

MULTI-LINGUAL AND MULTI-SPEAKER NEURAL TEXT-TO-SPEECH SYSTEM

by

LIU, ZHAOYU

A Thesis Submitted to
The Hong Kong University of Science and Technology
in Partial Fulfillment of the Requirements for
the Degree of Master of Philosophy
in Computer Science and Engineering

March 2020, Hong Kong

Copyright © by LIU, Zhaoyu 2020

Authorization

I hereby declare that I am the sole author of the thesis.

I authorize the Hong Kong University of Science and Technology to lend this thesis to other institutions or individuals for the purpose of scholarly research.

I further authorize the Hong Kong University of Science and Technology to reproduce the thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

LIU, ZHAOYU

MULTI-LINGUAL AND MULTI-SPEAKER NEURAL TEXT-TO-SPEECH SYSTEM

by

LIU, ZHAOYU

This is to certify that I have examined the above M.Phil. thesis
and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by
the thesis examination committee have been made.

ASSOCIATE PROF. B. MAK, THESIS SUPERVISOR

PROF. D.Y. YEUNG, HEAD OF DEPARTMENT

Department of Computer Science and Engineering

16 March 2020

ACKNOWLEDGMENTS

First, I would like to express my gratitude to my supervisor, Brian Mak, for his guidance, encouragement and support on my research throughout my MPhil study at the Hong Kong University of Science and Technology. I also appreciate all the group members of the speech group to share their valuable insights on various interesting research topics. I am especially grateful to the PHD group member, Zhu, Yingke, who helped me a lot on implementing the speaker verification systems.

I would also like to thank for the internship opportunity offered by Logistics and Supply Chain MultiTech R&D Centre and the computing resources provided during the intern period.

I also appreciate the participants of the mean opinion score tests for their precious time and their valuable comments on the synthesized speeches.

Finally, I would like to thank my family - my parents, LIU, Hongxing and Yang, Shufeng, and my sister LIU, Yang for their love and support throughout my study and my life.

TABLE OF CONTENTS

Title Page	i
Authorization Page	ii
Signature Page	iii
Acknowledgments	iv
Table of Contents	v
List of Figures	viii
List of Tables	x
Abstract	xi
Chapter 1 Introduction	1
1.1 Text-to-speech System and Its Applications	1
1.2 Proposed Model and Its Contributions	4
Chapter 2 Literature Review	7
2.1 Neural TTS	7
2.2 Multi-speaker TTS	9
2.3 Multi-lingual TTS	11
Chapter 3 TTS Basics	15
3.1 Linguistic Features	15
3.1.1 Phoneme Embedding	15
3.1.2 Tone/stress Embedding	15
3.1.3 Accents	16

3.2 Acoustic Features	20
3.3 Speaker Embedding	20
3.3.1 Speaker Verification	20
3.3.2 Equal Error Rate	22
3.4 Language Embedding	23
3.5 Mean Opinion Score	23
Chapter 4 System Design	26
4.1 Overview	26
4.2 Speech Corpora	26
4.2.1 Librispeech	26
4.2.2 SurfingTech	27
4.2.3 Aishell	27
4.2.4 CUSENT	27
4.3 Linguistic and Acoustic Features	27
4.3.1 Denoising	28
4.3.2 Forced Alignment	28
4.3.3 Input Representation	29
4.3.4 Acoustic Features	29
4.4 Speaker Embeddings - x-vectors	30
4.5 Neural Mel-spectrogram Synthesizer	32
4.5.1 The Encoder	33
4.5.2 Attention Mechanism	34
4.5.3 The Decoder	36
4.6 WaveNet Vocoder	37
Chapter 5 Preliminary Experiments and Results	40
5.1 Overview	40
5.2 Input Representation: Character/Phoneme	41
5.3 X-vector Dimension	42
5.4 IPA vs ARPABET	43
5.5 Speaker Embedding Normalization	43

Chapter 6 Evaluation & Results	49
6.1 Baseline	49
6.2 Proposed	49
6.3 Experiments and Results	49
6.4 Discussions	50
6.4.1 Intelligibility and Naturalness	50
6.4.2 Speaker Similarity	53
Chapter 7 Conclusions	55
References	56
Appendix A Phoneme Mappings	61
A.1 Pinyin to ARPABET	61
A.2 Jyupting to ARPABET	62
A.3 ARPABET Phoneme set	63

LIST OF FIGURES

1.1	A typical traditional TTS system architecture.	2
1.2	Multi-lingual multi-speaker TTS system using speaker embedding, language embedding and tone/stress embedding.	4
2.1	The TTS system: Deep Voice 1.	7
2.2	The TTS system: Tacotron 2.	8
2.3	The DNN based multi-speaker TTS system.	9
2.4	The TTS system: Multi-speaker Tacotron 2.	10
2.5	Deep Voice 2: Speaker embeddings in the segmentation model.	11
2.6	Deep Voice 2: Speaker embeddings in the duration model.	12
2.7	Deep Voice 2: Speaker embeddings in the fundamental frequency (F0) model.	13
2.8	Multi-lingual Tacotron 2.	14
3.1	Linguistic feature inputs for synthesizing an English utterance in native English, Mandarin speakers' native/accented English and Cantonese speakers' native/accented English.	17
3.2	Linguistic feature inputs for synthesizing a Cantonese utterance in native Cantonese, English speakers' native/accented Cantonese and Mandarin speakers' native/accented Cantonese.	18
3.3	Linguistic feature inputs for synthesizing a Mandarin utterance in native Mandarin, Cantonese speakers' native/accented Mandarin and English speakers' native/accented Mandarin.	19
3.4	The system flow of the GMM-UBM based conventional speaker verification system.	20
3.5	Question page for collecting language fluency information.	24
3.6	Question page for testing intelligibility and naturalness.	25
3.7	Question page for testing speaker similarity.	25
4.1	An example of forced alignment between the audio waveform and word/phoneme-level transcriptions.	28
4.2	The deep neural network architecture of the x-vector system.	31
4.3	The architecture of Tacotron 2 including the encoder and decoder of the synthesizer and the WaveNet.	32

4.4	A general frame work of attention mechanism.	34
4.5	The dilated causal convolutions layers in WaveNet.	37
4.6	Residual block with gated activation units in WaveNet.	38
5.1	t-SNE visualization of x-vectors.	45
5.2	t-SNE visualization of x-vectors after L2-norm normalization.	46
5.3	t-SNE visualization of x-vectors after whitening.	47

LIST OF TABLES

3.1	The index of one-hot tone/stress embedding and the tone or stress it represents in Cantonese, English and Mandarin.	16
3.2	Absolute category rating scale for MOS test in intelligibility, naturalness and speaker similarity.	24
4.1	The detailed configurations of the x-vector network where the x-vector dimension is denoted as n-D, F denotes the dimension of filterbanks features of one frame, T denotes the total number of frames of the input utterance, and N denotes the number of training speakers.	30
5.1	Librispeech SV EER (%) for x-vectors in different dimensions.	42
5.2	Effect of increasing number of training speakers on Librispeech SV EER (%) using 128-D x-vector.	42
5.3	Notations used in MOS results.	44
5.4	Intelligibility MOS (mean \pm standard deviation) on unseen speakers.	48
5.5	Naturalness MOS (mean \pm standard deviation) on unseen speakers.	48
5.6	Speaker similarity MOS (mean \pm standard deviation) on unseen speakers.	48
6.1	The MOS mean and 95% confidence interval using t-distribution for ground truth utterances in three languages.	51
6.2	The MOS mean and 95% confidence interval using t-distribution for English synthesized utterances for seen/unseen speakers using the mono-lingual baseline English model.	51
6.3	MOS mean and 95% confidence interval using t-distribution for synthesized utterances in 5 accents in three languages using the proposed multi-lingual model for unseen speakers.	51
A.1	The mapping table which maps the pinyin phonemes to the ARPABET phonemes where ‘j’, ‘q’ and ‘x’ are mapped to separate phonemes ‘J_M’, ‘Q_M’ and ‘X_M’ which are then concatenated to the ARPABET phoneme set.	61
A.2	The mapping table which maps the Jyupting phonemes to the ARPABET phonemes. 62	
A.3	ARPABET phoneme set including 39 phonemes used by the CMU pronouncing dictionary.	63

