lfloor \frac{i}{2} \rfloor}, \dots, h_{\lfloor \frac{T_{in}}{2} \rfloor}\right]$, where $h_{\lfloor \frac{i}{2} \rfloor}$ is one encoder hidden embedding vector, and $\lfloor \cdot \rfloor$ is the floor-rounding operator.

[21] Zhang et al. [129] use BNFs in their implementation, and we follow this design choice to replicate their system.

The first Bi-LSTM layer does the recurrent computations on X and outputs $h_{layer1} = \left[h^1_{layer1}, \dots, h^{T_{in}}_{layer1}\right]$. We then concatenate each two of the consecutive frames in h_{layer1} to form $\left[\left[h^1_{layer1}; h^2_{layer1}\right], \dots, \left[h^i_{layer1}; h^{i+1}_{layer1}\right], \dots, \left[h^{T_{in}-1}_{layer1}; h^{T_{in}}_{layer1}\right]\right]$. Finally, we feed the concatenated vectors to the second Bi-LSTM layer to produce h. In the case that we have an odd number of frames in the input sequence, we drop the last frame, which is generally a silent frame. The down-sampling effectively reduces the sequence length of the input, which speeds up the encoder computation by a factor of two and makes it easier for the attention mechanism to learn a meaningful alignment between the input and output sequences.

The decoder in this model has a similar neural-network structure as the speech synthesizer decoder in Section 5.4.2 (Figure 5.3), with only two differences: (1) to replicate Zhang et al. [129], we use the forward-attention technique [158] instead of eq. (5.4) to normalize the attention weights; (2) the locality constraint defined in equations (5.10) and (5.11) is discarded. The decoder predicts the output raw mel-spectrogram sequence $\hat{Y}^{Decoder}_{mel} = [\hat{y}^{mel}_1, \dots, \hat{y}^{mel}_{T_{out}}]$ and the stop token sequence $\hat{Y}_{stop} = [\hat{y}^{stop}_1, \dots, \hat{y}^{stop}_{T_{out}}]$ following equations (5.8) and (5.9), respectively. $\hat{Y}^{Decoder}_{mel}$ is also processed through a Post-Net to generate a residual-compensated mel spectrogram $\hat{Y}^{PostNet}_{mel}$, following eq. (5.12).

In addition, the baseline system uses multi-task learning to make the synthesized pronunciations more stable. Two independent phoneme classifiers, each containing one fully-connected layer and a softmax operation, are added to predict the input and output

phoneme sequences $\hat{Y}_{InP} = [\hat{y}_1^{inP}, \ldots, \hat{y}_{T_{in}}^{inP}]$ and $\hat{Y}_{OutP} = [\hat{y}_1^{outP}, \ldots, \hat{y}_{T_{out}}^{outP}]$, respectively. These phoneme classifiers are only used during training and are discarded in inference.

$$\hat{y}_i^{inP} = \text{PhonemeClassifier}_{\text{in}}(h_i) \tag{5.14}$$

$$\hat{y}_i^{outP} = \text{PhonemeClassifier}_{\text{out}}([q_i; c_i]) \tag{5.15}$$

The final loss function of the baseline system becomes,

$$\begin{aligned} L_{baseline} = w_1\big(\|Y_{mel} - \hat{Y}_{mel}^{Decoder}\|_2 + \|Y_{mel} - \hat{Y}_{mel}^{PostNet}\|_2\big) \\ + w_2 \text{CE}\big(Y_{stop}, \hat{Y}_{stop}\big) \\ + w_3 \Big(\text{CE}\big(Y_{inP}, \hat{Y}_{inP}\big) + \text{CE}\big(Y_{outP}, \hat{Y}_{outP}\big)\Big), \end{aligned} \tag{5.16}$$

where Y_{inP}, Y_{outP} are the ground input and output phoneme sequence, respectively.

To improve predictive performance, we propose a modification to the baseline system that applies forward-and-backward decoding during the training process. The forward-and-backward decoding technique maintains two separate decoders, i.e., the forward and backward decoders. The forward decoder processes the encoder outputs in the forward direction, whereas the backward decoder reads the encoder outputs reversely. Different variations of this technique have been applied to TTS [144] and ASR [143]. Figure 5.5 shows an overview of this procedure. During training, we add a backward decoder to the baseline model. The backward decoder has the same structure as the existing decoder (denoted as the forward decoder) but with a different set of weights. The backward decoder functions the same as the forward decoder except that it processes the encoder's output in *reverse order* and predicts the output mel-spectrogram $\hat{Y}_{mel}^{bwd}$ *reversely* as well. The backward decoder, like its forward counterpart, also predicts its own set of stop tokens $\hat{Y}_{stop}^{bwd}$,

output phoneme labels $\hat{Y}^{bwd}_{outP}$, and uses the shared PostNet to predict a refined mel-spectrogram $\hat{Y}^{bwd}_{mel-PostNet}$.

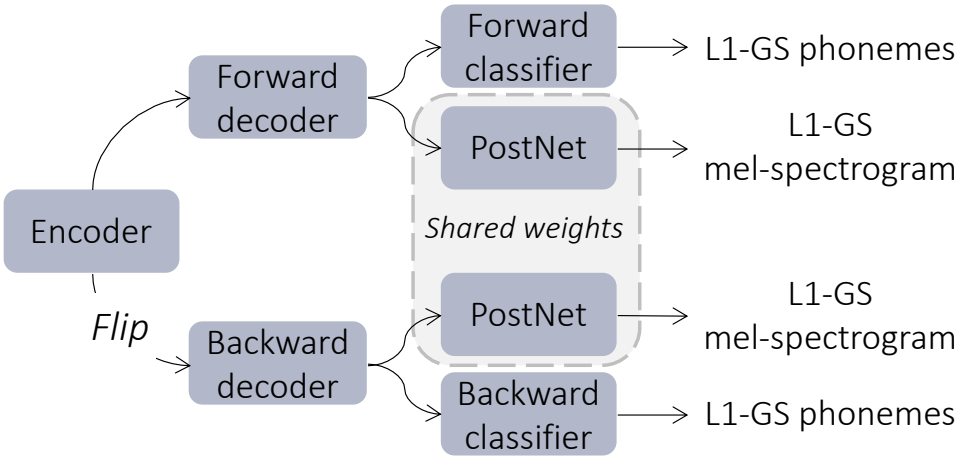

Figure 5.5: Proposed forward-and-backward decoding model for pronunciation-correction. The existing decoder in the baseline model is denoted as the forward decoder here. We omitted the other common components it shares with the baseline model. The PostNet of the two decoders shares the same set of weights. This forward-and-backward decoding procedure is only activated during training.

The loss terms contributed by adding this backward decoder are,

$$L_{bwd} = w_1\big(\|Y_{mel} - \hat{Y}^{bwd}_{mel}\|_2 + \|Y_{mel} - \hat{Y}^{bwd}_{mel-PostNet}\|_2\big) + \; w_2\mathrm{CE}\big(Y_{stop}, \hat{Y}^{bwd}_{stop}\big) + w_3\mathrm{CE}\big(Y_{outP}, \hat{Y}^{bwd}_{outP}\big). \tag{5.17}$$

Additionally, to force the two decoders to learn complementary information from each other, we train the two decoders to produce the same attention weights by including the following loss term,

$$L_{att} = w_4\|\alpha_{fwd} - \alpha_{bwd}\|_2, \tag{5.18}$$

where α_{fwd} and α_{bwd} are the attention weights of the forward and backward decoder, respectively.

The final loss term of the proposed system is,

$$L_{proposed} = L_{baseline} + L_{bwd} + L_{att}. \tag{5.19}$$

The rationale behind the forward-and-backward decoding is that RNNs are generally more accurate at the initial decoding time steps, but performance decreases as the predicted sequence becomes longer because the prediction errors accumulate due to the autoregression. By including two decoders that model the input data in two different directions, and by constraining them to produce similar attention weights, we force the two decoders to incorporate information from both the past and future, thus improving their modeling power. Note that we only use both decoders during training. During inference time, we keep either the forward or backward decoder and discard the other. Therefore, the model size is exactly the same as the baseline model.

5.5. Results

We conducted two experiments to evaluate the proposed FAC system on a thorough set of objective measures (e.g., word error rates, Mel Cepstral distortion) and subjective measures (degree of foreign accent, audio quality, and voice similarity.) In experiment 1, we evaluated the reference-based golden speaker (L1-GS) generated by the speech synthesizer (Section 5.4.2). Then, in experiment 2, we evaluate the reference-free golden speaker (L2-GS) produced by the pronunciation-correction system (Section 5.4.3). We start with introducing the speech corpora and common experimental settings first.

5.5.1. Data and common settings

For the FAC task (training the speech synthesizers, WaveGlow neural vocoders, and pronunciation-correction models), we used one native speaker (BDL; American accent) from CMU-ARCTIC corpus [105] and two non-native speakers (YKWK, Korean; TXHC, Chinese) from the L2-ARCTIC corpus[22] [106]. We split the data from all speakers into non-overlapping training (1032 utterances), validation (50 utterances), and testing (50 utterances) sets. Recordings from BDL were sampled at 16 kHz. Recordings in the L2-ARCTIC corpus were resampled from 44.1 kHz to 16 kHz to match BDL's sampling rate and were pre-processed with Audacity [132] to remove any ambient background noise. In all FAC tasks, we extracted 80-dim mel-spectrogram with a 10ms shift and 64ms window size. All neural network models were implemented in PyTorch [159] and trained with an NVIDIA Tesla P100 GPU.

5.5.2. Experiment 1: Evaluating the reference-based golden speaker (L1-GS)

We constructed the following three systems and compared their performance in generating L1-GS utterances. The objectives of this experiment were to determine the optimal speech embedding, and more importantly, to establish that L1-GS utterances captured the native accent and the L2 speaker identity, which is critical since they would be used as targets for the reference-free FAC task. Details of the model configurations and training are summarized in Appendix E.

- **Senone-PPG**: use the senone-PPG as the input (6,024 dimensions).

[22] https://psi.engr.tamu.edu/l2-arctic-corpus

- **Mono-PPG**: use the monophone PPG as the input (346 dimensions).
- **BNF**: use the bottleneck feature as the input (256 dimensions).

To generate the L1-GS utterances for testing, we extracted the three speech embeddings from speaker BDL's test set and drove the systems with their respective input. The output mel-spectrograms were then converted to speech through the WaveGlow vocoders.

5.5.2.1. Objective evaluation

In a first experiment, we computed the word error rate (WER) of L1-GS utterances synthesized using each of the three speaker embeddings. In our case, the speech recognizer consisted of the TDNN-F acoustic model combined with an unpruned 3-gram language model trained on the Librispeech transcripts. As a reference, we also computed WERs on test utterances from the L1 speaker (BDL) and the two L2 speakers (YKWK, TXHC). Results are summarized in Table 5.1. L1-GS utterances from the three systems achieve lower WERs than the corresponding utterances from the L2 speakers. Since the acoustic model had been trained on American English speech, a reduction in lower WERs can be interpreted as a reduction in the foreign-accentedness. The BNF system performs markedly better than the other two systems, achieving WERs that are close to those on L1 utterances. The Senone-PPG system performed the worst, despite the fact that it contains the most fine-grained triphone-level phonetic information. We offer an explanation in the discussion.

Table 5.1: Word error rates (%) on test utterances and the original speech.

	Senone-PPG	***Mono-PPG***	***BNF***	***Original speech***
YKWK	37.56	23.30	**9.50**	45.82
TXHC	28.05	23.53	**7.47**	44.57
Average	32.81	23.42	**8.49**	45.20
BDL		N/A		4.98

5.5.2.2. Subjective evaluation

To further evaluate the three L1-GS systems, we conducted formal listening tests to rate three perceptual attributes of the synthesized speech: accentedness, acoustic quality, and voice similarity. All listening tests were conducted through the Amazon Mechanical Turk platform[23]. Instructions were given in each test to help the participants focus on the target speech attribute. All tests included five calibration samples to detect cheating behaviors, as suggested by Buchholz and Latorre [113]; responses from participants who were deemed to have cheated were excluded. Ratings for the calibration samples were excluded, too. All participants received monetary compensation. All samples were randomly selected from the test set, and the presentation order of samples in every listening test was randomized and counter-balanced. All participants resided in the United States at the time of the recruitment and passed a qualification test where they identified several regional dialects in the United States. All participants were self-reported native English speakers.

[23] https://www.mturk.com

Accentedness test. Listeners were asked to rate the foreign accentedness of an utterance on a nine-point Likert-scale (1: no foreign accent; 9: heavily accented), which is widely used in the pronunciation training community [4]. Listeners were told that the native accent in this task was General American. Participants (N=20) rated 20 randomly selected utterances per system per L2 speaker. The utterances shared the same linguistic content in all conditions to ensure a fair comparison. As a reference, listeners also rated the same set of sentences for the L1 and L2 speakers. The results are summarized in the first row of Table 5.2. L1-GS utterances from the three systems were rated significantly ($p \ll 0.001$) more native-like than the original L2 speech, though not as much as the original L1 speech. Among the three systems, the BNF system significantly outperformed Mono-PPG, while Mono-PPG was rated significantly more native-like than Senone-PPG, all with $p \ll 0.001$.

Table 5.2: Accentedness (the lower, the better) and MOS ratings (the higher, the better) of the golden, native, and non-native speakers; the error ranges show the 95% confidence intervals; the same convention applies to the rest of the results.

	Senone-PPG	***Mono-PPG***	***BNF***	***Original L2***	***Original L1***
Accentedness	6.01 ± 0.26	5.48 ± 0.19	**4.30 ± 0.16**	6.77 ± 0.20	1.04 ± 0.04
MOS	3.43 ± 0.13	3.54 ± 0.09	**3.78 ± 0.05**	3.70 ± 0.06	4.63 ± 0.06

Acoustic quality. Listeners were asked to rate the acoustic quality of an utterance using a standard five-point (1: poor; 2: bad; 3: fair; 4: good; 5: excellent) Mean Opinion Score (MOS). Participants (N=20) listened to 20 randomly-selected sentences per L2

speaker per system. As in the accentedness test, listeners also rated the original utterances from the L1 and L2 speakers. The results are summarized in the second row of Table 5.2. As expected, the original native speech received the highest MOS. Among the three golden speaker voices, BNF achieved the highest MOS compared with the other two systems ($p \ll 0.001$). The Mono-PPG system obtained better acoustic quality than the Senone-PPG system ($p = 0.045$). Interestingly, L1-GS utterances from the BNF system received higher MOS than the original L2 speech (3.78 vs. 3.70, $p = 0.02$); we offer a possible explanation in Section 5.6.

Voice similarity test. Listeners were presented with a pair of speech samples, an L1-GS synthesis, and the original utterance from the corresponding L2 speaker. In the test, listeners first had to decide if the two samples were from the same speaker, and then rate their confidence level on a seven-point scale (1: not confident at all; 3: somewhat confident; 5: quite a bit confident; 7: extremely confident). To minimize the influence of accent, the two utterances had different linguistic contents and were played in reverse, following [12]. For each system, participants (N=20) rated 10 utterance pairs per speaker (20 utterance pairs for each system). Results are summarized in Table 5.3. Across the three systems, more than 70% of the listeners were "quite a bit" confident (4.82-4.93 out of 7) that the L1-GS utterance and the original L2 utterance had the same voice identity. Significance tests showed that there was no statistically significant difference between the preference percentages for the three systems.

Table 5.3: Voice similarity ratings. The first row shows the percentage of the raters that believed the synthesis and the reference audio clip were produced by the same speaker; the second row is the average rating of these raters' confidence level when they made the choice.

	Senone-PPG	***Mono-PPG***	***BNF***
Prefer "same speaker"	70.00 ± 9.12%	71.25 ± 6.38%	**73.75** ± **6.46**%
Average rater confidence	4.82	4.89	**4.93**

These results show that the BNF system outperforms the other two systems significantly in both objective and subjective measures. Therefore, for the remainder of this manuscript, we focus our evaluation on the BNF system, i.e., target L1-GS utterances for the reference-free (pronunciation-correction) FAC system are those from the BNF system.

5.5.3. Experiment 2: Evaluating the reference-free golden speaker (L2-GS)

In the second experiment, we directly converted L2 test utterances with the proposed pronunciation-correction model and compared it against the baseline system. Detailed model architecture configurations and training setups are included in Appendix F.

- **Baseline**: the system of Zhang et al. [129], a state-of-the-art VC system capable of modifying segmental and prosodic attributes between different speakers.
- **Proposed**: the baseline system with the forward-and-backward decoding, which added a backward decoder that has the same structure as the forward decoder during training. We performed the proposed accent conversion using the backward decoder during testing since it produced significantly better-quality speech compared to the forward decoder on the validation set.

5.5.3.1. Objective evaluations

For objective evaluations, we computed three measures, as suggested by [129], plus WER as a fourth:

- **MCD**: the Mel-Cepstral Distortion [21] between the L2-GS (actual output) and L1-GS speech (desired output). It was computed on time-aligned (Dynamic Time Warping) mel-cepstra between the L2-GS and the L1-GS audio. Lower MCD correlates with better spectral predictions. We used SPTK [160] and the WORLD vocoder [57] to extract the Mel-cepstra with a shift size of 10ms.
- **F_0 RMSE**: the F_0 RMSE between the L2-GS and L1-GS speech on voiced frames. Lower F_0 RMSE represents better pitch conversion performance. The F_0 and voicing features were extracted by the WORLD vocoder with the Harvest pitch tracker [161].
- **DDUR**: the absolute difference in duration between the L2-GS and L1-GS speech. Lower DDUR implies better duration conversion performance.
- **WER**: the word error rate for the L2-GS speech. Ideally, the L2-GS speech should have a lower WER than the original non-native speech, implying that the conversion reduced the foreign accent.

Results are summarized in Table 5.4. For all measures, we also computed the scores between the original L2 speech and the L1-GS speech as a reference. The proposed method obtained better WER, MCD, and DDUR scores, while the baseline method performed slightly better on the F_0 RMSE. More importantly, both systems were able to reduce the WER of the input L2 utterance. The proposed method reduced the WER of the

non-native speech by 9.26% (absolute) on average, which was significantly higher than the WER reduction of the baseline system (2.71% absolute).