MULTI-LINGUAL AND MULTI-SPEAKER NEURAL TEXT-TO-SPEECH SYSTEM

by

LIU, ZHAOYU

Department of Computer Science and Engineering

The Hong Kong University of Science and Technology

ABSTRACT

Recent studies in multi-lingual multi-speaker text-to-speech (TTS) systems proposed models that can synthesize high-quality speeches. However, some models are trained with proprietary corpora consisting of hours of speeches recorded by performing artists and require additional fine-tuning to enroll new voices. To reduce the cost of training corpora and support online enrollment of new voices, we investigate a novel multi-lingual multi-speaker neural TTS synthesis approach for generating high-quality native or accented speech for native/foreign seen/unseen speakers in English, Mandarin and Cantonese. The unique features of the proposed model make it possible to synthesize accented/fluent speeches for a speaker in a language that is not his/her mother tongue. Our proposed model extends the single speaker Tacotron-based TTS model by transfer learning technique which conditions the model on pretrained speaker embeddings, x-vectors, using a speaker verification system. We also replace the input character embedding with a concatenation of phoneme embedding and tone/stress embedding to produce more natural speech. The additional tone/stress embedding works as an extension of language embedding which provides extra controls

on accents over the languages. By manipulating the tone/stress input, our model can synthesize native or accented speech for foreign speakers. The WaveNet vocoder in the TTS model is trained with Cantonese speech and yet it can synthesize English and Mandarin speech very well. It demonstrates that conditioning the WaveNet on mel-spectrograms is good enough for it to perform well in multi-lingual speech synthesis. The mean opinion score (MOS) results show that the synthesized native speech of both unseen foreign and native speakers are intelligent and natural. The speaker similarity of such speech is also good. The lower scores of foreign accented speech suggests that it is distinguishable from native speech. The foreign accents we introduced can confuse the meaning of the synthesized speech perceived by human raters.

CHAPTER 1

INTRODUCTION

1.1 Text-to-speech System and Its Applications

The text to speech (TTS) system is a technology converting text to human speech. The conversion is also known as speech synthesis which is a computer-based process to generate speech. These systems are primarily designed to assist people with visual impairments to read various materials such as books, newspaper, magazines, etc. Nowadays TTS systems are widely applied in many areas not only to aid the disabled to read but also for study or entertainment by transforming read books to audio ones. TTS systems are embedded in many computer operating systems and online TTS services are also widely available with different qualities and for different purposes.

TTS systems can be evaluated in various aspects such as synthesis speed, synthesized audio quality, emotions, number of speaker voices, number of languages, new voice enrollment, costs, etc. Some TTS systems outperform the others in different aspects and are therefore used in different applications. TTS reader is one of the applications of TTS systems to read texts. It requires the TTS system to synthesize audios at real-time speed, i.e., the same speed at which the audio is played. It also requires the synthesized audio to be highly intelligible. The high audio naturalness is also required but it is less important than the intelligibility. Many TTS readers are built with traditional TTS systems. The cost could be expensive when using the concatenative-based TTS systems because it requires a large speech database of speech segments of the same speaker. However, the audio quality of the concatenative-based TTS systems is usually better than that of other approaches such as formant-based and parametric-based TTS systems. Emotions, supporting multiple speaker voices and new voice enrollments are nice additional features. In TTS readers, supporting multiple languages is usually achieved by building separate systems in every language. The costs of such TTS readers increase linearly with additional languages. Another application of TTS systems is the chatbot. A chatbot is a software to conduct conversations with human beings.

TTS systems can be applied in chatbots to interact with humans over speech instead of texts. The requirements for TTS systems used in chat bots are similar to those for TTS readers except that the emotions can become more important in conversational speeches. Moreover, TTS systems can be also applied in film dubbing. Film dubbing is more critical on synthesized audio quality and emotions. Real-time synthesis is usually not necessary. Even though it may be less critical on the costs and the other features such as multiple speaker voices and languages, it is always desired to reduce the costs by improving TTS systems.

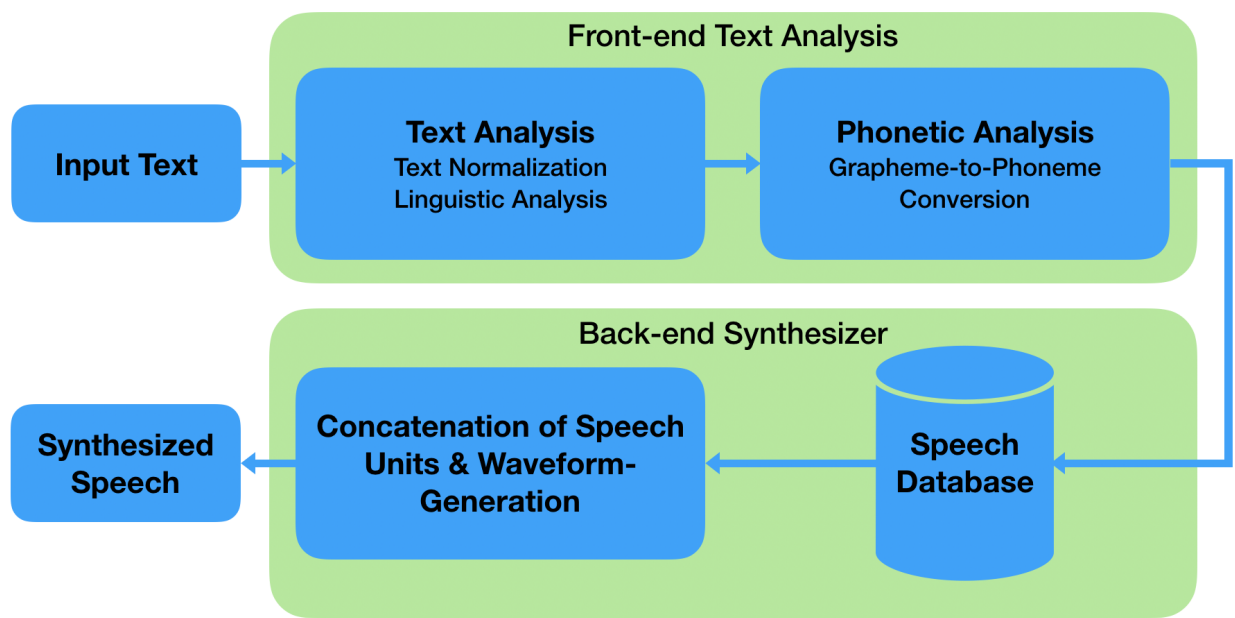


Figure 1.1: A typical traditional TTS system architecture.

A typical traditional TTS system as depicted in Figure 1.1 consists of a front-end text analyzer and a back-end synthesizer. Text analysis is necessary to prepare phoneme-level linguistic information for the synthesizer to synthesize the speech waveform. Text is normalized where numbers and abbreviations, which are ambiguous in pronunciations, are converted to orthographic forms. They are further converted to phonemes. Their locations are determined by an extra segmentation model and their durations are predicted by phoneme duration model. Fundamental frequency (F0) is also computed with the fundamental frequency estimator. All these features are input to the back-end to synthesize speech waveform. Sometimes the back-end may be further separated to two components as shown in Figure 1.2, a synthesizer and a vocoder, where the synthesizer converts

linguistic features to acoustic features and the vocoder convert acoustic features to speech waveform. Normally, acoustic features are intermediate features which can be directly computed from audio waveform by signal processing techniques to bridge the synthesizer and the vocoder. Some of the most popular conventional back-end systems includes concatenative unit-selection synthesis, parametric synthesis and formant synthesis. Despite the fact that traditional methods can be successful in producing speeches, recently developed neural synthesis techniques can outperform the traditional ones in various aspects.

In traditional TTS synthesis methods, system components such as the grapheme-to-phoneme model, phoneme duration model, segmentation model, fundamental frequency estimation model and synthesis model are trained separately. They require expert domain knowledge to produce high-quality synthesized speech. With the advance of deep learning, they are replaced by neural models. Recent studies [2, 33, 24, 26] propose integrated neural networks that simplify the training process. [24] also proves the neural synthesis can generate high-quality and natural speeches close to human's. Such neural-based TTS systems can greatly reduce the costs. These recent studies adopt a system architecture consisting of an end-to-end front-end model and a vocoder. This kind of architecture integrates multiple separate system components of traditional TTS systems. It reduces the amount of feature engineering and shortens the time for training new engineers for the maintenance. The audio quality of synthesized audios is also improved.

Extending TTS systems to support multiple voices can help to make them more adaptive to various applications. Another motivation of multi-speaker TTS systems is to reduce the amount of training data required per speaker since recording hours of speech of one speaker has been a big barrier in training TTS systems. Several recent works proved that a TTS system can be trained with tens of minutes of speech per speaker from many speakers. Multi-speaker TTS systems can synthesize audios in multiple speakers' voices by training a single model. It avoids building multiple separate single-speaker TTS systems to achieve the same purpose so as to reduce the cost. The cost of building training corpora can be reduced as well. Some TTS systems support new speaker enrollments where a new speaker's voice can be added to the model using a few minutes of their recordings. The synthesized audio quality of some multi-speaker TTS systems is lower than single-speaker TTS systems but is still acceptable.

Multi-lingual TTS systems extends previous TTS systems to support multiple languages. The fantastic feature of such systems is cross-lingual voice cloning which can synthesize target speakers' speeches in more than one languages not limited to their native language. New speakers' voices can be registered and cloned with their speech and the system can synthesize new speeches from them in all supporting languages. Voice cloning can be a unique feature for film dubbing to synthesize foreign speeches apart from the native language of the actors. Some multi-lingual TTS systems can also synthesize foreign speeches with various accents and fluency to distinguish between foreigners and secondary language learners.

1.2 Proposed Model and Its Contributions

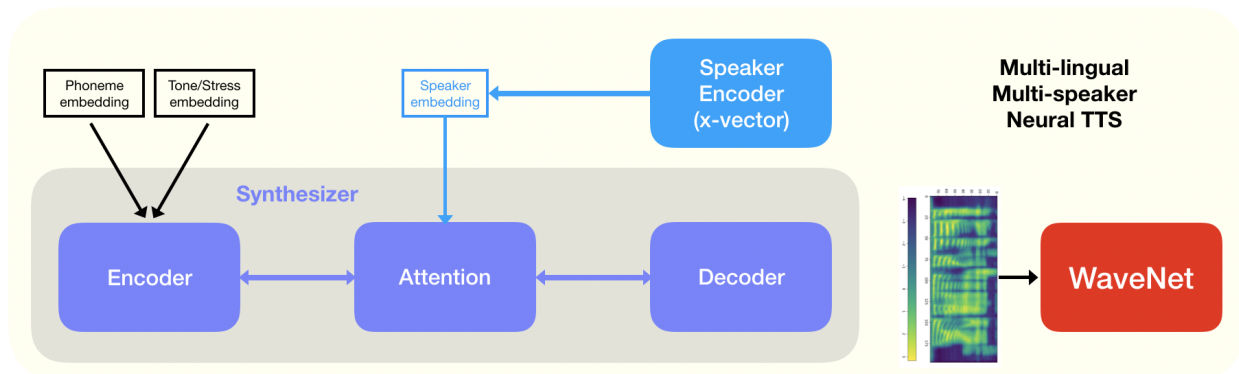


Figure 1.2: Multi-lingual multi-speaker TTS system using speaker embedding, language embedding and tone/stress embedding.

In this thesis, we propose a novel neural based multi-lingual multi-speaker TTS system architecture as shown in Figure 1.2. We demonstrate that our proposed system can generate both foreigner's native/accented speech and native speech for seen or unseen speakers without parallel corpuses. The system can synthesize native speeches in target speaker's voices which sound like native speakers in their mother language. It can also synthesize foreign native speeches which sound like foreigners speaking their secondary language fluently. It can also synthesize foreign accented speech which sounds like foreigners speaking foreign languages with the accent of their mother language. The proposed system can synthesize those speeches with different inputs. It is shown from mean opinion score (MOS) results that both the synthesized native and foreign native

speeches are high intelligible, natural but distinguishable from each other. The synthesized native speech is more intelligible and natural than synthesized foreign native speech. The proposed system also alleviates the need of large amount of speech data for new speakers enrollment and no additional training is necessary. With 2-3 minutes of speech from a new speaker, our model is able to extract the x-vector and synthesize high-quality speech in the target speaker's voice. Furthermore, the model can be trained with a simple mixture of multiple corpuses from different languages instead of parallel corpuses which require each utterance in one language having its translations in another language. The proposed model can synthesize high-quality audios in different voices and languages with a single model. Comparing with traditional TTS systems, the proposed model reduces the cost of building multiple models to support multiple voices and languages. Comparing with other existing multi-lingual multi-speaker models, the unique feature of the proposed model is that it can synthesize those high-quality speeches with various accents in a different language from the new speaker's native language in which the new data is recorded. One scenario to apply the proposed model is to synthesize fluent/accented foreign speeches of film actors, e.g., fluent/accented English speeches for Jackie Chan, who enrolls with only Cantonese utterances (and Cantonese is his mother tongue). Furthermore, the English and Mandarin corpora used to train the proposed model are open-source and it means the proposed model is reproducible. The corpora used to train the proposed model contain less than 30 minutes of data per speaker, so it is less costly compared with corpora with hours of speeches per speaker from performing artists.

The remaining of the thesis are organized as follows. Chapter 2 reviews the literature of the neural-based TTS systems including single speaker TTS, multi-speaker TTS and multi-lingual TTS. Chapter 3 introduces the important concepts of the proposed TTS system including the input linguistic and acoustic features, speaker embeddings and language embedding. It also introduce the concept of mean opinion score (MOS) as the subjective evaluation method of TTS systems. Chapter 4 describes the system design and the architecture including training corpuses, data preprocessing and the implementation of the components of the proposed system. Chapter 5 demonstrates some of the preliminary experiments and the results. Those experiments find the optimal of various substantial factors that affect the performance of proposed system. Chapter 6 discusses the experiment results of the baseline and proposed system in three aspects, intelligibility, naturalness and speaker

similarity. Finally, chapter 7 concludes the thesis.

CHAPTER 2

LITERATURE REVIEW

2.1 Neural TTS

There are several recent studies on neural TTS system where their results have demonstrated the huge potential of neural speech synthesis techniques.

Deep Voice [2] presents a neural TTS system which replaces each separate system component with a neural network-based model. The architecture is shown in Figure 2.1. The system can synthesize intelligible speech in real time or much faster than real time. However, according to their study, the faster synthesis speed may be achieved at a cost of speech quality.

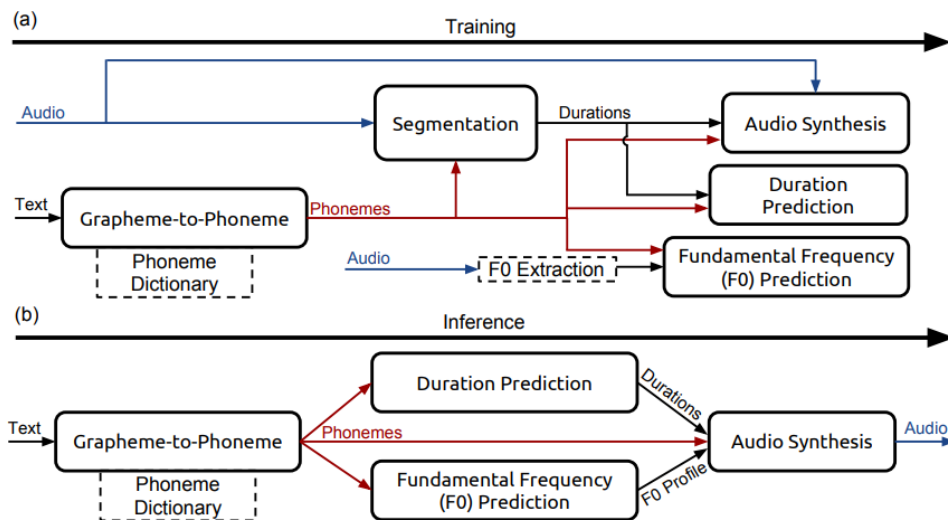


Figure 2.1: The TTS system: Deep Voice 1.