Table 5.4: Objective evaluation results of the reference-free FAC system, i.e., the pronunciation correction. The first row in each block shows the scores between the original L2 utterances and the L1-GS utterances. The last block shows the average values of the first two blocks. For all measurements, a lower value suggests better performance.

L2 speaker	*System*	*WER (%)*	*MCD (dB)*	*F_0 RMSE (Hz)*	*DDUR (sec)*
YKWK	Original	45.82	8.07	23.38	1.15
	Baseline	41.31	6.26	**18.43**	0.18
	Proposed	**34.54**	**6.10**	20.78	**0.15**
TXHC	Original	44.57	8.00	25.73	1.29
	Baseline	43.67	6.32	**19.40**	0.17
	Proposed	**37.33**	**6.29**	21.37	**0.15**
Average	Original	45.20	8.04	24.56	1.22
	Baseline	42.49	6.29	**18.92**	0.18
	Proposed	**35.94**	**6.20**	21.08	**0.15**

5.5.3.2. Subjective evaluation

Following the same protocol described in section 5.5.2.2, we asked participants to rate the accentedness, acoustic quality, and voice similarity of synthesized L2-GS utterances.

Accentedness test. Participants (N=20) rated 20 random samples per speaker per system, as well as the corresponding original audio. Results are compiled in the first row of Table 5.5. Both systems obtained significantly more native-like ratings than the original

L2 utterances ($p \ll 0.001$). More specifically, the baseline system reduced the accentedness rating by 15.5% (relative), while the proposed system achieved a 19.0% relative reduction. Further, the proposed system had a statistically-significant lower rating of foreign accentedness than the baseline ($p = 0.04$). As expected, the original L1 speech was rated less accented than all other systems.

Table 5.5: Accentedness (the lower, the better) and MOS (the higher, the better) ratings of the reference-free accent conversion systems and original L1 and L2 utterances.

	Baseline	***Proposed***	***Original L2***	***Original L1***
Accentedness	5.56 ± 0.23	**5.33 ± 0.28**	6.58 ± 0.26	1.07 ± 0.04
MOS	2.95 ± 0.12	**3.22 ± 0.10**	3.68 ± 0.10	4.80 ± 0.06

MOS test. Participants (N=20) rated 20 audio samples per speaker per system. We used the same MOS test as in experiment 1 to measure the acoustic quality of the synthesis. Results are shown in the second row of Table 5.5. The proposed system achieved significantly better audio quality than the baseline model (9.15% relative improvement; $p \ll 0.001$).

Voice similarity test. Participants (N=20) rated 10 utterance pairs per speaker per system (i.e., 20 utterance pairs for each system). This last experiment verified that the accent conversion retained the voice identity of the L2 speakers. The results are shown in Table 5.6. The majority of the participants thought the synthesis and the reference speech were from the same speaker, and they were "quite a bit confident" (5.00-5.12 out of 7)

about their ratings. Although the proposed system obtained higher ratings than the baseline system in terms of voice identity, the difference between the preference percentages was not statistically significant ($p = 0.12$), which was expected. The reason is that the input and output speech had different accents, but very similar voice identity. Therefore, both systems were not trained to modify the voice identity of the input audio. As a result, both the baseline system and the proposed system were able to keep the voice identity unaltered during the conversion process.

Table 5.6: Voice similarity ratings of the reference-free accent conversion task.

	Baseline	***Proposed***
Prefer "same speaker"	69.25 ± 11.08%	**73.00 ± 7.55%**
Average confidence rating	5.00	**5.12**

Aside from the objective and subjective scores, we provide an example of the attention weights produced by both systems on a test utterance in Figure 5.6. Qualitatively, we can observe that the attention weights of the baseline system contained an abnormal jump towards the end of the synthesis, while the proposed system produced smooth alignments at the same time steps. Additionally, the proposed method appears to have used a broader window to compute the attention context compared with the baseline, as reflected by the width of the attention alignment path. Therefore, the proposed system utilized more contextual information during the decoding process.

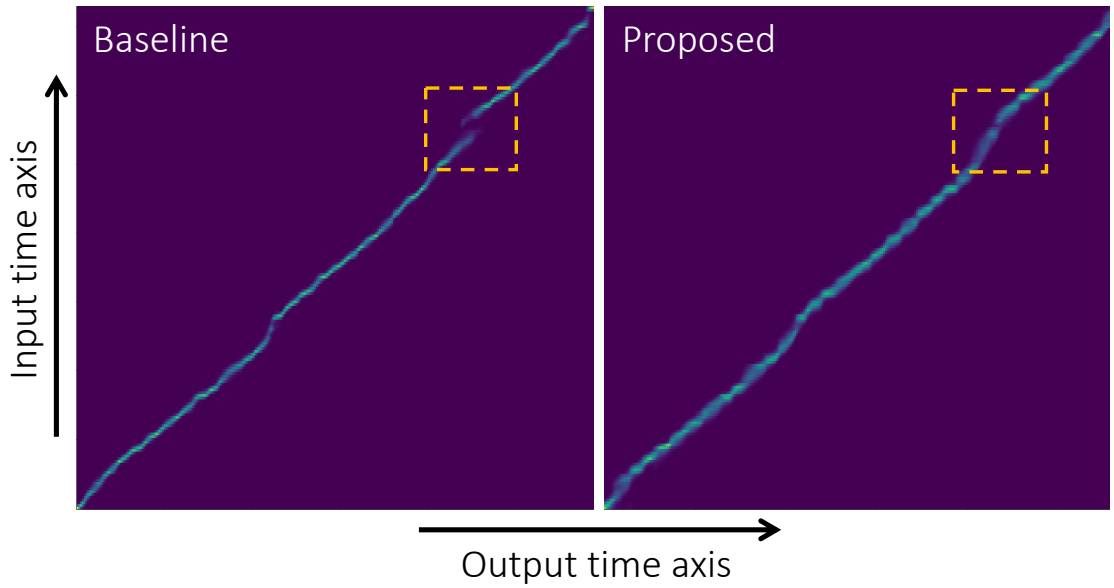

Figure 5.6: A qualitative comparison of the attention weights generated by the baseline and the proposed pronunciation-correction systems on one testing utterance.

5.6. Discussion

We have presented a system that can transform utterances from an L2 speaker to make them sound more native-like. Training the system requires two steps. In a first step, we train an accent-conversion system to transform utterances from a reference L1 speaker, so they have the voice identity of the L2 speaker. We refer to these transformed utterances as L1-GS utterances. In a second step, we train a pronunciation-correction system that can transform utterances from an L2 speaker to match the L1-GS utterances obtained in the first step. We refer to these transformed utterances as L2-GS utterances. We conducted two series of experiments to evaluate both steps of the process. In experiment 1, we tested three versions of the L1-GS system that used different speech embeddings at the input:

senone-PPG, monophone-PPG, and bottleneck feature (BNF). Both objective and subjective tests suggested that the BNF system was better than the senone- and monophone-PPG systems, both in terms of audio quality and accentedness. Since one of the major objectives of FAC is to capture the native accent as much as possible, the better accentedness rating of the BNF system suggests that it is more advantageous for the FAC task. It is also worth noting that the WER evaluation yielded similar WERs on BNF L1-GS utterances and on the original utterances from the L1 speaker, which further indicates that the accent reduction was successful. The majority of the human raters (73.75%) had high confidence that the BNF L1-GS shared the same voice identity as the target L2 speaker, which many prior FAC systems struggled to achieve. It was surprising to observe that the BNF L1-GS utterances were rated to have better audio quality than the original natural speech from the L2 speaker. Although this result indicates that the BNF L1-GS speech had high acoustic-quality, prior literature has established that native listeners can have negative bias [6, 7, 9] towards accented speech. Therefore, it is reasonable to argue that the native listeners rated the L2 speech not just based on the acoustic quality, but also based on implicit biases towards foreign accents. The discussion of native listeners' bias towards foreign accents is beyond the scope of this work, but we acknowledge that this negative bias does exist, and the majority of our listeners are monolingual. Native listeners might be able to reduce their bias with more exposure to foreign accents. At the same time, carefully designed listening test protocol (e.g., playing the utterances reversely as in the voice similarity tests) would also help control these factors when we use native listeners as test subjects.

Two probable factors explain why BNF outperformed the other two speech embeddings. First, we observed that during the training process, the BNF system converged to a better terminal validation loss, which suggested that the speech synthesizer could model the mel-spectrograms more accurately given BNF as the input, compared with the other speech embeddings. As a result, the BNF speech synthesizer produced speech syntheses with better quality. Second, although the BNF and PPGs contain a comparable amount of linguistic information, the process that converted the BNF to PPGs was a phoneme classification task. Therefore, it introduced recognition errors, which propagated to the speech synthesizer as mispronunciations and speech artifacts. One possible explanation for differences between the two PPGs is dimensionality reduction strategies; the monophone-PPG system used an empirical rule (reducing senones to monophones) to summarize the high-dimensional senone-PPG, while the senone-PPG system constructed a learnable transformation. Although it is possible for data-driven transforms to outperform empirical rules given enough data, the limited amount of data (~one hour of speech per speaker) available for the FAC task was not enough to produce a good transformation for senone-PPG.

In experiment 2, we achieved reference-free FAC by constructing a pronunciation-correction model that mapped speech directly from the L2 speaker to L1-GS. The results were promising; both the baseline model of Zhang et al. [129] and the proposed system were able to reduce the foreign accentedness of the input speech significantly, while retaining the voice identity of the L2 speaker. It was interesting to see that the baseline

system had difficulty converting an L2 speaker to the BNF L1-GS[24]. The major difficulty arises from the fact that L2 speakers generally have a large number of disfluencies (e.g., hesitations, pauses) in their speech, and their pronunciations are not always consistent due to their unfamiliarity with the second language. Although the seq2seq conversion model does not require explicit time-alignment between the source and target speakers, the attention mechanism and the decoder implicitly learn the alignment. The disfluencies and inconsistent pronunciations made it difficult for the attention and decoder to produce the proper alignment. The proposed method, on the other hand, computed the alignment between each pair of input and output sequences from two directions at training time, once in the forward and once in the backward direction. Both directions provided useful and complementary information about what a better alignment should be. By forcing the forward and the backward decoders to produce similar alignment weights at training time, we make the decoders incorporate information from both the past and future when generating the alignment. During inference time, only one decoder is needed to perform the reference-free accent conversion; therefore, the proposed system consumes exactly the same amount of inference recourses as the baseline system. The better accentedness and audio quality ratings can largely be attributed to the better alignments provided by the forward-and-backward decoding training technique, as illustrated in Figure 5.6.

[24] Since Zhang et al. [129] did not open-source their system, we replicated it following the recipe prescribed in their manuscript. When converting between two *native* American English speakers (speakers RMS and SLT from the CMU ARCTIC corpus), our pilot study (not shown here) verified that our implementation could produce speech with high audio quality and proper pronunciations.

The L2-GS generated by the reference-free FAC was rated as significantly less accented than the L2 speaker, though it still had a noticeable foreign accent. One possible explanation is that the pronunciation-correction model was not able to fully eliminate the foreign accent in heavily mispronounced or disfluent speech segments, and therefore some foreign accent cues from the input were carried over to the output speech. Furthermore, the current reference-free FAC model can only correct *phone substitution* errors. Neither removing *phone insertion* errors nor filling in *phone deletion* errors is possible without knowing the *canonical* phonetic transcription. The MOS ratings of the pronunciation-correction models were lower than the BNF L1-GS, which was expected since the output speech of the pronunciation-correction model was the re-synthesis of the L1-GS.

5.7. Conclusion

In this work, we propose a new reference-free FAC system that directly reduces the foreign accent in the input L2 utterances, in contrast to the majority of the existing methods, which require native reference utterances at inference time. The proposed system first constructs a parallel golden speaker corpus from L2 training utterances. Our experiments showed that bottleneck features produced the optimal golden speaker with the best audio quality and native accent. Then, we construct a pronunciation-correction model that adjusts the L2 speech to sound like the golden speaker. Our evaluations indicate that the reference-free FAC system can significantly reduce the foreign accentedness in L2 speech while retaining the voice identity. One possible future direction of this work is to use transfer learning [137] to reduce the amount of training data needed for the golden speaker

generation process. Another interesting research direction is to train a pronunciation-correction model that takes the canonical (correct) phonetic transcriptions as a supplementary input signal, in addition to the acoustic sequence from the L2 speaker. This may allow the system to correct not only the phone substitution errors, but also the insertion and deletion errors, thus improving the accentedness ratings. The source code and audio samples from this work can be found at https://guanlongzhao.github.io/demo/reference-free-ac.

6. L2-ARCTIC: A NON-NATIVE ENGLISH SPEECH CORPUS*

6.1. Overview

In this chapter, we introduce L2-ARCTIC, a speech corpus of non-native English that is intended for research in voice conversion, accent conversion, and mispronunciation detection. The current version (v5.0) includes recordings from 24 non-native speakers of English whose first languages (L1s) are Hindi, Korean, Mandarin, Spanish, Vietnamese, and Arabic, each L1 containing recordings from two male and two female speakers. Each speaker recorded approximately one hour of read speech from the Carnegie Mellon University ARCTIC prompts, from which we generated orthographic and forced-aligned phonetic transcriptions. In addition, we manually annotated 150 utterances per speaker to identify three types of mispronunciation errors: substitutions, deletions, and additions, making it a valuable resource not only for research in voice conversion and accent conversion but also in computer-assisted pronunciation training. The corpus is publicly accessible at https://psi.engr.tamu.edu/l2-arctic-corpus/.

6.2. Introduction

Voice conversion (VC) [88] aims to transform utterances from a source speaker to make them sound as if a target speaker had uttered them. The closely related problem of accent conversion (AC) [16] goes a step further, mixing the source speech's linguistic

---

* © 2018 ISCA. Reprinted, with permission, from G. Zhao *et al.*, "L2-ARCTIC: A non-native English speech corpus," in *Interspeech*, 2018, pp. 2783-2787. DOI: 10.21437/Interspeech.2018-1110. This reprint contains modifications to reflect the current development of the corpus.

content and accent with the target speaker's voice quality to create utterances with the target's voice but the content and pronunciation of the source speaker. When teaching a second language (L2), accent conversion can be used to create a "golden speaker," a synthesized voice that has the learner's voice quality but with a native speaker's accent (e.g., prosody, intonation, pronunciation) [12]. Several studies [10, 11] have suggested that having such a "golden speaker" to imitate can be beneficial in pronunciation training. Furthermore, in addition to providing language learners with a suitable voice to mimic, detecting mispronunciations is also a critical component for providing useful feedback to the learners in computer-assisted pronunciation training [162].

To train and evaluate voice and accent conversion systems designed for non-native speakers, one needs high-quality parallel recordings from the source and target speakers. Likewise, to develop and benchmark mispronunciation detection algorithms, detailed phoneme level annotations on pronunciation errors (e.g., phone substitution, additions, and deletions) are required. However, existing non-native English corpora (e.g., Speech Accent Archive [163] and IDEA [164]) do not fulfill these requirements (refer to Section 6.3 for a detailed discussion.)

To fill this gap, we have built a non-native English speech corpus that contains twenty-four (24) non-native speakers from six different native languages: Hindi, Korean, Mandarin, Spanish, Vietnamese, and Arabic. For *each* speaker, the corpus contains the following data:

- Speech recordings: over one hour of prompted recordings of phonetically-balanced short sentences

- Word level transcriptions: orthographic transcription and forced-aligned word boundaries for each sentence
- Phoneme level transcriptions: forced-aligned phoneme transcription for each sentence
- Manual annotations: a selected subset of utterances (~150), including 100 sentences produced by all speakers and 50 sentences that include phonemes likely to be difficult according to each speaker's L1, all annotated with corrected word and phone boundaries; phone substitution, deletion, and addition errors are also tagged

The dataset is hosted on an online archive and is freely available to the research community for non-commercial use. To the best of our knowledge, L2-ARCTIC is the first openly available corpus of its kind.

6.3. The need for a new L2 English corpus

A number of voice conversion studies [21, 92, 95, 165] have relied on the Carnegie Mellon University (CMU) ARCTIC speech corpus [105] and, more recently, the Voice Conversion Challenge (VCC) dataset [166]. However, little attention has been paid to voice conversion between non-native speakers of English, in part due to the lack of high-quality speech recordings from those speakers, despite 80% of the English speakers in the world being non-native [167]. For example, CMU ARCTIC only has a few accented English speakers[25], either native speakers of different English dialects or highly proficient non-native speakers, whereas the VCC dataset was recorded solely by professional voice

[25] JMK: Canadian accent; AWB: Scottish accent; KSP: Indian accent

talents who are native English speakers. Therefore, these standard corpora are not suitable for either voice conversion between non-native speakers or accent conversion tasks.

Among the non-native English corpora, the Speech Accent Archive [163] and IDEA [164] cover a wide range of native languages and speakers. However, each speaker only recorded a short paragraph (Speech Accent Archive) or a short free speech task (IDEA), and most of the recordings have strong background noise, making them ill-suited for voice/accent conversion. The Wildcat [168], LDC2007S08 [169], and NUFAESD [170] datasets have a limited number of recordings for each non-native speaker, and have restricted access –LDC2007S08 requires a fee, while Wildcat and NUFAESD are limited to designated research groups.

As for corpora for mispronunciation detection, the CU-CHLOE [171] and College Learners' Spoken English Corpus (COLSEC) [172] only contain speech and error tags from Chinese learners of English, and CU-CHLOE is (to our knowledge) not publicly available. The ISLE Speech Corpus [173] contains mispronunciation tags and is open for academic access, but it only focuses on a limited group of English learners (German and Italian). SingaKids-Mandarin [174] has a rich set of speech data, but it only focuses on mispronunciation patterns in Singapore children's Mandarin speech. In fact, most existing mispronunciation detection systems use their private datasets, which makes it difficult to compare experimental results across different publications [171, 175-177].