In contrast, Char2wav [26] and Tacotron [33] and its improved version Tacotron2 [24] resort to a totally end-to-end neural model¹ that uses an attention mechanism to convert a sequence of

¹Actually “end-to-end” here only means that both Char2Wav and Tacotron generate vocoder features, not speech audios, from some representation of input texts.

text directly to its corresponding sequence of acoustic features, from which speech audios may be generated using a vocoder. Char2Wav generates WORLD features [19] and uses SampleRNN [18] to generate speech, while Tacotron/Tacotron2 generates linear/mel spectrograms and uses the Griffin-Lim (GL) [11] and WaveNet [30] vocoder, respectively. Tacotron 2 can synthesize natural speech comparable to genuine human speech. These works prove that the neural network can be successfully applied to speech synthesis and it greatly simplifies the overall system architecture and training process.

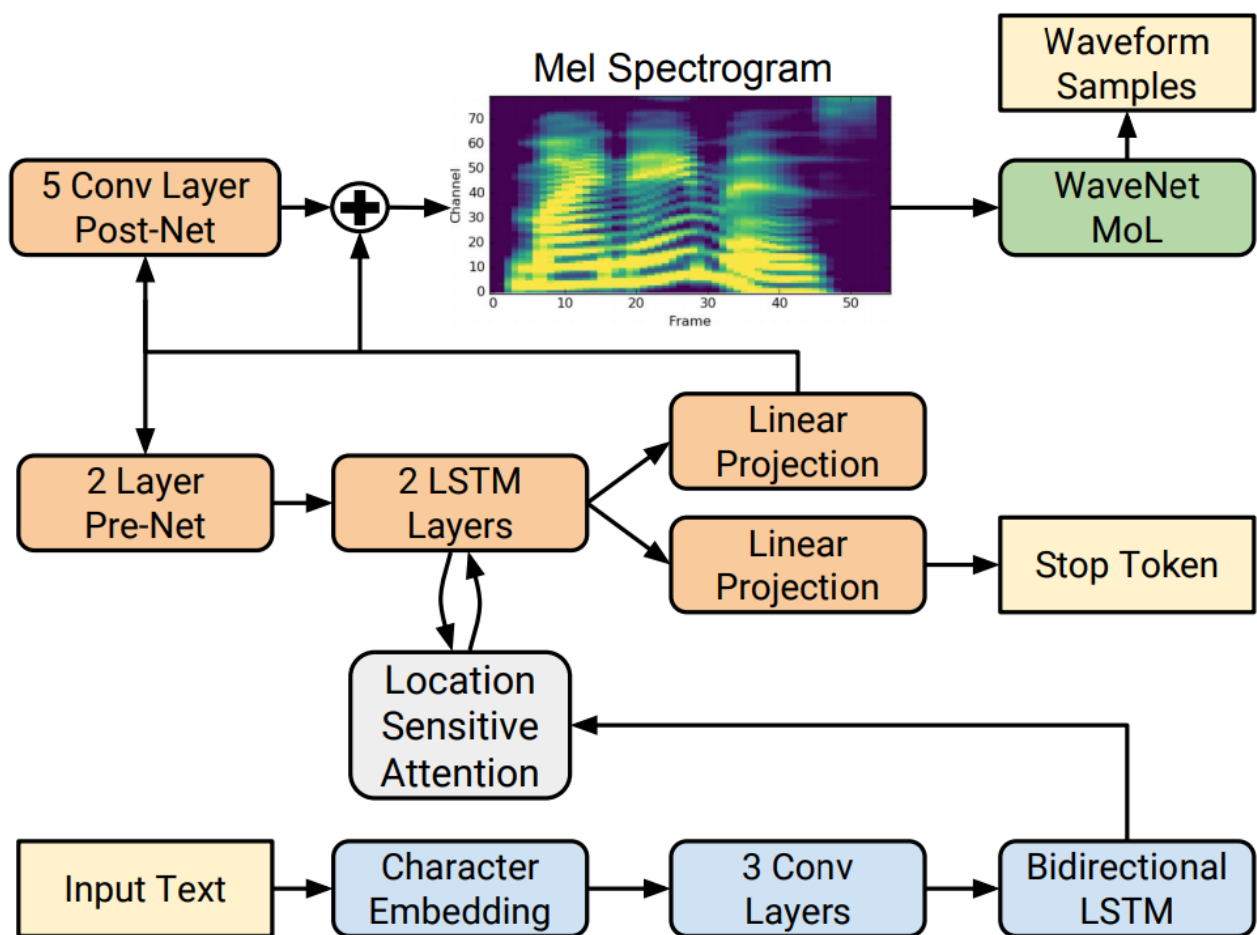


Figure 2.2: The TTS system: Tacotron 2.

2.2 Multi-speaker TTS

Single-speaker neural TTS systems can be readily extended to support multiple speakers' voices. There are several studies that present novel approaches to English Multi-speaker TTS system. Most of them introduce extra speaker embeddings to represent the identity of the training speakers. Different works integrate with speaker embedding in different ways. Some of them transfer the knowledge of pretrained speaker embeddings from other independent systems trained with a large number of speakers. Some other works introduce trainable speaker embedding to the TTS system which are jointly trained with the system parameters.

As in Figure 2.3, [8] takes the multi-task learning approach and duplicates the output layer for each of its training speakers so that each speaker is trained with its own speaker-dependent output layer while sharing other hidden layers in the model. Obviously, the model parameters in its output layer grow linearly with the number of training speakers and it will encounter problems dealing with a large number of speakers.

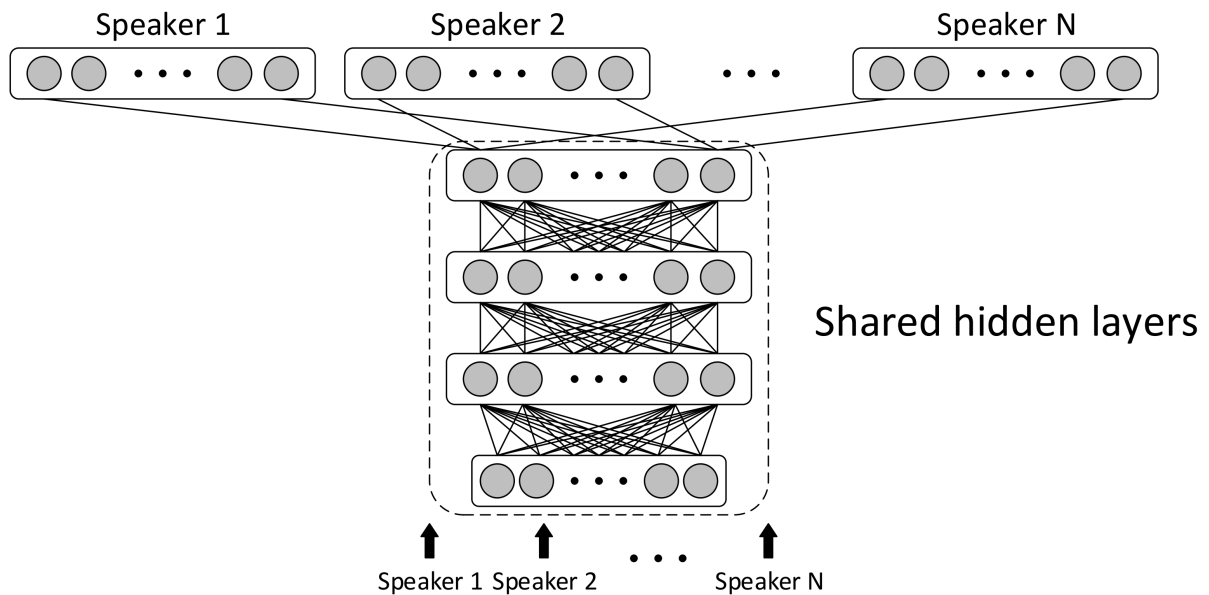


Figure 2.3: The DNN based multi-speaker TTS system.

Multi-speaker Tacotron, as in Figure 2.4 [13] is introduced by conditioning Tacotron 2's model on pre-trained d-vectors so that new speakers can be enrolled with a few seconds of speech. It uses

transfer learning technique to transfer the knowledge of pretrained speaker embeddings extracted from speaker verification systems to speech synthesis. The results shows the overall system can synthesize intelligent and natural speech in English. However, the results also shows the large number of speakers (over 18k speakers) is an important factor to train a successful system.

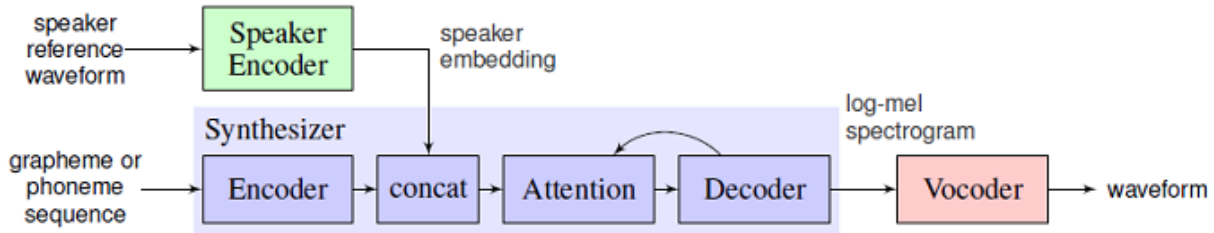


Figure 2.4: The TTS system: Multi-speaker Tacotron 2.

Similarly, Deep Voice 2 [10] as shown in Figures 2.5, 2.6 and 2.7 and Deep Voice 3 [21] extend Deep Voice to multi-speaker TTS. Unlike Tacotron 2, Deep Voice 2 and 3 condition each layer of the model with speaker embeddings which is jointly trained with the rest of the TTS system. For example, Deep Voice 3 claims to support 2400 voices. However, enrollment of new speakers in [10] and [21] will require additional training. VoiceLoop [29] uses a fixed-size memory buffer to accommodate speaker-dependent phonological information and facilitates multi-speaker synthesis by buffer shifts. New speaker embeddings can be trained by an optimization procedure while fixing the other model parameters. Neural Voice cloning [1] introduces a similar speaker adaptation method where both model parameters and speaker embeddings are fine-tuned with data from the new speaker.

Using trainable speaker embeddings or transferring the knowledge of pretrained speaker embeddings have both advantages and disadvantages. Trainable speaker embedding requires further training to synthesize speech for unseen speakers. However, it is more data efficient without requiring a large number of speakers. On the contrary, transfer learning separates the training of TTS system and speaker embeddings. New speaker enrollment is much easier with a pretrained speaker verification system.

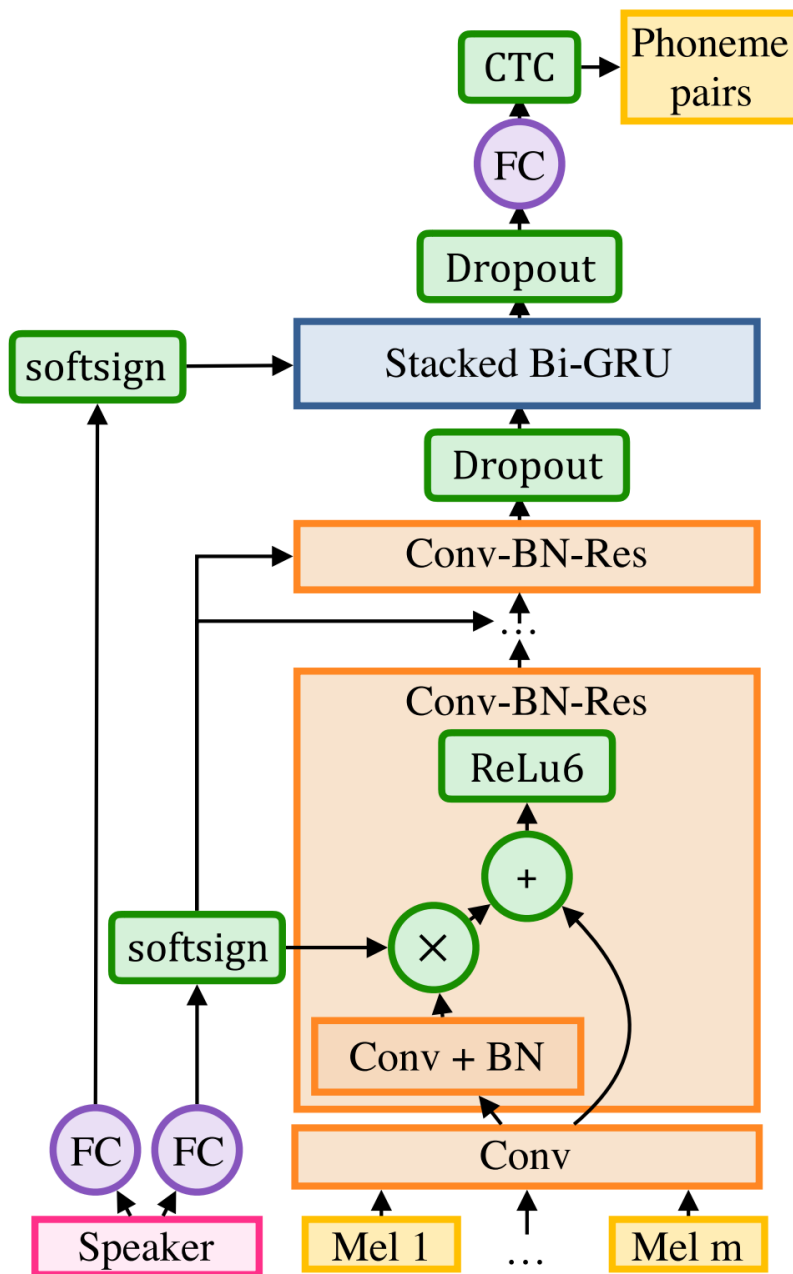


Figure 2.5: Deep Voice 2: Speaker embeddings in the segmentation model.

2.3 Multi-lingual TTS

Multi-lingual TTS further extends multi-speaker TTS to support synthesis in more than one language. For example, [6] introduces a cross-lingual TTS system in English and Mandarin trained

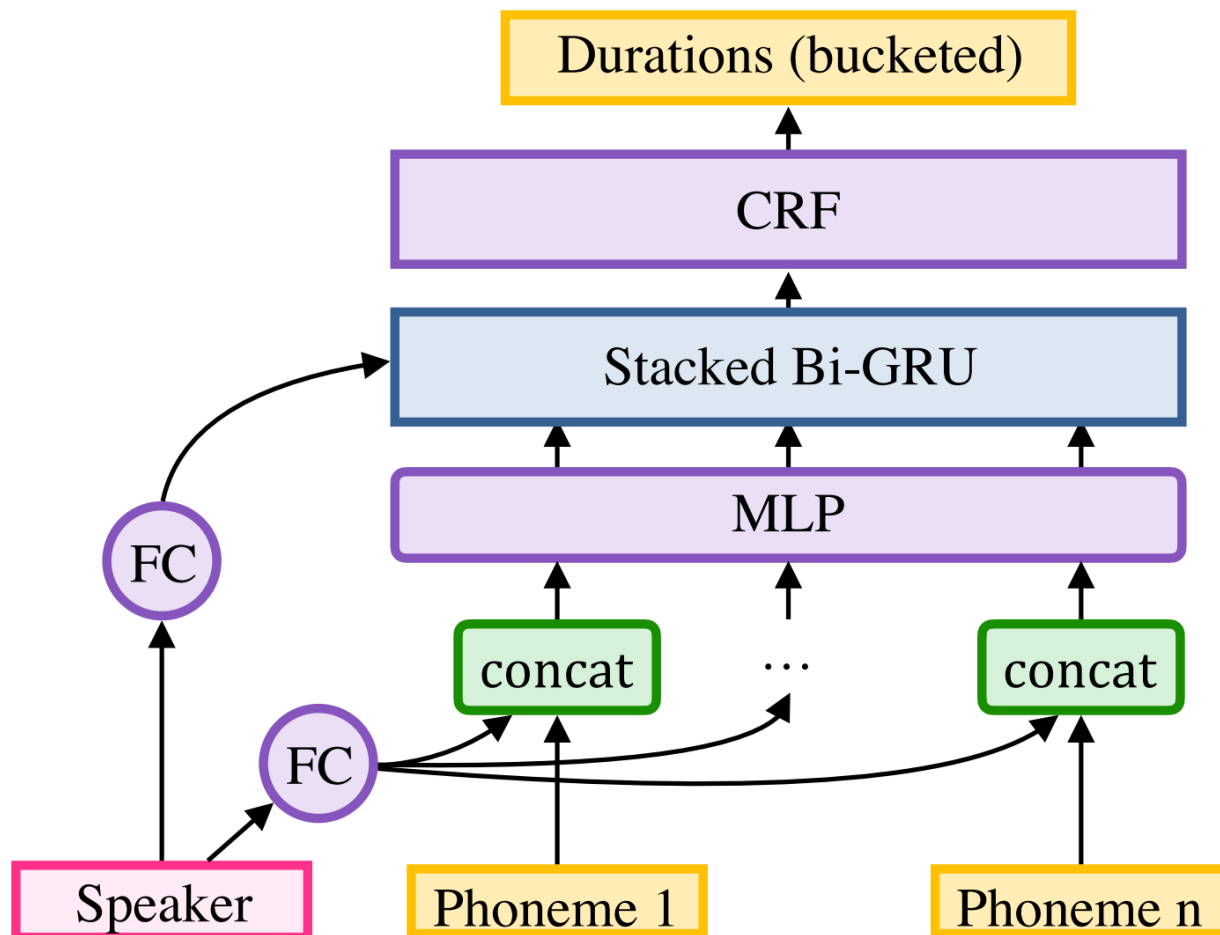


Figure 2.6: Deep Voice 2: Speaker embeddings in the duration model.