To overcome the insufficiencies outlined above, we constructed (and are now releasing) L2-ARCTIC to provide an open corpus for voice conversion between accented

speakers, accent conversion, and mispronunciation detection. Zhao et al. [84] have performed a preliminary evaluation on voice/accent conversion tasks using a subset of the speakers in L2-ARCTIC. Using a joint-density GMM with MLPG and global variance compensation [21] (128 mixtures, ~5 min of parallel training data) as the voice conversion system, they obtained 3.0 Mean Opinion Score (MOS) on the converted speech, which was also rated as similar to the target voice. Furthermore, an accent-conversion algorithm based on frame-alignment using posteriorgrams was able to generate speech that was perceived as similar to a non-native target voice but markedly less accented (98% preference compared to non-native speech). This manuscript presents preliminary results on a new task: mispronunciation detection**.**

6.4. Corpus curation procedure

The current L2-ARCTIC contains English speech of speakers from six different first languages: Hindi[26] [178], Korean, Mandarin, Spanish, Vietnamese, and Arabic. We chose these L1s because each one has a distinct foreign/non-native accent in English and provides unique challenges. **Indian** speakers of English typically have native-like English fluency but use segmental and suprasegmental features in ways that are distinct from American English. Thus, Indian speakers have both advantages in approaching pronunciation changes (e.g., familiarity and comfort with English) and disadvantages (comfort with

[26] Hindi is an Indo-Aryan language that is both an L1 and a language of wider communication. Thus, Hindi speakers in the corpus may use Hindi as an L2, speaking another Indian language as an L1. Educated Indian English is a stable contact variety of English.

their English variety makes it particularly difficult to adjust their speech to salient differences with American English.) **Korean** learners of English have a large number of high functional load consonant and vowel difficulties (errors with many minimal pairs). Prosodically, Korean and English employ suprasegmental systems that have little overlap [179, 180]. **Mandarin** (Chinese/Putonghua) learners of English have difficulty with a range of consonant and vowel sounds and in producing correct English stress, intonation, and juncture [181-183]. **Spanish** learners of English may have difficulties distinguishing a number of high functional load contrasts in English [184, 185]. Spanish is also a five-vowel language, and Spanish learners find the more complex English vowel system especially challenging. Like English, Spanish uses both word stress and nuclear stress for emphasis but, because it does not use the unstressed vowel schwa, realizes stress differently. **Vietnamese** learners of English encounter great difficulties in learning English pronunciation for multiple reasons. Like learners from other L1s, the English phonology system has a few sounds that are foreign to Vietnamese speakers. Also, Vietnamese speakers pronounce the ending sounds completely differently from native English speakers, making it difficult for them to achieve appropriate English pronunciation [186-188]. **Arabic** has significantly fewer vowels than English, and while Arabic has word stress, it does not use stress in the same way that English does [189, 190]. In the future, we may also include speakers from other L1s if we find them to be useful to the research community.

6.4.1. Participants

For the most current data release (v5.0), we recruited four speakers (two male and two female) for each of the L1s, for a total of 24 speakers[27]. Speakers were recruited from Iowa State University's student/faculty/staff body; their age range was from 22 to 43 years, with an average of 31 years (std: 6.2.) Their age of English onset ranged from 3 to 17 years with an average of 10.2 years (std: 4.3.) Detailed demographic information of the speakers is summarized in Table 6.1. The proficiency level of English was measured using TOEFL internet-Based Test (iBT) scores [109].

Table 6.1: Demographic information of the speakers. A few speakers did not report any English test score (denoted by "N/A"). Speaker ABA and THV reported their IELTS scores, and we converted them to a TOEFL iBT score following [108].

Speaker	*L1*	*Gender*	*TOEFL iBT Score*
HKK	Korean	M	114
YDCK	Korean	F	110
YKWK	Korean	M	N/A
HJK	Korean	F	115
BWC	Mandarin	M	80
LXC	Mandarin	F	86
TXHC	Mandarin	M	108
NCC	Mandarin	F	102
YBAA	Arabic	M	100
SKA	Arabic	F	79
ABA	Arabic	M	94-101
ZHAA	Arabic	F	N/A
EBVS	Spanish	M	70

[27] The speech data from the 24 speakers was released in three batches. The first release was made in the Spring of 2018, which consisted of 10 speakers from five of the six L1s. The second batch was released in Fall 2018 that included an extra ten speakers for the first five L1s. The third release (Spring 2019) added four Vietnamese speakers.

Table 6.1: Continued.

Speaker	*L1*	*Gender*	*TOEFL iBT Score*
NJS	Spanish	F	110
ERMS	Spanish	M	104
MBMPS	Spanish	F	N/A
RRBI	Hindi	M	91
TNI	Hindi	F	99
ASI	Hindi	M	101
SVBI	Hindi	F	N/A
HQTV	Vietnamese	M	81
PNV	Vietnamese	F	N/A
TLV	Vietnamese	M	79
THV	Vietnamese	F	79-93

6.4.2. Recording the corpus

To create the corpus, we used the 1,132 sentences in the CMU ARCTIC prompts. There were multiple reasons to choose these sentences. First, the ARCTIC prompts are phonetically balanced (100%, 79.6%, and 13.7% coverage for phonemes, diphones, and triphones, respectively), are open source, and can produce around one hour of edited speech. Second, the ARCTIC corpus itself has proven to work well with speech synthesis [52] and voice conversion tasks [21, 92, 95, 191]. Finally, the ARCTIC prompts are challenging for non-native English speakers so they can elicit potential pronunciation problems.

The speech was recorded in a quiet room at Iowa State University (ISU). We used a Samson C03U microphone and Earamble studio microphone pop filter for recordings; the microphone was placed 20 cm from the speaker to avoid air puffing. During each

recording session, a linguist guided the L2 speaker through the process, asking the speaker to re-record a sentence if the production contained significant disfluency or deviated from the prompt. All speakers were instructed to speak in a natural manner. The speech was sampled at 44.1 kHz and saved as a WAV file.

Once the recording was finished, we removed repetitions and false starts, performed amplitude normalization, and segmented the utterances into individual WAV files. All of the above were done in Audacity [132]. The utterances were carefully manually trimmed to remove the leading and trailing silence and non-speech sounds such as lip smacks.

6.4.3. Corpus annotations

Our corpus provides orthographic transcriptions at the word level. We used the Montreal forced-aligner [192] to produce phonetic transcriptions in PRAAT's TextGrid format [193], which contains word and phone boundaries (Figure 6.1). Further, we performed manual annotations on a selected subset of sentences for each speaker. For all the speakers, we annotated a common set of 100 sentences. In addition, we annotated 50 sentences that included phoneme difficulties that were L1-dependent. In the end, the corpus contains up to 150 curated phonetic transcriptions per speaker. Those transcriptions contain manually-adjusted word and phone boundaries, correct phoneme labels, mispronunciation error tags (phone additions, deletions, and substitutions), and comments from the annotators. To facilitate computer processing, we used the ARPAbet phoneme set for the phonetic transcriptions as well as the error tags. In the comment part of the transcriptions, however, annotators were allowed to use IPA symbols. Please refer to Appendix D for the

mapping between ARPAbet and IPA symbols. To ensure high-quality annotations, we developed automated scripts to check the annotation consistency and then asked human annotators to fix problems. The annotators (N=4) were Ph.D. students or postdoctoral fellows in the Applied Linguistics and Technology program at Iowa State University. They were experienced in transcribing speech samples of native or non-native English speakers.

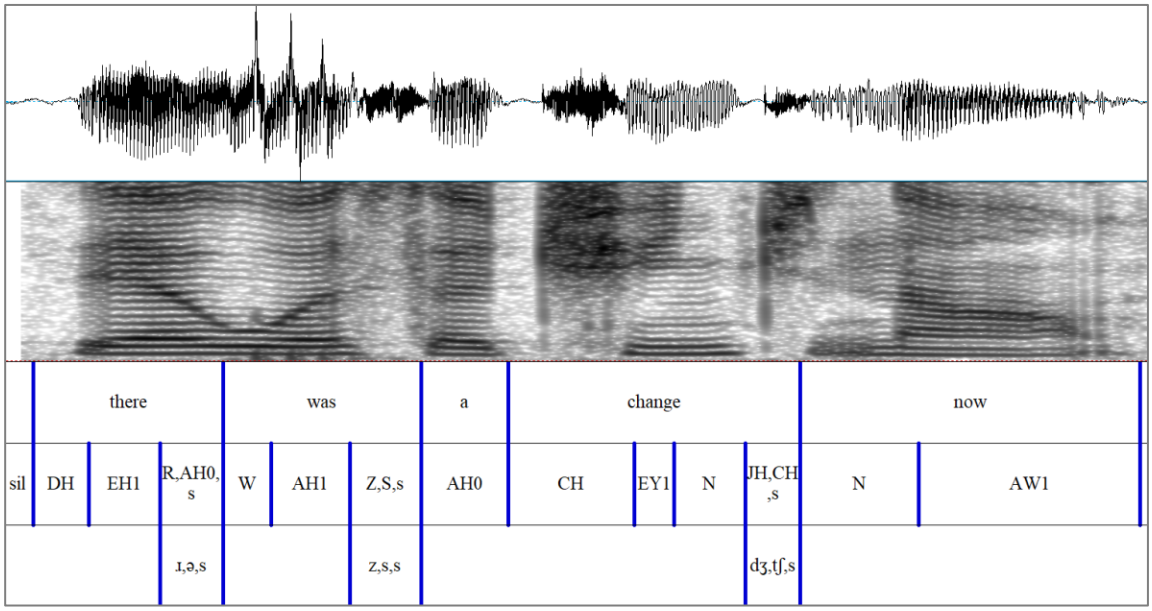

Figure 6.1: A TextGrid with manual annotations. Top to bottom: speech waveform, spectrogram, words, phonemes and error tags, comments from the annotator

6.5. Corpus statistics

In total, the dataset contains 26,867 utterances, with most speakers recording the full ARCTIC set (1,132 utterances.)[28] The total duration of the corpus is 27.1 hours, with

[28] Some speakers did not read all sentences, and a few sentences were removed for some speakers since those recordings did not have the required quality.

an average of 67.7 minutes (std: 8.6 minutes) of speech per L2 speaker. On average, each utterance is 3.6 seconds in duration. The pause before and after each utterance is generally no longer than 100 ms. Using the forced alignment results, we estimate a speech to silence ratio of 7:1 across the whole dataset. The dataset contains over 238,702 word segments, giving an average of around nine words per utterance, and over 851,830 phone segments (excluding silence). The phoneme distribution is shown in Figure 6.2.

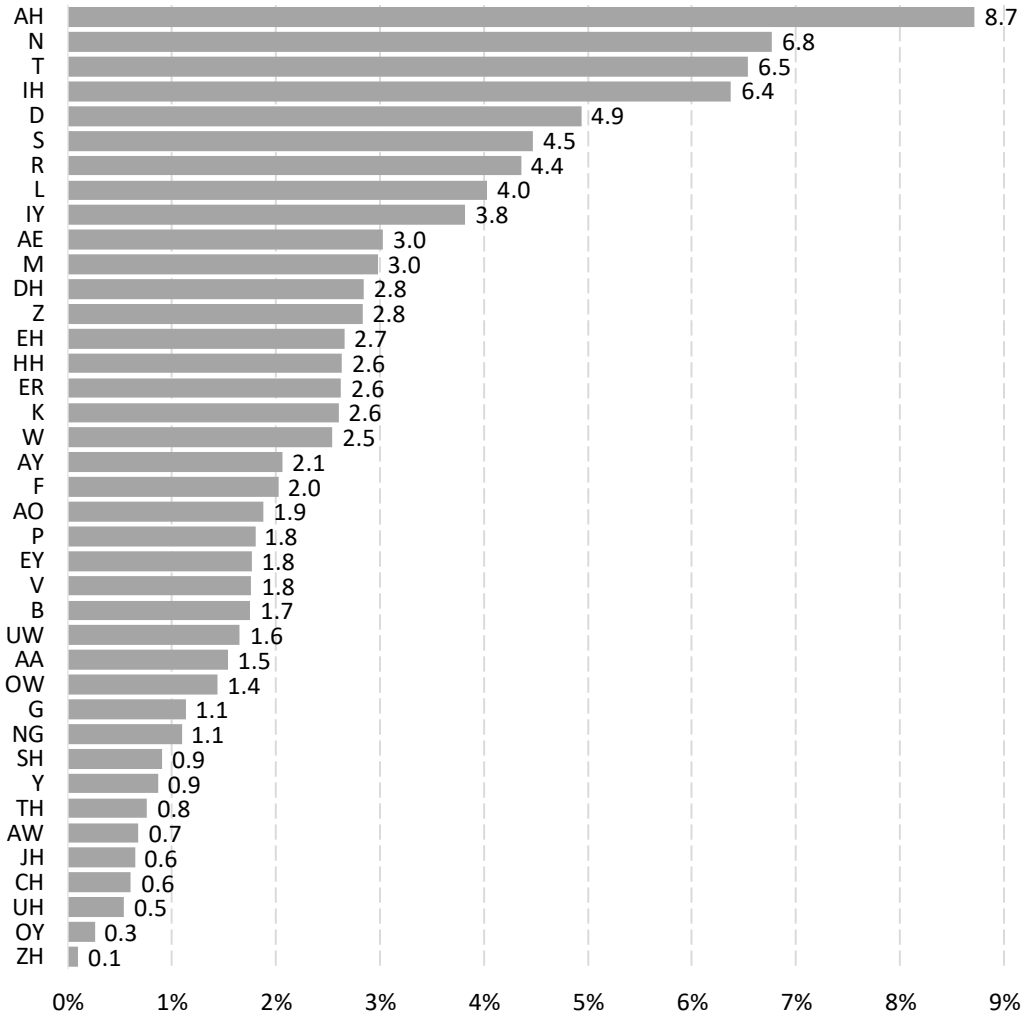

Figure 6.2: Phoneme distribution of the corpus

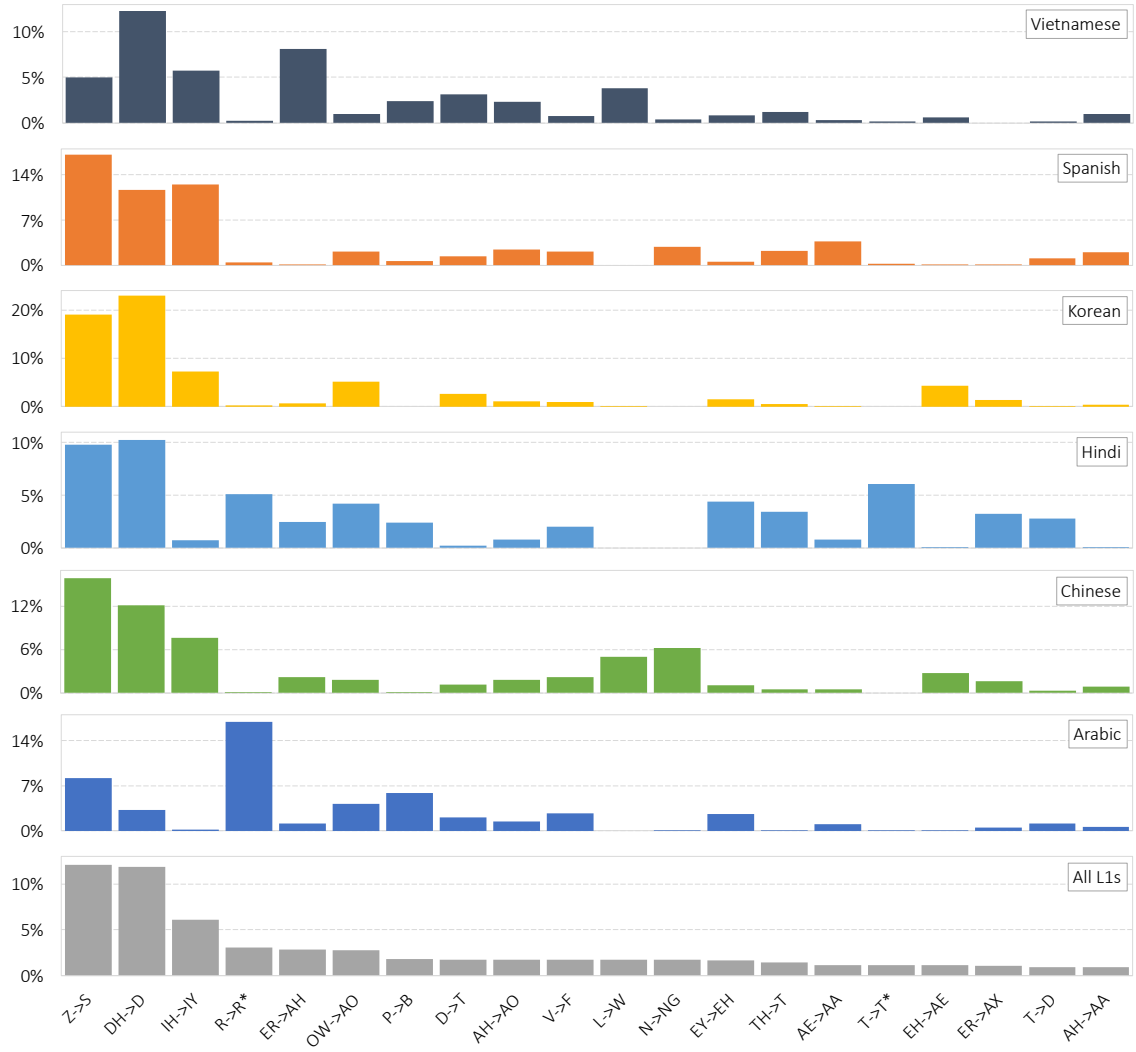

Figure 6.3: L1-dependent phone substitution error distributions and the aggregated results. Errors with low frequencies were omitted; all the values are the percentages with respect to the total number of each error type (i.e., normalized universally); Notations such as "R*" means it's a deviation from the canonical phoneme's pronunciation. In the example, it represents a deviated "R" sound.

Human annotators manually examined 3,599 utterances, annotating 14,098 phone substitutions, 3,420 phone deletions, and 1,092 phone additions. Figure 6.3 shows the top-20 most frequent phoneme substitution tags in the corpus. The most dominant substitution

errors were “Z→S,” (voicing) “DH→D,” (fricative to stop) “IH→IY,” “R→R*” (use of a deviated R sound for the American rhoticity), “ER→AH” (use of an open-mid back unrounded vowel instead of an r-colored vowel), and “OW→AO” (use of a tense vowel for lax, and vice versa.) Each contains English phoneme distinctions that lead to common substitution errors for varied American English learners.

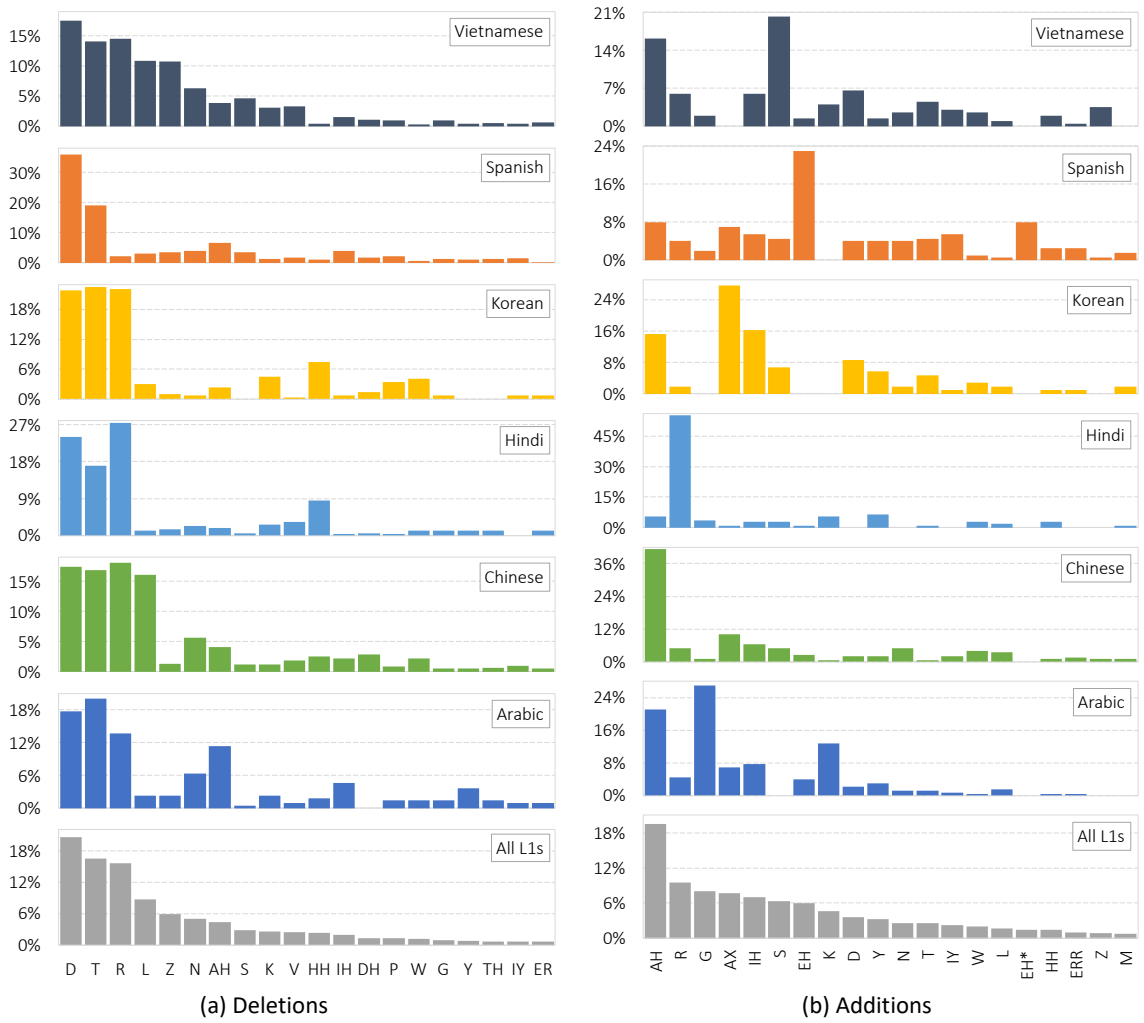

Figure 6.4: L1-dependent phone deletion and addition error distributions and the aggregated results. (a) Deletions. (b) Insertions. “ERR” means an erroneous pronunciation that is not in the ARPAbet phoneme set.

Figure 6.4 (a) shows the phone deletion errors in the annotations. In our sample group, the most frequent phoneme deletions were “D,” “T,” and “R,” almost always in non-initial position. Many non-native speakers of English do not pronounce the American English phoneme “R” in postvocalic position (e.g., in *car* and *farm.*) “T” and “D” often occur as word endings and in consonant clusters both within and across words, where they were often omitted. Figure 6.4 (b) shows the phone addition errors in the annotations. The ones that stood out were “AH,” “AX (schwa),” “IH,” “EH,” “R,” “G,” and “S.” The vowel additions simplify complex syllable structures with consonant clusters and so may serve to make the word more pronounceable.

Table 6.2 provides a breakdown of pronunciation errors by L1s. Although others have used L1 to predict L2 pronunciation errors [184, 185, 194], such predictions are often inaccurate when applied to individual learners. Thus, this list is meant to start a discussion of the types of errors that actually occur in L2-ARCTIC. For any interested readers, a detailed analysis of the pronunciation patterns of the Arabic speakers in L2-ARCTIC can be found in [195].

Table 6.2: Most frequent errors by native language; the top-5 error occurrences are listed in descending order.

L1	***Substitutions***	***Additions***	***Deletions***
Arabic	R→R*, Z→S, P→B OW→AO, DH→Z	G, AH, K, IH, AX	T, D, R, AH, N
Chinese	Z→S, DH→D, IH→IY N→NG, L→W	AH, AX, IH, N, R	R, D, T, L, N
Hindi	DH→D, Z→S, T→T* R→R*, EY→EH	R, Y, AH, K, G	R, D, T, HH, V

Table 6.2: Continued.

L1	*Substitutions*	*Additions*	*Deletions*
Korean	DH→D, Z→S, IH→IY OW→AO, EH→AE	AX, IH, AH, D, S	T, R, D, HH, K
Spanish	Z→S, IH→IY, DH→D AE→AA, N→NG	EH, AH, EH*, AX, IH	D, T, AH, IH, N
Vietnamese	DH→D, ER→AH, IH→IY Z→S, L→W	S, AH, D, IH, R	D, R, T, L, Z

6.6. Mispronunciation detection evaluation

This section provides reference results on mispronunciation detection using the 24 speakers that we have currently released. Our implementation of the mispronunciation detection system is based on the conventional Goodness of Pronunciation (GOP) method, as defined in [196]. The GOP method assigns a score for each phone segment and then uses thresholding (either phoneme-independent or phoneme-dependent thresholds) to determine the pronunciation errors. Since DNN-based acoustic models have shown to generate better GOP scores [175], we computed the GOP scores using a DNN acoustic model following the formulas proposed in [175],

$$\mathrm{GOP}(p, \boldsymbol{o}) \approx \log \frac{p(p|\boldsymbol{o}; t_s, t_e)}{\max_{q \in Q} p(q|\boldsymbol{o}; t_s, t_e)}, \tag{6.1}$$

$$\log p(p|\boldsymbol{o}; t_s, t_e) \approx \frac{1}{t_e - t_s + 1} \sum_{t=t_s}^{t_e} \log p(p|o_t)\,, \tag{6.2}$$

$$p(p|o_t) = \sum_{s \in p} p(s|o_t)\,, \tag{6.3}$$

where p is the canonical phoneme label of the given phone segment, Q is the predefined phoneme set of the language, and s is a senone that belongs to the phoneme p; $\boldsymbol{o}$ is the acoustic observation (acoustic feature frames) of the segment, and o_t is an acoustic feature frame at timestep t; t_s and t_e are the start and end frame indices of the segment, respectively; $p(s|o_t)$ is produced by the output layer of the acoustic model.

The acoustic model we used was a p-norm DNN model, as defined by Kaldi's Librispeech [157] training script[29]. It is a DNN trained with 960 hours of native English speech [104] and contains 5,816 output senones. The Word Error Rate (WER) of this acoustic model was around 5.5% on clean speech when combined with a 4-gram language model in decoding.

We used the phone-independent thresholding variation of the GOP method to make the classification decisions, i.e., if the GOP score of a phone segment was higher than a threshold $\boldsymbol{P}$, then it was accepted as correct pronunciation; otherwise, it was rejected as an error. As a preliminary result, we only focused on substitution errors since the GOP is not suited for detecting additions and deletions.

We tested the DNN-GOP method on the whole L2-ARCTIC set, i.e., 3,599 utterances. In the testing data, excluding the additions and deletion tags as well as the silences, there are 112,311 phone samples in total, where 14,098 (12.6%) were tagged as substitution errors. We set the *log* GOP threshold between -36 and 0 and made the step size 0.05. For each experiment condition, we computed the detection precision rate as N_{TN}/N and

[29] https://github.com/kaldi-asr/kaldi/blob/master/egs/librispeech/s5/local/nnet2/run_7a_960.sh

the recall rate as N_{TN}/N_{errors}, where N_{TN} is the number of correctly predicted substitution errors, N is the total number of segments predicted as substitution errors, and N_{errors} is the total number of substitution errors in the testing set. The Precision-Recall curve is shown in Figure 6.5. When we set the threshold to -2.5 (in log scale), the precision equals recall (32%). From this result, we can see that the dataset is quite challenging for the mispronunciation detection task because it contains speech data from different L1 backgrounds and recorded by speakers with a wide range of pronunciation challenges. This GOP implementation is open source and is available online[30].

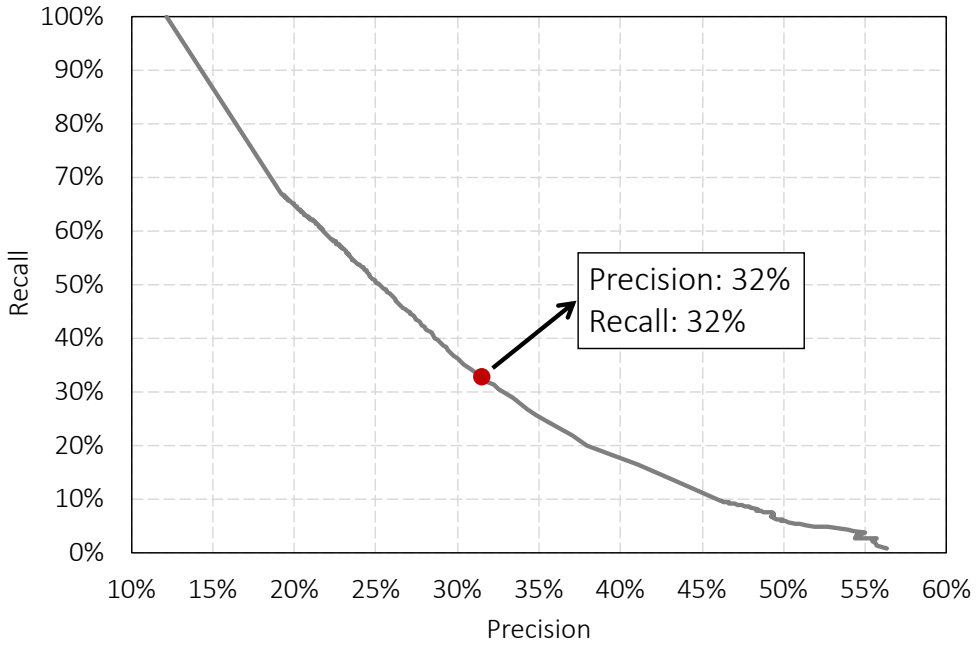

Figure 6.5: Precision-Recall curve of a phoneme-independent GOP system to demo mispronunciation detection on L2-ARCTIC

[30] https://github.com/guanlongzhao/kaldi-dnn-ali-gop

6.7. Suitcase corpus

On March 12, 2020, we released an additional "suitcase corpus" and included it in L2-ARCTIC. This portion of the L2-ARCTIC corpus involves **spontaneous** speech. We include recordings and annotations from 22 of the 24 speakers who recorded the L2-ARCTIC sentences. Speakers SKA and ASI did not participate in this task. Each speaker retold a story from a picture narrative used in applied linguistics research on comprehensibility, accentedness, and intelligibility. The pictures are generally known as the suitcase story[31]. Each retelling of the narrative was done after looking over the story and asking the researchers questions about what was happening. Few participants had questions regarding the pictures. The annotations of the suitcase corpus followed the same procedure, conventions, and standards that were applied to the scripted speech in L2-ARCTIC. The annotations were carried out by two research assistants trained in phonetic transcription. Each transcribed half of the recordings and then checked the half transcribed by the other research assistant. Finally, all transcriptions were checked by John Levis at Iowa State University, a co-PI for the project.

The total duration of the suitcase corpus is 26.1 minutes, with an average of 1.2 minutes (std: 41.5 seconds) per speaker. Using the manual annotation results, we estimate a speech-to-silence ratio of 2.3:1 across the whole dataset. The dataset contains around 3,083 word segments, giving an average of 140 words per recording, and around 9,458

[31] https://www.iris-database.org/iris/app/home/detail?id=york:822279

phone segments (excluding silence). The manual annotations include 1,673 phone substitutions, 456 phone deletions, and 90 phone additions.

6.8. Conclusion

This chapter has presented L2-ARCTIC, a new non-native English speech corpus designed for voice conversion, accent conversion, and mispronunciation detection tasks. Each speaker in L2-ARCTIC produced sufficient speech data to capture their voice identity and accent characteristics. Detailed annotations on mispronunciation errors are also included. Thus, it is possible to use this corpus to develop and evaluate mispronunciation detection algorithms. To the best of our knowledge, L2-ARCTIC is the first of its own kind, and we believe it fills gaps in both voice/accent conversion and pronunciation training. The corpus is released under the CC BY-NC 4.0 license [197] and is available at https://psi.engr.tamu.edu/l2-arctic-corpus/. Future work will be focusing on adding more speakers to the corpus.

7. CONCLUSION

7.1. Summary

In this dissertation, I develop three novel foreign accent conversion (FAC) systems to address the issues faced by previous systems. The first system trains a GMM that maps a native reference utterance to match the non-native speaker's voice identity while retaining the native accent. The GMM is trained with acoustic frame pairs between the two speakers that are aligned according to their phonetic similarity, which is measured by their symmetric KL-divergence in the PPG space. Compared with a previous accent conversion algorithm [16], which uses acoustic similarity in the MFCC space to produce the frame pairing, the new phonetic similarity frame pairing method achieves superior performance in terms of acoustic quality and nativeness. Also, I verify that the phonetic similarity frame pairing can operate on non-parallel speech corpora, and there is no statistically significant performance degradation when switching from parallel to non-parallel corpora. This finding is particularly interesting since, in real-world applications, it is difficult to assume that the accent conversion system has access to parallel corpora.

The second work builds a sequence-to-sequence speech synthesizer that can map PPGs to raw audio signals directly. By driving such a synthesizer trained on the non-native speech with PPGs extracted from native utterances, one can obtain accent conversion that can accurately capture the non-native speaker's voice identity. Experiments prove that this method can obtain significantly better voice identity and acoustic quality than the system proposed in the first work, while still being able to achieve a significantly lower foreign

accentedness rating than the non-native speech. Human listeners even rated the output accent conversion to be as natural as original unmodified natural speech, which is extremely promising. Another advantage of the system is that it does not require any data from the native reference speaker during training, and one can use any reference speaker to drive the speech synthesizer, making it possible to generate speech with different speaking styles.

In the third work, I construct a proof-of-concept FAC system that does not need a native reference utterance at synthesis time. It is one of the first few systems that are capable of performing reference-free accent conversion. Utilizing the powerful speech synthesizer proposed in the second work, I create a synthetic golden speaker using a dataset of training utterances from the non-native speaker and a native speaker. The reference-free accent conversion model is then constructed by training a sequence-to-sequence pronunciation correction model that maps the speech from the non-native speaker to the golden speaker, correcting the prosodic and segmental pronunciation errors. In this work, I first investigate different speech embeddings as the input features to construct the speech synthesizer and find that using bottleneck features can outperform the PPGs significantly in acoustic quality and accentedness ratings. I then verify that the pronunciation correction model can generate intelligible speech with a significantly less foreign accent than the input non-native speech while retaining the voice identity.

In the fourth work, I curate a first-of-its-kind speech corpus for the accent conversion task. This corpus includes speakers from a wide range of native languages and contains sufficient speech data from each speaker (~one hour of speech from each speaker) to

allow the development of modern data-driven approaches for accent conversion. More importantly, the corpus also provides detailed phoneme-level mispronunciation annotations. Those annotations can benefit other research on accent conversion since the annotations provide information on where the foreign accents occur. Most of all, we release this corpus to the public free of charge for research purposes, and we have already seen people using this corpus in research projects or as instructional materials in the classroom.