with their phoneme inputs in IPA representation without language embedding. It succeeds in synthesizing speech in two languages, however, it can only synthesize native speech but not accented speech. It uses the Griffin & Lim [11] vocoder (instead of WaveNet or other neural-based high fidelity vocoders) resulting in synthesized speech of lower quality.

[35] presents a TTS system as shown in Figure 2.8. Even though it shares many ideas in our system, there are the following notable differences:

- Most importantly, results in this thesis are reproducible as we use publicly available training corpora: English from LibriSpeech and Mandarin from SurfingTech, Aishell and Cantonese

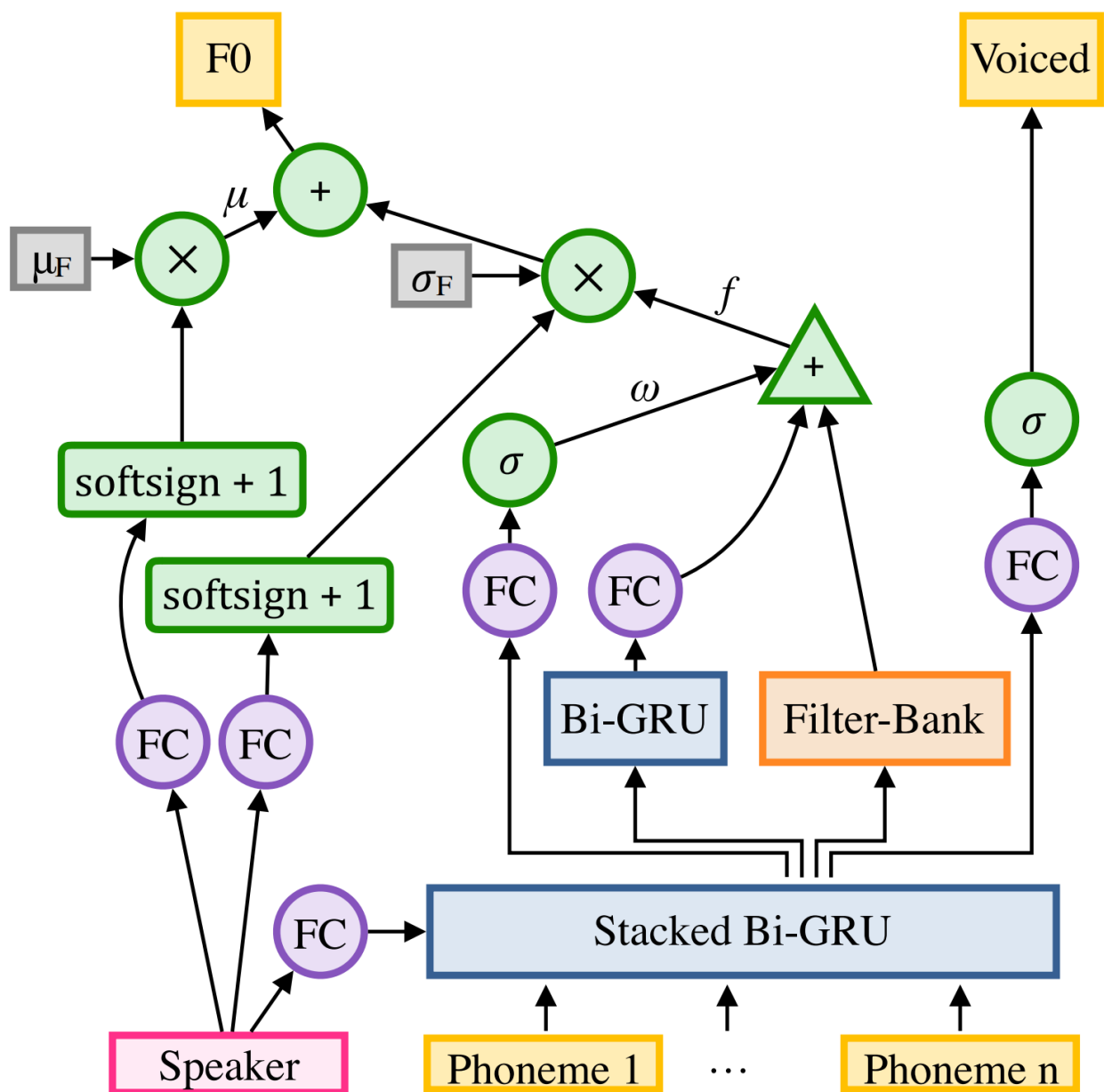


Figure 2.7: Deep Voice 2: Speaker embeddings in the fundamental frequency (F0) model.

from CUSENT, while the system in [35] is trained on proprietary data.

- [35] aims at synthesizing speech with a few training speakers' voices; thus, their training data consists of few speakers (some are professional voice actors) but each has tens of hours

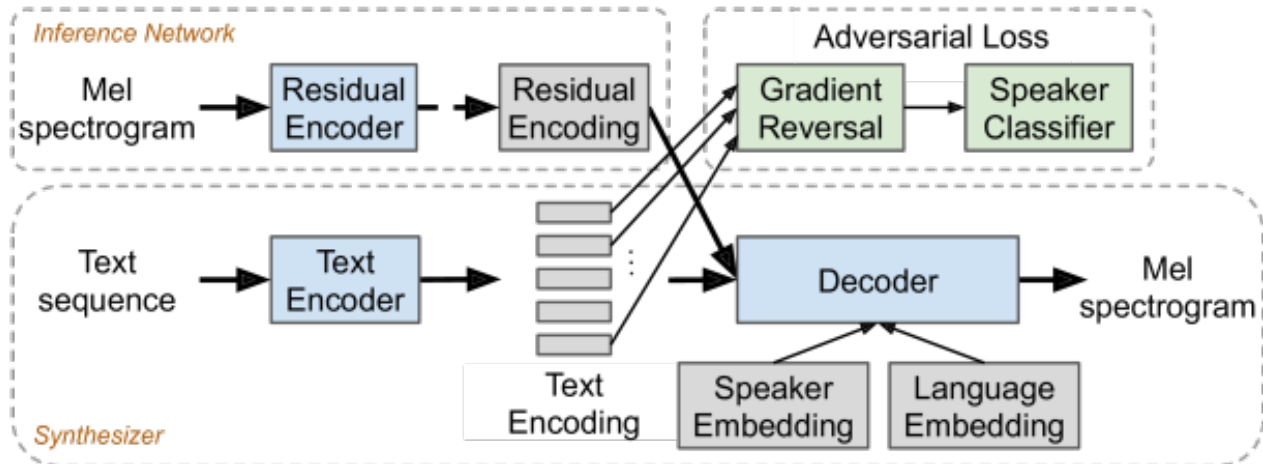


Figure 2.8: Multi-lingual Tacotron 2.

of speech. On the contrary, we train our system on hundreds of speakers with less than 25 minutes of speech from each speaker. We believe our system is more generalizable to new speakers and we report results on unseen speakers while [35] does not.

- Both our system and theirs employ shared phonemes for inputs and speaker embeddings and stress and tone embeddings. However, we use the state-of-the-art x-vector for speaker embedding while theirs is d-vector. We expect our synthesized speech will be better in terms of speaker similarity, especially for unseen test speaker.
- Our model is simpler with no residual encoding nor adversarial training. Instead, we investigate on the effect of various normalization methods on the speaker embedding vectors for enhancing the intelligibility, naturalness and speaker similarity of the synthesized speech.
- We also investigate the effect of training the WaveNet vocoder with one language (Cantonese) only to synthesize speech of the intended languages (Cantonese, English and Mandarin) in the system.