7.2. Contributions

The major contributions of this dissertation are,

- Developed a new phonetic similarity frame pairing method that reduced the foreign accentedness introduced by previous frame pairing methods
- Verified that the phonetic similarity frame pairing worked equally well on parallel and non-parallel corpora
- Constructed a sequence-to-sequence speech synthesizer for accent conversion that resolved the "third speaker" issue faced by a large portion of the existing methods, while delivering speech signals with close-to-human audio quality
- Showed that using the bottleneck feature as the speech embedding outperformed PPGs in constructing the sequence-to-sequence speech synthesizer
- Showed that it is possible to perform foreign accent conversion directly on the non-native input speech without the help of a native reference utterance, and
- Collected and released the first open-source foreign accent conversion corpus

7.3. Future work

7.3.1. Improvements on the first work

In the first work, I tested the frame-pairing method on a GMM-based spectral conversion model. However, the frame pairing is independent of the spectral conversion method since the output of the frame pairing algorithm is simply a lookup table between speech frames. Therefore, an interesting future work would be testing the phonetic similarity frame pairing with spectral conversion models that can produce better-quality speech, such as DNNs [150] and direct waveform modification [77]. Another worth noting future work is to compare different speech embeddings and distance metrics for measuring the phonetic similarity. Currently, I use the symmetric KL-divergence between the senone-PPGs as the distance measurement, and this has two potential issues. First, the senone-PPGs generally have high dimensionality; therefore, computing pair-wise phonetic similarity between a large number of speech frames is expensive even with the current optimized parallel implementation I used. Using speech embeddings with lower dimensionality would lead to significantly faster processing speed. Second, although symmetric KL-divergence is widely used to compute the distance between distributions, and computing the symmetric KL-divergence between two PPG vectors is mathematically correct, the underlying meaning of the dimensions in PPG is categorical (i.e., phonetic units) rather than numerical. Therefore, the symmetric KL-divergence may not measure the phonetic similarity accurately. For example, imagine we are computing the symmetric KL-divergence between PPG vector $a = [1, 0, 0]$ and two other vectors $b = [0, 1, 0]$ and $c = [0, 0, 1]$, where the three dimensions correspond to phoneme /ʌ/ (vowel; as in “hut”), /ɔ/

(vowel; as in “ought”), and /t/ (consonant; as in “tea”). Numerically, the symmetric KL-divergence between a and b is the same as that between a and c, which is $+\infty$. However, phonetically, a is closer to b than c because a and b are closely sounding vowels while c is a stop consonant. Therefore, more meaningful distance metrics that consider categorical differences [198] would likely lead to more accurate frame pairing and better accent conversion performance.

7.3.2. Improvements on the second work

The currently proposed method uses around one hour of speech per speaker to train the speaker-dependent speech synthesizer and neural vocoder. In future works, I would like to relax this requirement and allow training the model with fewer data from the non-native speaker. One promising research direction is to train a multi-speaker speech synthesizer [137] with the help of speaker embeddings. Speaker embeddings are high-level representations of speaker identity and are widely used in speaker recognition and verification [199, 200]. Common speaker embeddings include the i-vector [46], d-vector [201], and x-vector [202]. A multi-speaker speech synthesizer takes the speech embedding as the input and conditions its acoustic feature predictions on the *speaker* embedding of the given speaker. If the multi-speaker speech synthesizer is trained with a large number of speakers, e.g., using the VoxCeleb corpus [203], the synthesizer would be able to statistically interpolate the voice identity of the output speech given a new speaker embedding. In this case, one only needs a few utterances from the non-native speaker to extract the speaker embedding and generate the accent conversion.

Based on my preliminary examinations, a WaveGlow neural vocoder trained on a single speaker's data can generalize well to speakers from the same gender. Prior research [68, 204] studied training a WaveNet neural vocoder with limited data as well as supporting multi-speaker synthesis. One possible future direction is to extend these training techniques to the WaveGlow vocoder.

7.3.3. Improvements on the third work

Future works will focus on improving the audio quality and nativeness of the pronunciation correction model. Since the output of the pronunciation correction model can be considered as a re-synthesis of the synthetic golden speaker, the audio quality of the synthetic golden speaker speech can be treated as the upper bond for that of the accent conversions. There are two potential ways to improve the audio quality of the accent conversions. The first research direction is to improve the audio quality of the synthetic golden speaker itself. This might be possible with the multi-speaker speech synthesizer described in the previous section. The rationale is that prior research on sequence-to-sequence speech synthesizers [64, 74] has shown that when the corpus contains enough data (~24 hours of speech), the synthesized speech and original natural speech are indistinguishable to the human ear. The second research direction is applying a better pronunciation correction model to reduce the audio quality gap between the accent conversions and the synthetic golden speaker utterances. One possible improvement in the conversion model is to use a data augmentation technique proposed in [129] to stabilize the training. The data augmentation technique first uses forced alignment to segment original utterances into short fragments; it then pairs utterance fragments from the non-native and golden speakers

that contain the same linguistic content; finally, the utterance fragment pairs are added to the existing training corpus to serve as additional training data. Also, in preliminary investigations, I have found that the scheduled sampling technique [205] might have a positive effect on the generalization of the model. The scheduled sampling technique forces the model to predict the output acoustic frames in a *true* autoregressive fashion (i.e., using previously predicted outputs to generate the next one) that matches the inference process, contrary to the currently common practice that uses teacher enforcing (i.e., use the ground-truth data in the autoregression process to predict the next output frame) during training.

7.3.4. Use cross-lingual data for model training

It is generally challenging to collect speech data in a person's second language. For example, it could take more than five hours to record one hour of high-quality speech for the L2-ARCTIC speakers with the guidance of a phonetician, because it is hard for language learners to speak fluently in their second language. In contrast, it takes much less effort to collect a speech corpus in a person's native language. Therefore, it would be beneficial if we could train the accent conversion models using data collected in an L2 speaker's native tongue. Recently, a few works have used PPGs to perform cross-lingual voice conversion [130, 206] and TTS [127]. Speech embeddings such as PPGs model the underlying phonetic information of the acoustic signal. If we use a multilingual acoustic model [207, 208] to extract the speech embeddings, the resulting speech embeddings would be able to represent phonetic information for multiple languages and thus can be used for cross-lingual accent conversion. In a preliminary study conducted with the first proposed work of this dissertation, I applied the phonetic similarity frame pairing between

Portuguese and English. Initial results were promising and generated intelligible accent converted speech. Future work needs to verify this preliminary experiment formally. The second proposed work can also be extended to use cross-lingual data by using the speech embeddings extracted with a multilingual acoustic model to train the speech synthesizer.

7.3.5. Use the proposed accent conversion systems in pronunciation training

A major motivation for foreign accent conversion is that this technique might be useful for computer-assisted pronunciation training. Ding et al. [138] built a web application named Golden Speaker Builder that encapsulated a sparse-coding based accent conversion algorithm and applied the web application to a pronunciation training experiment. They found that using accent converted speech as the training material in a three-week pronunciation training study, a group of advanced Korean learners of English made significant improvements in speech comprehensibility and fluency. However, their accent conversion algorithm generated synthetic speech with artifacts and low audio quality. Thus, there remains room for improvements that might give the learners better training outcomes. During this dissertation research, I developed a new version of the Golden Speaker Builder (see Appendix B for more details), where I replaced the previous accent conversion algorithm with the one proposed in the first work. Future work will focus on testing the new version of the Golden Speaker Builder in the field, which consists of recruiting L2 learners and using the speech syntheses from accent conversion as the training materials. This research will provide an understanding of the pedagogical values of accent conversion and shed light on future research directions within accent conversion.

7.3.6. Use L2-ARCTIC in other tasks

Future works will also investigate other applications that can take advantage of the L2-ARCTIC corpus. First, since L2-ARCTIC contains a diverse range of non-native accents, researchers can use it to develop speech recognizers for accented speech [209]. Second, the detailed phoneme-level mispronunciation labels can be used to develop and evaluate new mispronunciation detection algorithms [175, 210]. Third, each speaker in the L2-ARCTIC corpus provides a relatively large amount of data suitable for developing modern TTS systems. Therefore, future research can use L2-ARCTIC data to investigate accented text-to-speech synthesis [117].

REFERENCES

[1] T. Piske, I. R. MacKay, and J. E. Flege, "Factors affecting degree of foreign accent in an L2: A review," *Journal of Phonetics,* vol. 29, no. 2, pp. 191-215, 2001.

[2] R. C. Major, *Foreign accent: The ontogeny and phylogeny of second language phonology*. Routledge, 2001.

[3] J. E. Flege, M. J. Munro, and I. R. MacKay, "Factors affecting strength of perceived foreign accent in a second language," *The Journal of the Acoustical Society of America,* vol. 97, no. 5, pp. 3125-3134, 1995.

[4] M. Munro and T. Derwing, "Foreign accent, comprehensibility, and intelligibility in the speech of second language learners," *Language Learning,* vol. 45, no. 1, pp. 73-97, 1995.

[5] J. E. Flege, "The production and perception of foreign language speech sounds," *Human communication and its disorders - a review,* pp. 224-401, 1988.

[6] A. Gluszek and J. F. Dovidio, "Accents, nonverbal behavior, and intergroup bias," in *The Handbook of Intergroup Communication*: Routledge, 2012, pp. 109-121.

[7] A. Gluszek and J. F. Dovidio, "Speaking with a nonnative accent: Perceptions of bias, communication difficulties, and belonging in the United States," *Journal of Language and Social Psychology,* vol. 29, no. 2, pp. 224-234, 2010.

[8] R. Lippi-Green, "Accent, standard language ideology, and discriminatory pretext in the courts," *Language in Society,* vol. 23, no. 2, pp. 163-198, 1994.

[9] M. Dragojevic, H. Giles, A.-C. Beck, and N. T. Tatum, "The fluency principle: Why foreign accent strength negatively biases language attitudes," *Communication Monographs,* vol. 84, no. 3, pp. 385-405, 2017.

[10] K. Probst, Y. Ke, and M. Eskenazi, "Enhancing foreign language tutors - in search of the golden speaker," *Speech Communication,* vol. 37, no. 3, pp. 161-173, 2002.

[11] M. P. Bissiri, H. R. Pfitzinger, and H. G. Tillmann, "Lexical stress training of German compounds for Italian speakers by means of resynthesis and emphasis," in *Australian International Conference on Speech Science & Technology*, 2006, pp. 24-29.

[12] D. Felps, H. Bortfeld, and R. Gutierrez-Osuna, "Foreign accent conversion in computer assisted pronunciation training," *Speech Communication,* vol. 51, no. 10, pp. 920-932, 2009.

[13] S. Aryal, D. Felps, and R. Gutierrez-Osuna, "Foreign accent conversion through voice morphing," in *Interspeech*, 2013, pp. 3077-3081.

[14] D. Felps and R. Gutierrez-Osuna, "Developing objective measures of foreign-accent conversion," *IEEE Transactions on Audio, Speech, and Language Processing,* vol. 18, no. 5, pp. 1030-1040, 2010.

[15] M. Huckvale and K. Yanagisawa, "Spoken language conversion with accent morphing," in *ISCA Workshop on Speech Synthesis*, 2007, pp. 64-70.

[16] S. Aryal and R. Gutierrez-Osuna, "Can voice conversion be used to reduce non-native accents?," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2014, pp. 7879-7883.

[17] S. Aryal and R. Gutierrez-Osuna, "Reduction of non-native accents through statistical parametric articulatory synthesis," *The Journal of the Acoustical Society of America,* vol. 137, no. 1, pp. 433-446, 2015.

[18] S. Aryal and R. Gutierrez-Osuna, "Articulatory-based conversion of foreign accents with Deep Neural Networks," in *Interspeech*, 2015, pp. 3385-3389.

[19] S. Aryal and R. Gutierrez-Osuna, "Articulatory inversion and synthesis: towards articulatory-based modification of speech," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013, pp. 7952-7956.

[20] S. Aryal and R. Gutierrez-Osuna, "Accent conversion through cross-speaker articulatory synthesis," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 7694-7698.

[21] 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.

[22] P. W. Schönle, K. Gräbe, P. Wenig, J. Höhne, J. Schrader, and B. Conrad, "Electromagnetic articulography: Use of alternating magnetic fields for tracking movements of multiple points inside and outside the vocal tract," *Brain and Language,* vol. 31, no. 1, pp. 26-35, 1987.

[23] S. Narayanan, K. Nayak, S. Lee, A. Sethy, and D. Byrd, "An approach to real-time magnetic resonance imaging for speech production," *The Journal of the Acoustical Society of America,* vol. 115, no. 4, pp. 1771-1776, 2004.

[24] T. Hazen, W. Shen, and C. White, "Query-by-example spoken term detection using phonetic posteriorgram templates," in *IEEE Workshop on Automatic Speech Recognition & Understanding (ASRU)*, 2009, pp. 421-426.

[25] J. Zhang, Z. Ling, L.-J. Liu, Y. Jiang, and L.-R. Dai, "Sequence-to-sequence acoustic modeling for voice conversion," *IEEE/ACM Transactions on Audio, Speech, and Language Processing,* vol. 27, no. 3, pp. 631-644, 2019.

[26] H. Miyoshi, Y. Saito, S. Takamichi, and H. Saruwatari, "Voice conversion using sequence-to-sequence learning of context posterior probabilities," in *Interspeech*, 2017, pp. 1268-1272.

[27] T. Kaneko, H. Kameoka, K. Hiramatsu, and K. Kashino, "Sequence-to-sequence voice conversion with similarity metric learned using generative adversarial networks," in *Interspeech*, 2017, pp. 1283-1287.

[28] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in Neural Information Processing Systems*, 2014, pp. 3104-3112.

[29] M. R. Schroeder, "The speech signal," in *Computer Speech*, vol. 35: Springer, Berlin, Heidelberg, 1999.

[30] A. Moyer, *Foreign accent: The phenomenon of non-native speech*. Cambridge University Press, 2013.

[31] E. H. Lenneberg, *Biological foundations of language*. New York: Wiley, 1967, p. 489.

[32] T. Scovel, "Foreign accents, language acquisition, and cerebral dominance," *Language Learning,* vol. 19, no. 3-4, pp. 245-253, 1969.

[33] E. H. Lenneberg, "The biological foundations of language," *Hospital Practice,* vol. 2, no. 12, pp. 59-67, 1967.

[34] M. H. Long, "Maturational constraints on language development," *Studies in Second Language Acquisition,* vol. 12, no. 3, pp. 251-285, 1990.

[35] T. Scovel, *A time to speak: A psycholinguistic inquiry into the critical period for human speech*. Newbury House Publishers, 1988.

[36] M. S. Patkowski, "Age and accent in a second language: A reply to James Emil Flege," *Applied Linguistics,* vol. 11, no. 1, pp. 73-89, 1990.

[37] S. Oyama, "The concept of the sensitive period in developmental studies," *Merrill-Palmer Quarterly of Behavior Development,* vol. 25, no. 2, pp. 83-103, 1979.

[38] J. E. Flege, "Age of learning and second language speech," in *Second Language Acquisition and The Critical Period Hypothesis*: Routledge, 1999, pp. 111-142.

[39] E. Bialystok, "The structure of age: In search of barriers to second language acquisition," *Second Language Research,* vol. 13, no. 2, pp. 116-137, 1997.

[40] B. Goldstein, "Transcription of Spanish and Spanish-influenced English," *Communication Disorders Quarterly,* vol. 23, no. 1, pp. 54-60, 2001.

[41] L. A. Helman, "Building on the sound system of Spanish: Insights from the alphabetic spellings of English-language learners," *The Reading Teacher,* vol. 57, no. 5, pp. 452-460, 2004.

[42] H. You, A. Alwan, A. Kazemzadeh, and S. Narayanan, "Pronunciation variations of Spanish-accented English spoken by young children," in *Interspeech*, 2005, pp. 749-752.

[43] K. Ohata, "Phonological differences between Japanese and English: Several potentially problematic," *Language Learning,* vol. 22, pp. 29-41.

[44] S. Duanmu, *The phonology of standard Chinese*. Oxford University Press, 2007.

[45] H. Behravan, V. Hautamäki, S. M. Siniscalchi, T. Kinnunen, and C.-H. Lee, "I-Vector modeling of speech attributes for automatic foreign accent recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing,* vol. 24, no. 1, pp. 29-41, 2016.

[46] 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, 2011.

[47] K. N. Stevens, *Acoustic phonetics*. The MIT Press, 2000.

[48] D. O'Shaughnessy, *Speech Communication: Human and Machine*. Addison-Wesley Pub. Co., 1987.

[49] T. Fukada, K. Tokuda, T. Kobayashi, and S. Imai, "An adaptive algorithm for mel-cepstral analysis of speech," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1992, vol. 1, pp. 137-140.

[50] S. B. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE*

Transactions on Acoustics, Speech, and Signal Processing vol. 28, no. 4, pp. 357-366, 1980.

[51] S. Young, "A review of large-vocabulary continuous-speech," *Signal Processing Magazine,* vol. 13, no. 5, p. 45, 1996.

[52] H. Zen *et al.*, "The HMM-based speech synthesis system (HTS) version 2.0," in *ISCA Workshop on Speech Synthesis*, 2007, pp. 294-299.

[53] H. Zen, A. Senior, and M. Schuster, "Statistical parametric speech synthesis using deep neural networks," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2013, pp. 7962-7966.

[54] H. Zen, K. Tokuda, and A. W. Black, "Statistical parametric speech synthesis," *Speech Communication,* vol. 51, no. 11, pp. 1039-1064, 2009.

[55] H. Kawahara, "Speech representation and transformation using adaptive interpolation of weighted spectrum: vocoder revisited," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 1997, vol. 2, pp. 1303-1306.

[56] Y. Ohtani, T. Toda, H. Saruwatari, and K. Shikano, "Maximum likelihood voice conversion based on GMM with STRAIGHT mixed excitation," in *Interspeech*, 2006, pp. 2266-2269.

[57] 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.

[58] M. Morise, H. Kawahara, and T. Nishiura, "Rapid F0 estimation for high-SNR speech based on fundamental component extraction," *IEICE Transactions on Information and Systems,* vol. 93, pp. 109-117, 2010.

[59] M. Morise, "CheapTrick, a spectral envelope estimator for high-quality speech synthesis," *Speech Communication,* vol. 67, pp. 1-7, 2015.

[60] M. Morise, "Platinum: A method to extract excitation signals for voice synthesis system," *Acoustical Science and Technology,* vol. 33, no. 2, pp. 123-125, 2012.

[61] M. Morise and Y. Watanabe, "Sound quality comparison among high-quality vocoders by using re-synthesized speech," *Acoustical Science and Technology,* vol. 39, no. 3, pp. 263-265, 2018.