CHAPTER 3

TTS BASICS

3.1 Linguistic Features

The TTS system accepts linguistic features as input. In traditional approaches, linguistic features are at phoneme-level which needs text analysis preprocessing. In some mono-lingual systems, character embeddings are used instead of phoneme embeddings to simplify the procedure. However, it is reported by several works that systems [13, 35] working with phoneme embeddings outperform character embedding significantly.

3.1.1 Phoneme Embedding

Phoneme embedding is an vector representing phonemes in languages. All possible phonemes in one target language form a phoneme set. Each phoneme is represented by an embedding jointly trained with the system. For example, a popular English phoneme set is ARPABET. In English TTS systems, we use the ARPABET phoneme set with 39 phonemes.

3.1.2 Tone/stress Embedding

Tone/stress embedding in this thesis refers to the one-hot embedding representing tones in tonal languages and stresses in English. There are 5 tones in Mandarin, no tone and tone one, two, three and four. Similarly, there are six tones in Cantonese. English is a non-tonal language with three stresses, primary stress, secondary stress and no stress.

There are two methods to input tone and stress information to a TTS system. In mono-lingual systems, tone/stress and phonemes are not separated. For example, in English TTS systems, there are ‘AA1’, ‘AA2’ and ‘AA0’ phonemes in the phoneme set for phoneme ‘AA’ with primary,

secondary and no stress, respectively. However, in multi-lingual systems, we only include one phoneme ‘AA’ in the phoneme set and the stress information is separately stored in the tone/stress embedding to reduce the size of the phoneme set.

Index	Tone/stress
0	Mandarin: Neutral tone
1	Mandarin: Tone one
2	Mandarin: Tone two
3	Mandarin: Tone three
4	Mandarin: Tone four
5	English: No stress
6	English: Primary stress
7	English: Secondary stress
8	Cantonese: High level (Tone one)
9	Cantonese: Mid rising (Tone two)
10	Cantonese: Mid level (Tone three)
11	Cantonese: Low falling (Tone four)
12	Cantonese: Low rising (Tone five)
13	Cantonese: Low level (Tone six)

Table 3.1: The index of one-hot tone/stress embedding and the tone or stress it represents in Cantonese, English and Mandarin.

3.1.3 Accents

We synthesize speeches in five accents in each language. For example, for language A, we synthesize (1) the native speech in language A spoken by its native speakers; (2-3) the foreign native speech spoken by native speakers of the other languages B and C; (4-5) the foreign accented speech spoken by native speakers of the other languages B and C. We would like to demonstrate the differences between native speeches spoken by speakers in three different languages as well as the differences between foreign accented and foreign native speeches to imitate the different levels of fluency (foreign to fluent) they have in a second language. This is achieved by manipulating the tone/stress inputs among different cases as shown in Figure 3.1, 3.2 and 3.3.

In each language, our proposed system will first convert the native phoneme representation to ARPABET phonemes without stresses. To synthesize different accents for an utterance, we

input the same phoneme inputs but different tone/stress inputs. To synthesize speeches with native accents (1), (2) and (3) in the target language, we input the natural tone/stress in the target language. We input a constant tone/stress representation from the native language of the target speaker to synthesize speeches with foreign accented accents. For example, Mandarin speakers utter their accented Cantonese and English speech with the tone one in Mandarin.

Transcript: Through out the centuries people has explained the rainbow in various ways.

Mono-lingual System

phoneme input

sil TH R UW0 AW1 T ... ~

Native English

Multi-lingual System

phoneme input

sil TH R UW AW T ... ~

tone/stress input

0 5 5 5 6 5 ... 0

Native English

0 5 5 5 6 5 ... 0

Native English spoken by Mandarin speaker

0 1 1 1 1 1 ... 0

Accented English spoken by Mandarin speaker

0 5 5 5 6 5 ... 0

Native English spoken by Cantonese speaker

0 8 8 8 8 8 ... 0

Accented English spoken by Cantonese speaker

Figure 3.1: Linguistic feature inputs for synthesizing an English utterance in native English, Mandarin speakers' native/accented English and Cantonese speakers' native/accented English.

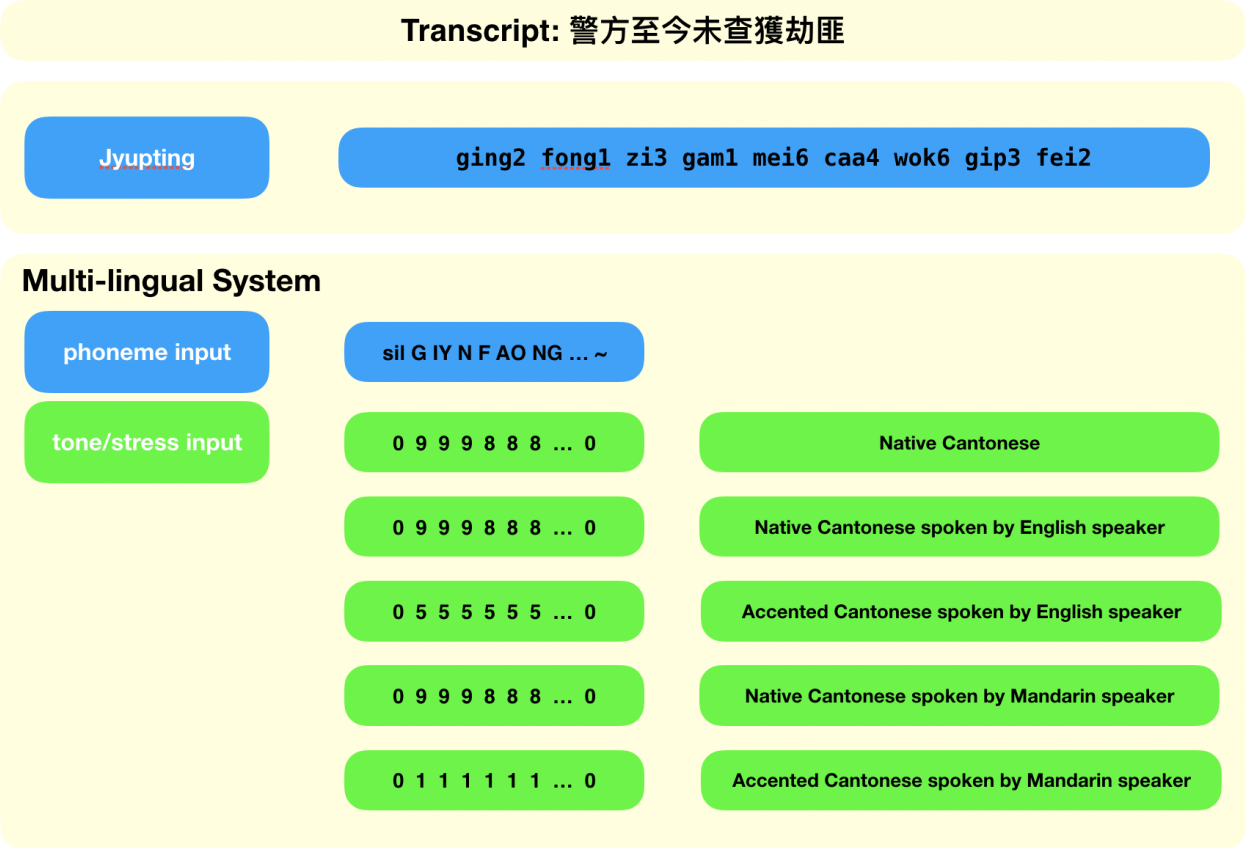


Figure 3.2: Linguistic feature inputs for synthesizing a Cantonese utterance in native Cantonese, English speakers' native/accented Cantonese and Mandarin speakers' native/accented Cantonese.



Figure 3.3: Linguistic feature inputs for synthesizing a Mandarin utterance in native Mandarin, Cantonese speakers’ native/accented Mandarin and English speakers’ native/accented Mandarin.