[62] A. v. d. Oord *et al.*, "WaveNet: A generative model for raw audio," in *ISCA Workshop on Speech Synthesis*, 2016, p. 125.

[63] A. Tamamori, T. Hayashi, K. Kobayashi, K. Takeda, and T. Toda, "Speaker-dependent WaveNet vocoder," in *Interspeech*, 2017, pp. 1118-1122.

[64] R. Prenger, R. Valle, and B. Catanzaro, "WaveGlow: A flow-based generative network for speech synthesis," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2019, pp. 3617-3621.

[65] Z. Jin, A. Finkelstein, G. J. Mysore, and J. Lu, "FFTNet: A real-time speaker-dependent neural vocoder," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2018, pp. 2251-2255.

[66] J.-M. Valin and J. Skoglund, "LPCNet: Improving neural speech synthesis through linear prediction," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2019, pp. 5891-5895.

[67] A. v. d. Oord *et al.*, "Parallel WaveNet: Fast high-fidelity speech synthesis," in *International Conference on Machine Learning*, 2018, pp. 3918-3926.

[68] T. Hayashi, A. Tamamori, K. Kobayashi, K. Takeda, and T. Toda, "An investigation of multi-speaker training for WaveNet vocoder," in *IEEE Workshop on Automatic Speech Recognition & Understanding (ASRU)*, 2017, pp. 712-718.

[69] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with LSTM," *Neural Computation,* vol. 12, no. 10, pp. 2451-2471, 2000.

[70] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," in *NIPS 2014 Workshop on Deep Learning*, 2014.

[71] 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.

[72] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 4960-4964.

[73] Y. Wang *et al.*, "Tacotron: Towards end-to-end speech synthesis," in *Interspeech*, 2017, pp. 4006-4010.

[74] J. Shen *et al.*, "Natural TTS synthesis by conditioning wavenet on mel spectrogram predictions," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2018, pp. 4779-4783.

[75] D. L. Rubin and K. A. Smith, "Effects of accent, ethnicity, and lecture topic on undergraduates' perceptions of nonnative English-speaking teaching assistants," *International Journal of Intercultural Relations,* vol. 14, no. 3, pp. 337-353, 1990.

[76] S. Lev-Ari and B. Keysar, "Why don't we believe non-native speakers? The influence of accent on credibility," *Journal of Experimental Social Psychology,* vol. 46, no. 6, pp. 1093-1096, 2010.

[77] K. Kobayashi, T. Toda, G. Neubig, S. Sakti, and S. Nakamura, "Statistical singing voice conversion with direct waveform modification based on the spectrum differential," in *Interspeech*, 2014, pp. 2514-2518.

[78] T. Nakashika, T. Takiguchi, and Y. Ariki, "Voice conversion using RNN pre-trained by recurrent temporal restricted Boltzmann machines," *IEEE/ACM Transactions on Audio, Speech and Language Processing,* vol. 23, no. 3, pp. 580-587, 2015.

[79] C. Liberatore, S. Aryal, Z. Wang, S. Polsley, and R. Gutierrez-Osuna, "SABR: Sparse, Anchor-Based Representation of the speech signal," in *Interspeech*, 2015, pp. 608-612.

[80] G. Hinton *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine,* vol. 29, no. 6, pp. 82-97, 2012.

[81] I. J. Taneja, "On generalized information measures and their applications," in *Advances in Electronics and Electron Physics*, vol. 76: Elsevier, 1989, pp. 327-413.

[82] D. Felps, C. Geng, and R. Gutierrez-Osuna, "Foreign accent conversion through concatenative synthesis in the articulatory domain," *IEEE Transactions on Audio, Speech, and Language Processing,* vol. 20, no. 8, pp. 2301-2312, 2012.

[83] F.-L. Xie, F. K. Soong, and H. Li, "A KL divergence and DNN-based approach to voice conversion without parallel training sentences," in *Interspeech*, 2016, pp. 287-291.

[84] G. Zhao, S. Sonsaat, J. Levis, E. Chukharev-Hudilainen, and R. Gutierrez-Osuna, "Accent conversion using phonetic posteriorgrams," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2018, pp. 5314-5318.

[85] Q. Yan, S. Vaseghi, D. Rentzos, and C.-H. Ho, "Analysis and synthesis of formant spaces of British, Australian, and American accents," *IEEE Transactions on Audio, Speech, and Language Processing,* vol. 15, no. 2, pp. 676-689, 2007.

[86] J. Jügler, F. Zimmerer, J. Trouvain, and B. Möbius, "The perceptual effect of L1 prosody transplantation on L2 speech: The case of French accented German," in *Interspeech*, 2016, pp. 67-71.

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

[88] S. H. Mohammadi and A. Kain, "An overview of voice conversion systems," *Speech Communication,* vol. 88, pp. 65-82, 2017.

[89] D. Erro, A. Moreno, and A. Bonafonte, "Voice conversion based on weighted frequency warping," *IEEE Transactions on Audio, Speech, and Language Processing,* vol. 18, no. 5, pp. 922-931, 2010.

[90] E. Godoy, O. Rosec, and T. Chonavel, "Voice conversion using dynamic frequency warping with amplitude scaling, for parallel or nonparallel corpora," *IEEE Transactions on Audio, Speech, and Language Processing,* vol. 20, no. 4, pp. 1313-1323, 2012.

[91] T. Nakashika, T. Takiguchi, and Y. Ariki, "Voice conversion using speaker-dependent conditional restricted Boltzmann machine," *EURASIP Journal on Audio, Speech, and Music Processing,* vol. 2015, no. 8, 2015.

[92] L. Sun, S. Kang, K. Li, and H. Meng, "Voice conversion using deep bidirectional long short-term memory based recurrent neural networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 4869-4873.

[93] R. Takashima, T. Takiguchi, and Y. Ariki, "Exemplar-based voice conversion in noisy environment," in *IEEE Spoken Language Technology Workshop*, 2012, pp. 313-317.

[94] Z. Wu, E. S. Chng, and H. Li, "Exemplar-based voice conversion using joint nonnegative matrix factorization," *Multimedia Tools and Applications,* vol. 74, no. 22, pp. 9943-9958, 2015.

[95] G. Zhao and R. Gutierrez-Osuna, "Exemplar selection methods in voice conversion," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2017, pp. 5525-5529.

[96] W. Xiong *et al.*, "Toward human parity in conversational speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing,* vol. 25, no. 12, pp. 2410-2423, 2017.

[97] X. Zhang, J. Trmal, D. Povey, and S. Khudanpur, "Improving deep neural network acoustic models using generalized maxout networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 215-219.

[98] S. P. Rath, D. Povey, K. Veselý, and J. Cernocký, "Improved feature processing for deep neural networks," in *Interspeech*, 2013, pp. 109-113.

[99] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proceedings of COMPSTAT'2010*, Y. Lechevallier and G. Saporta, Eds.: Physica-Verlag HD, 2010, pp. 177-186.

[100] S. Panchapagesan and A. Alwan, "Frequency warping for VTLN and speaker adaptation by linear transformation of standard MFCC," *Computer Speech & Language,* vol. 23, no. 1, pp. 42-64, Jan 2009.

[101] D. J. Berndt and J. Clifford, "Using dynamic time warping to find patterns in time series," in *AAAI Workshop on Knowledge Discovery in Databases*, 1994, pp. 359-370.

[102] T. Toda, A. W. Black, and K. Tokuda, "Spectral conversion based on maximum likelihood estimation considering global variance of converted parameter," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2005, vol. 1, pp. I/9-I12 Vol. 1.

[103] S. J. Young, J. J. Odell, and P. C. Woodland, "Tree-based state tying for high accuracy acoustic modelling," in *Proceedings of the Workshop on Human Language Technology*, 1994, pp. 307-312: Association for Computational Linguistics.

[104] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 5206-5210.

[105] J. Kominek and A. W. Black, "The CMU ARCTIC speech databases," in *ISCA Workshop on Speech Synthesis*, 2004, pp. 223-224.

[106] G. Zhao *et al.*, "L2-ARCTIC: A non-native English speech corpus," in *Interspeech*, 2018, pp. 2783-2787.

[107] M. Chalhoub-Deville and C. E. Turner, "What to look for in ESL admission tests: Cambridge certificate exams, IELTS, and TOEFL," *System,* vol. 28, no. 4, pp. 523-539, 2000.

[108] ETS, "Linking TOEFL iBT Scores to IELTS Scores - A Research Report," 2010.

[109] Y. Cho and B. Bridgeman, "Relationship of TOEFL iBT® scores to academic performance: Some evidence from American universities," *Language Testing,* vol. 29, no. 3, pp. 421-442, 2012.

[110] H. Kawahara, A. d. Cheveigné, H. Banno, T. Takahashi, and T. Irino, "Nearly defect-free F0 trajectory extraction for expressive speech modifications based on STRAIGHT," in *Interspeech*, 2005, pp. 537-540.

[111] M. Morise, "D4C, a band-aperiodicity estimator for high-quality speech synthesis," *Speech Communication,* vol. 84, pp. 57-65, 2016.

[112] H. Kawahara, M. Morise, T. Takahashi, R. Nisimura, T. Irino, and H. Banno, "TANDEM-STRAIGHT: A temporally stable power spectral representation for periodic signals and applications to interference-free spectrum, F0, and aperiodicity estimation," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2008, pp. 3933-3936.

[113] S. Buchholz and J. Latorre, "Crowdsourcing preference tests, and how to detect cheating," in *Interspeech*, 2011, pp. 3053-3056.

[114] M. J. Munro, T. M. Derwing, and C. S. Burgess, "The detection of foreign accent in backwards speech," in *Proceedings of the 15th International Congress of Phonetic Sciences*, 2003, pp. 535-538.

[115] M. J. Munro, T. M. Derwing, and C. S. Burgess, "Detection of nonnative speaker status from content-masked speech," *Speech Communication,* vol. 52, no. 7-8, pp. 626-637, 2010.

[116] J. Yamagishi, "Personal communication at ICASSP'18," ed. Calgary, Canada, 2018.

[117] G. E. Henter, X. Wang, J. Lorenzo-Trueba, M. Kondo, and J. Yamagishi, "Cyborg speech: Deep multilingual speech synthesis for generating segmental foreign

accent with natural prosody," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2018, pp. 4799-4803.

[118] N. Umeda, "Vowel duration in American English," *The Journal of the Acoustical Society of America,* vol. 58, no. 2, pp. 434-445, 1975.

[119] N. Umeda, "Consonant duration in American English," *The Journal of the Acoustical Society of America,* vol. 61, no. 3, pp. 846-858, 1977.

[120] J. Lorenzo-Trueba *et al.*, "The voice conversion challenge 2018: Promoting development of parallel and nonparallel methods," in *Odyssey: The Speaker and Language Recognition Workshop*, 2018, pp. 195-202.

[121] I. M. Quintanilha, L. W. P. Biscainho, and S. L. Netto, "Towards an end-to-end speech recognizer for Portuguese using deep neural networks," in *Simpósio Brasileiro de Telecomunicações e Processamento de Sinais*, 2017, pp. 408-412.

[122] M. M. Azevedo, *A contrastive phonology of Portuguese and English*. Georgetown University Press, 1981.

[123] S. Ding *et al.*, "Golden Speaker Builder: an interactive online tool for L2 learners to build pronunciation models," in *Pronunciation in Second Language Learning and Teaching*, 2017, pp. 25-26.

[124] S. Zhao, S. N. Koh, S. I. Yann, and K. K. Luke, "Feedback utterances for computer-adied language learning using accent reduction and voice conversion method," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013, pp. 8208-8212.

[125] Y. Miao, M. Gowayyed, and F. Metze, "EESEN: End-to-end speech recognition using deep RNN models and WFST-based decoding," in *IEEE Workshop on Automatic Speech Recognition & Understanding (ASRU)*, 2015, pp. 167-174.

[126] A. Lee, Y. Zhang, and J. Glass, "Mispronunciation detection via dynamic time warping on deep belief network-based posteriorgrams," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013, pp. 8227-8231.

[127] L. Sun, H. Wang, S. Kang, K. Li, and H. M. Meng, "Personalized, cross-lingual TTS using phonetic posteriorgrams," in *Interspeech*, 2016, pp. 322-326.

[128] L. Sun, K. Li, H. Wang, S. Kang, and H. Meng, "Phonetic posteriorgrams for many-to-one voice conversion without parallel data training," in *IEEE International Conference on Multimedia and Expo (ICME)*, 2016, pp. 1-6.

[129] J.-X. Zhang, Z.-H. Ling, Y. Jiang, L.-J. Liu, C. Liang, and L.-R. Dai, "Improving sequence-to-sequence acoustic modeling by adding text-supervision," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2019, pp. 6785-6789.

[130] Y. Zhou, X. Tian, H. Xu, R. K. Das, and H. Li, "Cross-lingual voice conversion with bilingual phonetic posteriorgram and average modeling," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2019, pp. 6790-6794.

[131] D. P. Kingma and P. Dhariwal, "Glow: Generative flow with invertible 1x1 convolutions," in *Advances in Neural Information Processing Systems*, 2018, pp. 10236-10245.

[132] Audacity® [Online]. Available: http://www.audacityteam.org/

[133] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980,* 2014.

[134] 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.

[135] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning*, 2015, pp. 448-456.

[136] G. Zhao and R. Gutierrez-Osuna, "Using phonetic posteriorgram based frame pairing for segmental accent conversion," *IEEE/ACM Transactions on Audio, Speech, and Language Processing,* vol. 27, no. 10, pp. 1649-1660, 2019.

[137] Y. Jia *et al.*, "Transfer learning from speaker verification to multispeaker text-to-speech synthesis," in *Advances in Neural Information Processing Systems*, 2018, pp. 4485-4495.

[138] S. Ding *et al.*, "Golden speaker builder - An interactive tool for pronunciation training," *Speech Communication,* vol. 115, pp. 51-66, 2019.

[139] R. Wang and J. Lu, "Investigation of golden speakers for second language learners from imitation preference perspective by voice modification," *Speech Communication,* vol. 53, no. 2, pp. 175-184, 2011.

[140] O. Turk and L. M. Arslan, "Subband based voice conversion," in *Seventh International Conference on Spoken Language Processing*, 2002.

[141] Y. Oshima, S. Takamichi, T. Toda, G. Neubig, S. Sakti, and S. Nakamura, "Non-native speech synthesis preserving speaker individuality based on partial correction of prosodic and phonetic characteristics," in *Interspeech*, 2015, pp. 299-303.

[142] F. Biadsy, R. J. Weiss, P. J. Moreno, D. Kanvesky, and Y. Jia, "Parrotron: An end-to-end speech-to-speech conversion model and its applications to hearing-impaired speech and speech separation," in *Interspeech*, 2019, pp. 4115-4119.

[143] M. Mimura, S. Sakai, and T. Kawahara, "Forward-backward attention decoder," in *Interspeech*, 2018, pp. 2232-2236.

[144] Y. Zheng *et al.*, "Forward-backward decoding for regularizing end-to-end TTS," in *Interspeech*, 2019, pp. 1283-1287.

[145] M. Brand, "Voice puppetry," in *26th Annual Conference on Computer Graphics and Interactive Techniques*, 1999, pp. 21-28.

[146] B. Denby and M. Stone, "Speech synthesis from real time ultrasound images of the tongue," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2004, vol. 1, pp. I-685.

[147] R. Mumtaz, S. Preuß, C. Neuschaefer-Rube, C. Hey, R. Sader, and P. Birkholz, "Tongue contour reconstruction from optical and electrical palatography," *IEEE Signal Processing Letters,* vol. 21, no. 6, pp. 658-662, 2014.

[148] A. Toutios, T. Sorensen, K. Somandepalli, R. Alexander, and S. S. Narayanan, "Articulatory synthesis based on real-time magnetic resonance imaging data," in *Interspeech*, 2016, pp. 1492-1496.

[149] G. Zhao, S. Ding, and R. Gutierrez-Osuna, "Foreign accent conversion by synthesizing speech from phonetic posteriorgrams," in *Interspeech*, 2019, pp. 2843-2847.

[150] S. H. Mohammadi and A. Kain, "Voice conversion using deep neural networks with speaker-independent pre-training," in *IEEE Spoken Language Technology Workshop*, 2014, pp. 19-23.

[151] J. Zhang, Z. Ling, and L.-R. Dai, "Non-parallel sequence-to-sequence voice conversion with disentangled linguistic and speaker representations," *IEEE/ACM Transactions on Audio, Speech, and Language Processing,* vol. 28, pp. 540-552, 2019.

[152] K. Tanaka, H. Kameoka, T. Kaneko, and N. Hojo, "AttS2S-VC: Sequence-to-sequence voice conversion with attention and context preservation mechanisms," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2019, pp. 6805-6809.

[153] H. Kameoka, K. Tanaka, T. Kaneko, and N. Hojo, "ConvS2S-VC: Fully convolutional sequence-to-sequence voice conversion," *arXiv preprint arXiv:1811.01609,* 2018.

[154] S. Liu *et al.*, "End-to-end accent conversion without using native utterances," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2020, pp. 6289-6293.

[155] D. Povey, G. Cheng, Y. Wang, K. Li, H. Xu, and S. Khudanpur, "Semi-orthogonal low-rank matrix factorization for deep neural networks," in *Interspeech*, 2018, pp. 3743-3747.

[156] V. Peddinti, D. Povey, and S. Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts," in *Interspeech*, 2015, pp. 3214-3218.

[157] D. Povey *et al.*, "The Kaldi speech recognition toolkit," in *IEEE Workshop on Automatic Speech Recognition & Understanding (ASRU)*, 2011.

[158] J.-X. Zhang, Z.-H. Ling, and L.-R. Dai, "Forward attention in sequence-to-sequence acoustic modeling for speech synthesis," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2018, pp. 4789-4793.

[159] A. Paszke *et al.*, "PyTorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems*, 2019, pp. 8024-8035.

[160] K. Tokuda *et al.* (2017). *Speech Signal Processing Toolkit (SPTK) version 3.11*. Available: http://sp-tk.sourceforge.net/

[161] M. Morise, "Harvest: A high-performance fundamental frequency estimator from speech signals," in *Interspeech*, 2017, pp. 2321-2325.

[162] J. Levis, "Computer technology in teaching and researching pronunciation," *Annual Review of Applied Linguistics,* vol. 27, pp. 184-202, 2007.

[163] S. Weinberger. Speech accent archive [Online]. Available: http://accent.gmu.edu

[164] P. Meier. IDEA: International Dialects of English Archive [Online]. Available: http://www.dialectsarchive.com/

[165] Y.-C. Wu, H.-T. Hwang, C.-C. Hsu, Y. Tsao, and H.-M. Wang, "Locally linear embedding for exemplar-based spectral conversion," in *Interspeech*, 2016, pp. 1652-1656.

[166] T. Toda *et al.*, "The Voice Conversion Challenge 2016," in *Interspeech*, 2016, pp. 1632-1636.

[167] G. Braine, *Nonnative Speaker English Teachers: Research, Pedagogy, and Professional Growth* (ESL & Applied Linguistics Professional Series). New York and London: Routledge, 2010.

[168] K. J. Van Engen, M. Baese-Berk, R. E. Baker, A. Choi, M. Kim, and A. R. Bradlow, "The Wildcat Corpus of native-and foreign-accented English: Communicative efficiency across conversational dyads with varying language alignment profiles," *Language and Speech,* vol. 53, no. 4, pp. 510-540, 2010.

[169] T. Lander. CSLU: Foreign Accented English Release 1.2 LDC2007S08 [Online]. Available: https://catalog.ldc.upenn.edu/ldc2007s08

[170] T. Bent and A. R. Bradlow, "The interlanguage speech intelligibility benefit," *The Journal of the Acoustical Society of America,* vol. 114, no. 3, pp. 1600-1610, 2003.

[171] K. Li, X. Qian, and H. Meng, "Mispronunciation detection and diagnosis in L2 English speech using multidistribution deep neural networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing,* vol. 25, no. 1, pp. 193-207, 2017.

[172] H. Yang and N. Wei, "Construction and data analysis of a Chinese learner spoken English corpus," ed: Shanhai Foreign Languse Eduacation Press, 2005.

[173] W. Menzel *et al.*, "The ISLE corpus of non-native spoken English," in *Proceedings of LREC 2000: Language Resources and Evaluation Conference, vol. 2*, 2000, pp. 957-964.

[174] N. F. Chen, R. Tong, D. Wee, P. X. Lee, B. Ma, and H. Li, "SingaKids-Mandarin: Speech corpus of Singaporean children speaking Mandarin Chinese," in *Interspeech*, 2016, pp. 1545-1549.

[175] W. Hu, Y. Qian, F. K. Soong, and Y. Wang, "Improved mispronunciation detection with deep neural network trained acoustic models and transfer learning based logistic regression classifiers," *Speech Communication,* vol. 67, pp. 154-166, 2015.

[176] Y.-B. Wang and L.-s. Lee, "Supervised detection and unsupervised discovery of pronunciation error patterns for computer-assisted language learning," *IEEE/ACM*

Transactions on Audio, Speech and Language Processing, vol. 23, no. 3, pp. 564-579, 2015.

[177] H. Huang, H. Xu, X. Wang, and W. Silamu, "Maximum F1-score discriminative training criterion for automatic mispronunciation detection," *IEEE/ACM Transactions on Audio, Speech, and Language Processing,* vol. 23, no. 4, pp. 787-797, 2015.

[178] P. Pramod, "Indian English pronunciation," in *The Handbook of English Pronunciation*, M. Reed and J. Levis, Eds.: Wiley Blackwell, 2015, pp. 301-319.

[179] S.-A. Jun, "Prosody in sentence processing: Korean vs. English," *UCLA Working Papers in Phonetics,* vol. 104, pp. 26-45, 2005.

[180] M. Ueyama and S.-A. Jun, "Focus realization of Japanese English and Korean English intonation," *UCLA Working Papers in Phonetics,* pp. 110-125, 1996.

[181] J. Anderson‐Hsieh, R. Johnson, and K. Koehler, "The relationship between native speaker judgments of nonnative pronunciation and deviance in segmentais, prosody, and syllable structure," *Language Learning,* vol. 42, no. 4, pp. 529-555, 1992.

[182] M. C. Pennington and N. C. Ellis, "Cantonese speakers' memory for English sentences with prosodic cues," *The Modern Language Journal,* vol. 84, no. 3, pp. 372-389, 2000.

[183] J. Chang, "Chinese speakers," *Learner English,* vol. 2, pp. 310-324, 1987.

[184] J. Morley, "Teaching American English pronunciation," *TESOL Quarterly,* vol. 27, no. 4, pp. 759-761, 1993.

[185] B. Smith, *Learner English: A teacher's guide to interference and other problems*. Cambridge University Press, 2001.

[186] H. C. Tam, "Common pronunciation problems of Vietnamese learners of English," *VNU Journal of Foreign Studies,* vol. 21, no. 1, 2005.

[187] Z. Patil, "Rethinking the objectives of teaching English in Asia," *Asian EFL Journal,* vol. 10, no. 4, pp. 227-240, 2008.

[188] N. Nguyen, "Interlanguage phonology and the pronunciation of English final consonant clusters by native speakers of Vietnamese," Unpublished Master's Thesis, 2008.

[189] M. Benrabah, "Word-stress - a source of unintelligibility in English," *IRAL-International Review of Applied Linguistics in Language Teaching,* vol. 35, no. 3, pp. 157-166, 1997.

[190] K. De Jong and B. A. Zawaydeh, "Stress, duration, and intonation in Arabic word-level prosody," *Journal of Phonetics,* vol. 27, no. 1, pp. 3-22, 1999.

[191] D. Erro, E. Navas, and I. Hernaez, "Parametric voice conversion based on bilinear frequency warping plus amplitude scaling," *IEEE Transactions on Audio, Speech, and Language Processing,* vol. 21, no. 3, pp. 556-566, 2013.

[192] 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.

[193] P. Boersma, "Praat, a system for doing phonetics by computer," *Glot International,* vol. 5, no. 9, pp. 341-345, 2001.

[194] M. Munro, "How well can we predict L2 learners' pronunciation difficulties?," *CATESOL Journal,* vol. 30, no. 1, pp. 267-282, 2018.

[195] I. Lučić Rehman, A. Silpachai, J. Levis, G. Zhao, and R. Gutierrez-Osuna, "The English pronunciation of Arabic speakers - A data-driven approach to segmental error identification," *Language Teaching Research,* in press.

[196] S. M. Witt and S. J. Young, "Phone-level pronunciation scoring and assessment for interactive language learning," *Speech Communication,* vol. 30, no. 2, pp. 95-108, 2000.

[197] Creative Commons Attribution-NonCommercial 4.0 International Public License [Online]. Available: https://creativecommons.org/licenses/by-nc/4.0/legalcode

[198] S. Boriah, V. Chandola, and V. Kumar, "Similarity measures for categorical data: A comparative evaluation," in *SIAM International Conference on Data Mining*, 2008, pp. 243-254.

[199] F. Richardson, D. Reynolds, and N. Dehak, "Deep neural network approaches to speaker and language recognition," *IEEE Signal Processing Letters,* vol. 22, no. 10, pp. 1671-1675, 2015.

[200] M. Li, J. Kim, A. Lammert, P. K. Ghosh, V. Ramanarayanan, and S. Narayanan, "Speaker verification based on the fusion of speech acoustics and inverted articulatory signals," *Computer Speech & Language,* vol. 36, pp. 196-211, 2016.

[201] E. Variani, X. Lei, E. McDermott, I. L. Moreno, and J. Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," in

IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2014, pp. 4052-4056.

[202] D. Snyder, D. Garcia-Romero, A. McCree, G. Sell, D. Povey, and S. Khudanpur, "Spoken language recognition using X-vectors," in *Odyssey: The Speaker and Language Recognition Workshop*, 2018, pp. 105-111.

[203] A. Nagrani, J. S. Chung, W. Xie, and A. Zisserman, "Voxceleb: Large-scale speaker verification in the wild," *Computer Speech & Language,* vol. 60, p. 101027, 2020.

[204] L.-J. Liu, Z.-H. Ling, Y. Jiang, M. Zhou, and L.-R. Dai, "WaveNet vocoder with limited training data for voice conversion," in *Interspeech*, 2018, pp. 1983-1987.

[205] S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer, "Scheduled sampling for sequence prediction with recurrent neural networks," in *Advances in Neural Information Processing Systems*, 2015, pp. 1171-1179.

[206] Y. Zhou, X. Tian, R. K. Das, and H. Li, "Many-to-many cross-lingual voice conversion with a jointly trained speaker embedding network," *IEEE APSIPA ASC,* 2019.

[207] J. Cui *et al.*, "Multilingual representations for low resource speech recognition and keyword search," in *IEEE Workshop on Automatic Speech Recognition & Understanding (ASRU)*, 2015, pp. 259-266.

[208] S. Toshniwal *et al.*, "Multilingual speech recognition with a single end-to-end model," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2018, pp. 4904-4908.

[209] J. Shor *et al.*, "Personalizing ASR for dysarthric and accented speech with limited data," in *Interspeech*, 2019, pp. 784-788.

[210] W. Li, S. M. Siniscalchi, N. F. Chen, and C.-H. Lee, "Improving non-native mispronunciation detection and enriching diagnostic feedback with DNN-based speech attribute modeling," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 6135-6139.

[211] Z. Z. Wu, T. Kinnunen, E. S. Chng, and H. Z. Li, "Text-independent F0 transformation with non-parallel data for voice conversion," in *Interspeech*, 2010, pp. 1732-1735.

[212] A. Krogh and J. A. Hertz, "A simple weight decay can improve generalization," in *Advances in Neural Information Processing Systems*, 1992, pp. 950-957.

[213] S. Kanai, Y. Fujiwara, and S. Iwamura, "Preventing gradient explosions in gated recurrent units," in *Advances in Neural Information Processing Systems*, 2017, pp. 435-444.

[214] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450,* 2016.

APPENDIX A

LIST OF PUBLICATIONS

Following is the list of publications related to this dissertation work.

Journal articles

1. **G. Zhao** and R. Gutierrez-Osuna, "Using phonetic posteriorgram based frame pairing for segmental accent conversion," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 10, pp. 1649-1660, 2019.
2. S. Ding, **G. Zhao**, C. Liberatore, and R. Gutierrez-Osuna, "Learning structured sparse representations for voice conversion," *IEEE/ACM Transactions on Audio, Speech, and Language Processing,* vol. 28, pp. 343-354, 2019.
3. S. Ding, C. Liberatore, S. Sonsaat, I. Lučić Rehman, A. Silpachai, **G. Zhao**, E. Chukharev-Hudilainen, J. Levis, and R. Gutierrez-Osuna, "Golden speaker builder–An interactive tool for pronunciation training," *Speech Communication,* vol. 115, pp. 51-66, 2019.
4. I. Lučić Rehman, A. Silpachai, J. Levis, **G. Zhao**, and R. Gutierrez-Osuna, "The English pronunciation of Arabic speakers–A data-driven approach to segmental error identification," *Language Teaching Research,* in press.

Conference proceedings

1. S. Ding, C. Liberatore, **G. Zhao**, S. Sonsaat, E. Chukharev-Hudilainen, J. Levis, and R. Gutierrez-Osuna, "Golden Speaker Builder: an interactive online tool for

L2 learners to build pronunciation models," in *Pronunciation in Second Language Learning and Teaching*, 2017, pp. 25-26.

2. **G. Zhao** and R. Gutierrez-Osuna, "Exemplar selection methods in voice conversion," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2017, pp. 5525-5529.
3. G. Angello, A. B. Manam, **G. Zhao**, and R. Gutierrez-Osuna, "Training behavior of successful tacton-phoneme learners," presented at the IEEE Haptics Symposium (WIP), 2018.
4. **G. Zhao**, S. Sonsaat, J. Levis, E. Chukharev-Hudilainen, and R. Gutierrez-Osuna, "Accent conversion using phonetic posteriorgrams," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2018, pp. 5314-5318.
5. C. Liberatore, **G. Zhao**, and R. Gutierrez-Osuna, "Voice conversion through residual warping in a sparse, anchor-based representation of speech," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 5284-5288.
6. S. Ding, **G. Zhao**, C. Liberatore, and R. Gutierrez-Osuna, "Improving sparse representations in exemplar-based voice conversion with a phoneme-selective objective function," in *Interspeech*, 2018, pp. 476-480.
7. **G. Zhao** *et al.*, "L2-ARCTIC: A non-native English speech corpus," in *Interspeech*, 2018, pp. 2783-2787.

8. **G. Zhao**, S. Ding, and R. Gutierrez-Osuna, "Foreign accent conversion by synthesizing speech from phonetic posteriorgrams," in *Interspeech*, 2019, pp. 2843-2847.

Under Review

1. A. Silpachai, I. Lučić Rehman, T. A. Barriuso, J. Levis, **G. Zhao**, E. Chukharev-Khudilaynen, and R. Gutierrez-Osuna, "The effect of voice type and task on L2 learners' awareness of pronunciation errors," submitted to *Language Awareness*. (Journal article)
2. **G. Zhao**, S. Ding, and R. Gutierrez-Osuna, "Reference-free foreign accent conversion," submitted to *IEEE/ACM Transactions on Audio, Speech, and Language Processing*. (Journal article)
3. S. Ding, **G. Zhao**, and R. Gutierrez-Osuna, "Improving the speaker identity of non-parallel many-to-many voice conversion with adversarial speaker recognition," submitted to *Interspeech*, 2020. (Conference article)
4. A. Das, **G. Zhao**, and R. Gutierrez-Osuna, "Understanding the effect of voice quality and accent on talker similarity," submitted to *Interspeech*, 2020. (Conference article)

APPENDIX B

GOLDEN SPEAKER BUILDER BACKEND SYSTEM

During this dissertation study, I helped develop a web application called Golden Speaker Builder (GSB) to allow a naïve user to build a golden speaker of their own. The web application is hosted on https://goldenspeaker.engl.iastate.edu/speech/.

The web application was implemented in the Django framework. It has a user interfacing frontend and a signal processing backend. The frontend was written in HTML5 and JavaScript, and decorated with Bootstrap and CSS. The frontend handles the following functionalities: login, record sentences, edit recordings, build "Golden Speaker," and practice with "Golden Speaker." The web application was hosted through Nginx. For more information about the frontend part of the web application, please refer to [138]. Shaojin Ding developed the frontend.

The signal processing backend is an implementation of the algorithm proposed in Chapter 3. The backend can be further split into the training and inference part. The training part takes in a set of recordings from the native reference speaker (the *teacher*) and the non-native speaker (the *student*), producing an accent conversion model. The inference part takes an utterance from the teacher and passes it through the accent conversion model and generates the accent conversion synthesis.

The entire backend codebase lives on https://github.com/guanlongzhao/ppg-gmm. The backend runs on a Linux server and invokes Matlab and Kaldi [157] for most of the

operations. The backend uses Matlab as the primary interface since most of the dependencies were written in Matlab. All source code files contain detailed documentation.

B.1. Step 1: Installation

To install the backend system, simply clone the entire codebase from GitHub to the host machine, and then follow the most current installation instructions included in the README file. Please consult `script/demo.m` for a thorough end-to-end walkthrough of the main functionalities of this software package.

B.2. Step 2: Feature extraction

This step calls function `dataPrep` to pre-cache the necessary data files for model training. The input to this step is a list of audio files and their corresponding orthographic transcriptions. Please read the API documentation in the `dataPrep` function to learn about its syntax and other input parameters. The output of this step is a list of cache files that contain all the necessary features for the corresponding audio files. The function will first resample all input audio files into 16 kHz (for compatibility reasons) and normalize all the transcriptions, then produce the following features,

- PPG: 5816-dim phonetic posteriorgram produced by calling Kaldi's binary tools
- Forced-alignment: frame-level phoneme labels produced by calling the Montreal Forced Aligner [192]
- Acoustic features: produced by the WORLD vocoder [57]. The features include: 513-dim spectral envelope; 2-dim band-aperiodicity; 1-dim fundamental frequency (F_0); 1-dim binary voicing indicator (vuv); and 25-dim mel-cepstrum (extracted from the spectral envelope using SPTK [160])

The output cache files are then stored on the Linux file system, and the `dataPrep` function returns their paths. We perform this feature extraction step for both the teacher and student utterances.

B.3. Step 3: Train the accent conversion model

This step takes pre-processed training data from the teacher and the student to produce the accent conversion model, which consists of a spectral conversion sub-model and a pitch conversion sub-model. Please read the documentation in the referenced functions for their input and output parameters.

B.3.1. Train the spectral conversion module

This step calls `buildGMMmodelGSB`. We first use `framePairingPPG` to create the frame pairing outlines in equations (3.3) -(3.5). The implementation of the frame pairing uses a highly optimized pairwise symmetric KL-divergency routine (`KLDiv5`), which benefits from vectorized computations. The routine has a concise implementation as follows,

```
% In MATLAB syntax
% x: d * m matrix, each column is an input vector
% y: d * n matrix, each column is an input vector
% D: m * n matrix, D(i, j) = KL(x(:, i), y(:, j))

function D = KLDiv5(x, y)
    logx = log(x + eps);
    logy = log(y + eps);
    D = bsxfun(@plus, dot(y, logy, 1), dot(x, logx, 1)')...
    - x' * logy - logx' * y;
end
```

We then train a joint GMM model on the resulting frame pairs. The outputs of this step are the GMM model parameters saved in a Matlab object.

B.3.2. Train the pitch conversion module

This step calls `buildPitchModelGSB`, which supports building the pitch model in two different modes. The first one is the standard mean-and-variance normalization method introduced in Section 3.4.2.4. The second (and default) mode is the histogram equalization method proposed by Wu et al. [211], which works better than the mean-and-variance normalization approach.

B.3. Step 3: Inference

This step calls `voiceConversionInterfaceGSB`. The function changes the input teacher utterance's voice identity to match the learner's using the pre-trained spectral and pitch conversion models.

APPENDIX C

PRACTICAL MODEL-BUILDING STRATEGIES

In the process of building machine learning models, many tricks and conventions are generally not included in the paper descriptions, yet they may affect the quality of the models significantly. In this appendix, I introduce some practical (and sometimes empirical) model-building strategies that I learned through trial-and-error and extensive literature-survey/tutorial-reading during this dissertation research. Some of the strategies are generic for any type of machine learning models, and some are specific to the models used in this work. There is no one-size-fits-all solution for all machine learning problems. Thus, the audience should be mindful of the tips introduced in this appendix, since they may be beneficial only under certain conditions, and sometime, they may even hurt your model performance. This appendix first introduces the basics of building your hardware and software environment. Then, it describes some essential guidelines for speech data pre-processing. Lastly, it offers model training tips for the FAC systems introduced in this work.

C.1. Build a stable development environment

Maintaining a stable and reconfigurable development environment, which involves both the hardware and software, is crucial to generate reproducible results and makes it easier to iterate through research ideas quickly. Software and hardware platforms iterate rapidly. Therefore, the recommendations in this section might only apply to the services available at the time of this writing (June 16, 2020). Any future readers should also consult the most up-to-date documentation of their hardware and software packages.

C.1.1. Hardware platform

For most modern machine learning problems, we generally need high-end computer hardware to accelerate the computations. A common practice is to maintain a local development machine, which contains the minimum hardware to debug the code and run basic experiments, and a remote server that runs formal experiments. For example, in this dissertation work, I use a desktop computer that has an NVIDIA 1070 graphic computing unit (GPU) for development, and then upload the codes to an Amazon Web Service (AWS) cloud GPU instance to finish the model training.

The choice of the cloud GPU instance type greatly affects the model training speed. Some key parameters to look for in a GPU are the number of CUDA cores (for general floating-point computations), the number of Tensor cores (for reduced precision computations, e.g., half-precision), and the maximum available amount of graphic memory (RAM). The AWS P3 instances are equipped with the NVIDIA Tesla V100 GPUs, and they are suitable for training large models with large datasets. The P2 instances have NVIDIA K80 GPUs, which are built on a relatively old GPU architecture. The G4 instances contain NVIDIA T4 GPUs, which are geared towards training relatively small models and inference tasks because they have a relatively smaller number of CUDA cores compared with the V100 GPUs. However, since the T4 GPUs have Tensor cores and large RAM, if the code is optimized for half-precision training, the training speed can be fast. It is also preferable to perform parallel training if the code can be optimized to run on multiple GPUs simultaneously. Economically, the P3 instances are expensive (p3.2xlarge,

$3.06/h; one GPU/instance, same applies to the other examples), the P2 instances are outdated and relatively expensive (p2.xlarge, $0.9/h), while the G4 instances are more affordable (g4dn.xlarge, $0.526/h). Therefore, I use the G4 instances extensively in this work. There are multiple cloud computing services available on the market, for example, AWS, Microsoft Azure, and Google Cloud. Among these, AWS has the best customer support and the most mature ecosystem.

A factor many people often overlook when they perform machine learning training jobs is the choice of CPU. Although generally, we do not use CPUs directly for training tasks, they handle data processing and data transfer between the RAM and GPU. If the CPU is slow, then the bottleneck of computation becomes the CPU rather than the GPU. In addition, it is preferable to store data and other model training artifacts on a fast Solid State Drive (SSD) instead of a traditional mechanical hard drive to reduce I/O overhead. When possible, one should preload all data into the main memory or create a RAM disk for data I/O. If the physical RAM could not accommodate the whole data repository, one handy solution is to create a virtual memory space (e.g., a swap partition on Linux), which can generally double the available RAM without a significant I/O performance loss.

C.1.2 Software environment

Machine learning tasks often rely on existing third-party dependencies to extract features (e.g., `SPTK`, `librosa`), perform numeric computing (e.g., `numpy`, `scipy`), or modeling training (e.g., `TensorFlow`, `PyTorch`). It is crucial to maintain a manageable

and non-conflicting installation for all these dependencies. A good practice is to use package-managing toolchain like `conda`[32] or Python `venv`[33] to create a separate "sandbox" *workspace* for every new project. This also helps with the reproducibility of the project since other users can use the configurations of an existing *workspace* to duplicate the software environment. The AWS cloud computing instances often come with a pre-configured software environment managed by `conda`, and one can replicate the same configuration on their local development machines.

Another good practice is to use version control tools (e.g., `git`) to keep track of the project codes. This not only helps with versioning but also makes exploring many research ideas easier through branching. Well-maintained version history also provides a clear path for debugging and reduces implementation mistakes. More importantly, the majority of machine learning tasks involve fine-tuning a set of hyperparameters for the optimal performance, and a good version control system can keep records on what has been tested, and what remains to explore. Along this line, it is important to decouple model hyperparameters from the model implementation through good software engineering. A common practice is to keep all the parameters in a single configuration file and never hardcode the hyperparameters. When possible, it is recommended to use the protocol buffer syntax[34] (or other similar mechanisms) for the configuration file to allow flexible extensibility and backward compatibility.

---

[32] https://anaconda.org/
[33] https://docs.python.org/3/tutorial/venv.html
[34] https://developers.google.com/protocol-buffers

C.2. Speech data pre-processing

- Background noise filtering: Ideally, the speech data should be recorded in a quiet environment, and the speaker should speak with an appropriate volume. If the recordings contain consistent background noise, it can generally be removed through conventional signal processing methods like spectral subtraction before further processing. Popular recording software like Audacity also has built-in noise filtering functions.
- Utterance duration: For models that operate on the frame-level or without recurrent structure, the utterance duration does not matter much. For models that utilize recurrent structures like LSTM, the utterance duration should not be too long (for example, greater than 10 seconds). A good practice is to segment speech recordings into shorter sentences, either manually or through forced-alignment time boundaries.
- Silent segments in speech: Leading and trailing silent segments generally do not contain useful information and thus can generally be trimmed. Silent segments located within an utterance can be tagged by voice activity detection (VAD) or forced-alignment, and they may contain prosodic information. These utterance internal silence segments should be handled case-by-case. For example, in a voice conversion task, the model only needs to convert the speech segments and keep the silences unmodified. In speech recognition tasks, the prior distribution of the silent segments is an important model parameter.

- Sampling rate: Higher sampling rates provide better audio quality and frequency resolution. However, high sampling rates generally require more computing resources. In speech synthesis tasks, common sampling rates include 16 kHz, 22.05 kHz, and 44.1 kHz. Speech recognition tasks can use data with even lower sampling rates, e.g., 8 kHz telecommunication speech. It is important to keep the sampling rate consistent throughout the model building.
- Frame shift and window size: The frame shift size controls the frame rate, which directly affects the processing speed. Generally, we can use 1ms, 5ms, or 10ms. For speech synthesis tasks, a lower frame rate generally improves the synthesis quality. The common window size can be 25ms, 50ms, or 80ms. The exact value for the frame shift and window size may vary depending on the application. A worth-noting implementation issue is that different toolkits may process the last few frames in an utterance differently. Some tools (e.g., the default mode in `Kaldi`) would omit the last few frames that do not fit in a complete analysis window. Some other tools would keep the frames that are computed on incomplete analysis windows. Generally, these trailing frames are silent, and either omitting or retaining them would not lead to computational issues, but it is important to properly align the feature frames coming from different toolkits. I generally truncate the feature sequences from different data sources to the length of the shortest sequence, i.e., ignoring the last few silent frames, if any.

C.3. Model training

The models I introduce in this dissertation are all open-sourced online,

- Chapter 3: https://github.com/guanlongzhao/ppg-gmm
- Chapter 4: https://github.com/guanlongzhao/fac-via-ppg
- Chapter 5: https://github.com/guanlongzhao/reference-free-ac

However, when applying the existing codes to new corpora, the choice of hyperparameters and model stopping criteria can greatly affect the model performance. In this section, I introduce some tips and empirical rules that may help future applications.

C.3.1. Model in Chapter 3

This spectral conversion model introduced in Chapter 3 is a GMM, and we tested it with 128 mixtures and 96-dim dynamic spectral features. Generally, it should take less than 100 iterations for the GMM to converge during training, which takes less than half an hour, depending on the CPU specifications. I tested using 30, 40, 50, 100, and 1000 utterances for training the GMM. Using 1000 utterances for training produces the best syntheses, but the model can produce intelligible speech even with as little as five minutes of training data. Sometimes, the training might diverge. If that happens, I would recommend re-running the training multiple times.

C.3.2. Tacotron speech synthesizer in chapters 4 and 5

- Training data: One hour of speech from one speaker would be sufficient
- The number of neurons in each layer: Please use the values included in the open-source repositories as a reference starting point for customization. Smaller training corpora should use a fewer number of neurons and vice versa.

If the amount of data is sufficient, I generally prefer wider (more neurons per layer) over deeper (more layers) neural network structure, especially for recurrent layers like LSTMs, since deep LSTMs are unstable during training.

- Learning rate: A learning rate between 1×10^{-4} and 1×10^{-3} should be able to lead the model close to convergence. However, if the validation loss keeps fluctuating by a large margin during the training stage, it means that the learning rate is too large.
- Batch size: The general rule of thumb is to use the largest batch size that can fit in the GPU's memory. This value may vary by corpora since the largest batch size is determined by the longest sequence in the training data. Therefore, if there are a few abnormally long utterances in the training data, it might be beneficial to omit these utterances in exchange for larger batch sizes.
- Convergence: The model can be considered converged if the validation loss reaches a plateau. More specifically, I generally train the models for 30k-60k steps with a batch size of at least 6. The whole training process can take up to 24 hours on an AWS G4 single GPU instance.
- Speed up training: New models can be initialized with weights from a pre-trained model. Performing model adaptation instead of training from random initial weights can significantly reduce the iterations needed to reach convergence.

C.3.3. WaveGlow vocoder

- Training data: One hour of speech from one speaker would be sufficient.

- Neural network architecture: Please follow the default settings included in the code repository.
- Learning rate: A learning rate of 1×10^{-4} is sufficient. Changing the learning rate to a smaller value such as 1×10^{-5} after the validation loss stops changing may lead to better convergence.
- Batch size: At least 3 or 4.
- Convergence: The model can start to generate intelligible speech after being trained for 80k steps. Models trained for 200k-300k steps can generally produce high-quality sounds. The vocoder can be trained for up to 600k steps without seeing overfitting, although the improvements passing beyond 300k steps might be marginal. The total training time for 200k steps can be up to three days on an AWS G4 single GPU instance.
- Speed up training: The WaveGlow vocoder training process can also be accelerated by initializing from a pre-trained model.

C.3.4. Pronunciation correction models in Chapter 5

- Training data: One hour of speech from one speaker might be sufficient. More data that contains consistent mispronunciations can help the generalization of the model.
- Neural network architecture: Please use the default settings included in the code repository as a starting point for exploration.

- Learning rate: Please use the learning rate scheduler described in Section 5.5.3. You may need to modify the scheduler based on the number of training epochs you set.
- Batch size: At least 12-16.
- Convergence: For the *baseline* method (Zhang et al. [129]), it starts to converge at around 35k steps. For the *proposed* method (baseline + forward-and-backward decoding), it starts to converge at around 20k steps. The total training time can be up to one day on an AWS G4 single GPU instance. The proposed method trains approximately 2x slower than the baseline method due to the forward-and-backward decoding method. Besides the validation loss, the WER and the MCD between the converted speech and the ground-truth data also serve as useful indicators for model performance during the training process. These measurements should be computed on the validation set. WER is a proxy to the foreign accentedness of the syntheses, and MCD reflects the accuracy of the spectral conversion.

APPENDIX D

MAPPING BETWEEN ARPABET AND IPA SYMBOLS

Table D.1: The mapping between ARPABET and IPA symbols, with examples.

Index	*ARPABET*	*IPA*	*Example*	*Annotation*	*Type*
1	AA	ɑ	odd	AA D	vowel
2	AE	æ	at	AE T	vowel
3	AH	ʌ	hut	HH AH T	vowel
4	AO	ɔ	ought	AO T	vowel
5	AW	aʊ	cow	K AW	vowel
6	AX	ə	discus	D IH S K AX S	vowel
7	AY	aɪ	hide	HH AY D	vowel
8	B	b	be	B IY	stop
9	CH	tʃ	cheese	CH IY Z	affricate
10	D	d	dee	D IY	stop
11	DH	ð	thee	DH IY	fricative
12	EH	ɛ	Ed	EH D	vowel
13	ER	ɝ	hurt	HH ER T	vowel
14	EY	eɪ	ate	EY T	vowel
15	F	f	fee	F IY	fricative
16	G	g	green	G R IY N	stop
17	HH	h	he	HH IY	aspirate
18	IH	ɪ	it	IH T	vowel
19	IY	i	eat	IY T	vowel
20	JH	dʒ	gee	JH IY	affricate
21	K	k	key	K IY	stop
22	L	ɬ	lee	L IY	liquid
23	M	m	me	M IY	nasal
24	N	n	knee	N IY	nasal
25	NG	ŋ	ping	P IH NG	nasal
26	OW	oʊ	oat	OW T	vowel
27	OY	ɔɪ	toy	T OY	vowel
28	P	p	pee	P IY	stop

Table D.1: Continued.

Index	*ARPABET*	*IPA*	*Example*	*Annotation*	*Type*
29	R	ɹ	read	R IY D	liquid
30	S	s	sea	S IY	fricative
31	SH	ʃ	she	SH IY	fricative
32	T	t	tea	T IY	stop
33	TH	θ	theta	TH EY T AH	fricative
34	UH	ʊ	hood	HH UH D	vowel
35	UW	u	two	T UW	vowel
36	V	v	vee	V IY	fricative
37	W	w	we	W IY	semivowel
38	Y	j	yield	Y IY L D	semivowel
39	Z	z	zee	Z IY	fricative
40	ZH	ʒ	seizure	S IY ZH ER	fricative
41	SIL	N/A	N/A	SIL	silence

APPENDIX E

MODEL DETAILS OF THE SPEECH SYNTHESIZERS

The table below summarizes the neural network architectures of the three speech synthesizers. It is worth noting that the input PreNet produced a 512-dim summarization from the senone-PPG, which is higher than the dimensionality of the monophone-PPG and BNF. We did experiment on a lower dimensionality (256) in the input PreNet, which lead to significant artifacts and mispronunciations. Therefore, we used the current setting for the Senone-PPG system in order to generate intelligible speech syntheses to compare with the other two systems.

The models were trained using the Adam optimizer [133] with a constant learning rate of 1×10^{-4} until convergence, which was monitored by the validation loss. We applied a 1×10^{-6} weight decay [212] and a gradient clipping [213] of 1.0 during training. The batch size was set to 8 and the weight terms w_1 and w_2 in eq. (5.13) were set to 1.0 and 0.005, based on preliminary experiments [149].

Table E.1: Neural network architecture of the speech embedding to mel-spectrogram synthesizers.

Component	***Parameters***
Input-dim	6024 (Senone-PPG) / 346 (Mono-PPG) / 256 (BNF)
Input PreNet *Optional: Senone-PPG only*	Two fully connected (FC) layers, each has 512 ReLU units 0.5 dropout [134] rate Output-dim: 512
Convolutional layers	Three 1-D convolution layers (kernel size 5) Batch normalization [135] after each layer Output-dim: 512 (Senone-PPG) / 346 (Mono-PPG) / 256 (BNF)
Encoder	One-layer Bi-LSTM, 256 cells in each direction Output-dim: 512
Decoder PreNet	Two FC layers, each has 256 ReLU units 0.5 dropout rate Output-dim: 256
Attention LSTM	One-layer LSTM, 0.1 dropout rate Output-dim: 512
Attention layers	v in eq. (5.5) has 256 dims Eq. (5.6), $k = 32$, $r = 31$ Eq. (5.10), $w = 20$
Decoder LSTM	One-layer LSTM, 0.1 dropout rate Output-dim: 512
PostNet	Five 1-D convolution layers (kernel size 5) 0.5 dropout rate 512 channels in first four layers 80 channels in last layer Output-dim: 80

APPENDIX F

MODEL DETAILS OF THE PRONUNCIATION CORRECTION MODELS

The table below summarizes the model details of the baseline pronunciation correction model. On top of the baseline model, the proposed model adds a backward decoder that has the same structure (attention modules, decoder LSTM, and PreNet) as the baseline model's decoder. The phoneme prediction ground-truth labels were per-frame phoneme labels (with word positions) that were produced by force-aligning the audio to its orthographic transcriptions. We note that the phoneme predictions were only required in training, not testing. For both models, the training was performed with the Adam optimizer with a weight decay of 1×10^{-6} and a gradient clip of 1.0. The initial learning rate was 1×10^{-3} and was kept constant for the first 20 epochs, then exponentially decreased by a factor of 0.99 at each epoch for the next 280 epochs, and then kept constant at the terminal learning rate. The batch size was 16. The loss term weights w_1, w_2, w_3, and w_4 in equations (5.16) to (5.19) were empirically set to 1.0, 0.05, 0.5, and 100.0.

Table F.1: Neural network architecture of the baseline pronunciation correction model.

Component	***Parameters***
Input layer	80-dim mel-spectrum + 256-dim BNF
Encoder	Two-layer Pyramid Bi-LSTM 256 cells / direction / layer Frame sub-sampling rate: 2 With layer normalization [214] Output-dim: 512
Decoder PreNet	Two FC layers, each has 256 ReLU units 0.5 dropout rate Output-dim: 256
Attention mechanism	One-layer LSTM Forward-attention technique [158] for attention weights Output-dim: 512
Decoder LSTM	One-layer LSTM Output-dim: 512
PostNet	Five 1-D convolution layers (kernel size 5), 0.5 dropout rate 512 channels in first four layers and 80 channels in last layer Output-dim: 80
Input Phoneme Classifier	One FC layer + softmax Output-dim: 346
Output Phoneme Classifier	One FC layer + softmax Output-dim: 346