3.2 Acoustic Features

Acoustic features are the acoustic properties of speech signals for speech analysis. Basic acoustic features are volume, pitch and timber. Commonly used acoustic features in speech synthesis are the Mel Frequency Cepstral Coefficients (MFCC), Mel-frequency spectra and Linear Predictive Coefficients (LPC). Acoustic features are often used as the intermediate features bridging the synthesizer and the vocoder in the back-end of a TTS system. The proposed model in this thesis uses the mel-frequency spectra as acoustic features.

3.3 Speaker Embedding

In the experiments conducted in this thesis, speaker embedding is the conditional input to the TTS system representing the identity of speakers. As in [13], they are extracted from a pretrained speaker verification system and their knowledge are then transferred to TTS systems.

3.3.1 Speaker Verification

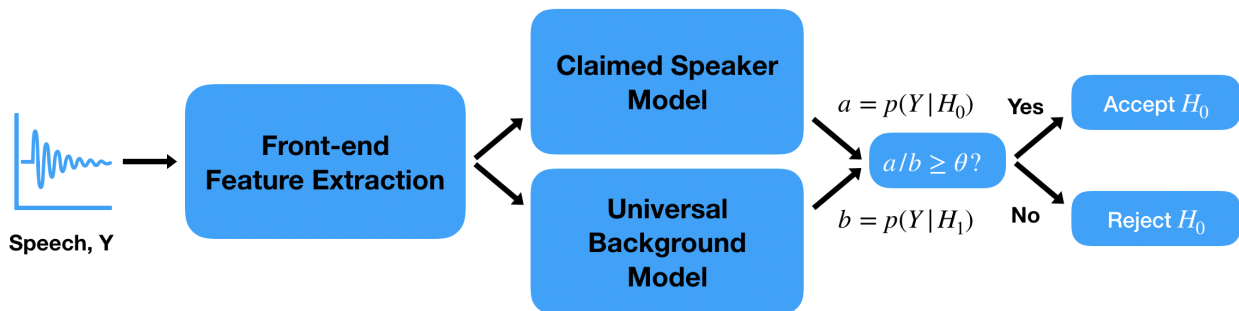


Figure 3.4: The system flow of the GMM-UBM based conventional speaker verification system.

Speaker verification is a task to verify if the input speech utterance is spoken by the claimed speaker. Before the advent of deep learning the dominant conventional speaker verification approach is based on the Gaussian Mixture Model-Universal Background Model (GMM-UBM) which produced top performances in the annual NIST Speaker Recognition Evaluations (SRE) according to [23]. In general, given a speech utterance, Y , containing speech from a single speaker and a

claimed speaker identity, S , a speaker verification system should make a final decision between the two hypotheses:

$$H_0 : Y \text{ is from the claimed speaker } S \quad (3.1)$$

and

$$H_1 : Y \text{ is not from the claimed speaker } S. \quad (3.2)$$

The final decision, ‘accept’ or ‘reject’ the hypothesis H_0 , is made using a likelihood ratio test:

$$\frac{p(Y|H_0)}{p(Y|H_1)} \begin{cases} \geq \theta, & \text{accept } H_0; \\ < \theta, & \text{reject } H_0, \end{cases} \quad (3.3)$$

where $p(Y|H_i), i \in \{0, 1\}$ is the likelihood of the hypotheses H_i given Y , and θ is the threshold.

Different speaker verification systems model the likelihoods differently. As shown in Figure 3.4, the GMM-UBM based approach with speaker adaptation [23] uses the GMM-UBM trained by many background speakers to model the likelihood of H_1 and uses the speaker model which is adapted from the UBM trained by speaker-dependent speech to model the likelihood of H_0 . However, speaker representations are introduced in the recent speaker verification systems including the d-vector [31], i-vector [15, 4, 9] and x-vector [25] systems with the Probabilistic Linear Discrimination Analysis (PLDA) back-end.

The i-vector-PLDA system consists of the UBM-based i-vector system and the PLDA scoring back-end. The i-vector system defines a low-dimensional space named total variability space which models both speaker and channel variabilities. In the contrary, these variabilities are modeled separately in another SV approach proposed in [14] before the i-vector. In the i-vector approach, given a speech utterance, its feature vectors are modeled by a GMM supervector, M , with the mean supervector m from the trained UBM and the low-rank covariance matrix T . The new total variability space is defined in Equation 3.4 by factor analysis:

$$M = m + Tw, \quad (3.4)$$

where w consists of the loading factors. To train the i-vector system, a GMM-UBM is first trained with many background speakers' speech. The mean vector of the UBM is denoted as m . The total variability matrix T can be further trained with the Expectation Maximization (EM) algorithm. After training the i-vector system, a PLDA back-end is trained with extracted i-vectors for scoring the likelihoods.

The d-vectors are the speaker representations extracted from a deep neural network (DNN) trained with frame-level speech features and the softmax cost function. The x-vectors are also DNN-based speaker representations. However, they are different from the d-vectors by aggregating the frame-level features to utterance-level features using a statistics pooling layer. PLDA is also used in [25] as the back-end for better performance. The details of the x-vector system is described in Section 4.4.

D-vectors proposed in [32] are used for multi-speaker TTS synthesis by transfer learning in [13]. In our thesis, we test the performance of the state-of-the-art x-vectors in speech synthesis.

3.3.2 Equal Error Rate

To evaluate speaker verification systems, a series of trials are prepared with pairs of utterances and speaker embeddings. If the utterance is uttered by the claimed speaker identity represented by the embedding, the trial is labeled as target, otherwise non-target. Equal error rate is the acceptance and rejection error rates (FAR and FRR), defined in Equations 3.7 and 3.8, when they are equal. The system is better if the equal error rate is lower.

$$\textit{False Accept (FA): accept when the speaker is an impostor} \quad (3.5)$$

$$\textit{False Reject (FR): reject when the speaker is legitimate.} \quad (3.6)$$

$$\textit{False Acceptance Rate (FAR)} = \frac{\textit{Number of FA errors}}{\textit{Number of non-target trials}} \quad (3.7)$$

$$\textit{False Rejection Rate (FRR)} = \frac{\textit{Number of FR errors}}{\textit{Number of target trials}} \quad (3.8)$$

3.4 Language Embedding

Since it is possible to have the same pattern of phoneme inputs among the utterances in different languages, language embedding is another conditional input to TTS system which represents the language identity when the linguistic features alone can't reveal the ambiguity of the input language. However, language embedding is unnecessary in our system where the input tone/stress embedding is language-dependent.

3.5 Mean Opinion Score

Mean opinion score (MOS) is a popular subjective measure of speech quality. Currently, there are no good objective measures available, thus MOS is widely adopted in evaluating TTS systems. MOS is the arithmetic mean of the human raters' opinions on the synthesized speeches. Absolute category rating scale is used in MOS tests. For example, a commonly used scale is the Bad-Excellent scale mapped to rational numbers from 1 to 5. Table 3.2 shows the mapping used in our MOS tests. We use the Bad-Excellent scale and the Dissimilar-Same Speaker scale which are mapped to numbers 1-5 with 0.5 increments. The Bad-Excellent scale is used for the measure of intelligibility and naturalness. The Dissimilar-Same Speaker scale is for the measure of speaker similarity.

Figure 3.6 and 3.7 show the question page for testing each aspect. We design the MOS tests such that each rater is required to indicate their fluency in each language in three levels, foreign, fluent or native. The question page is shown in Figure 3.5. Only the responses from respondents who indicate at least fluent if not native in all languages are used to compute the MOS.

Rating	Bad-Excellent	Dissimilar-Same Speaker
5	Excellent	Same speaker
4.5	-	-
4	Good	Similar
2.5	-	-
3	Fair	Slightly similar
2.5	-	-
2	Poor	Slightly dissimilar
1.5	-	-
1	Bad	Dissimilar

Table 3.2: Absolute category rating scale for MOS test in intelligibility, naturalness and speaker similarity.

Q4. Please indicate your fluency in Cantonese, English and Mandarin

	Native	Fluent	Foreign
Cantonese	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
English	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Mandarin	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Figure 3.5: Question page for collecting language fluency information.

Q6. Please listen to the audio and give your opinion on its intelligibility (I) and naturalness (N).

Transcript: I opened my eyes upon a strange and weird landscape.

▶ 0:00 / 0:04 ———— 🔊 ⋮

Bad 1 2 Poor 3 4 Fair 5 6 Good 7 8 Excellent 9

Intelligibility

Naturalness

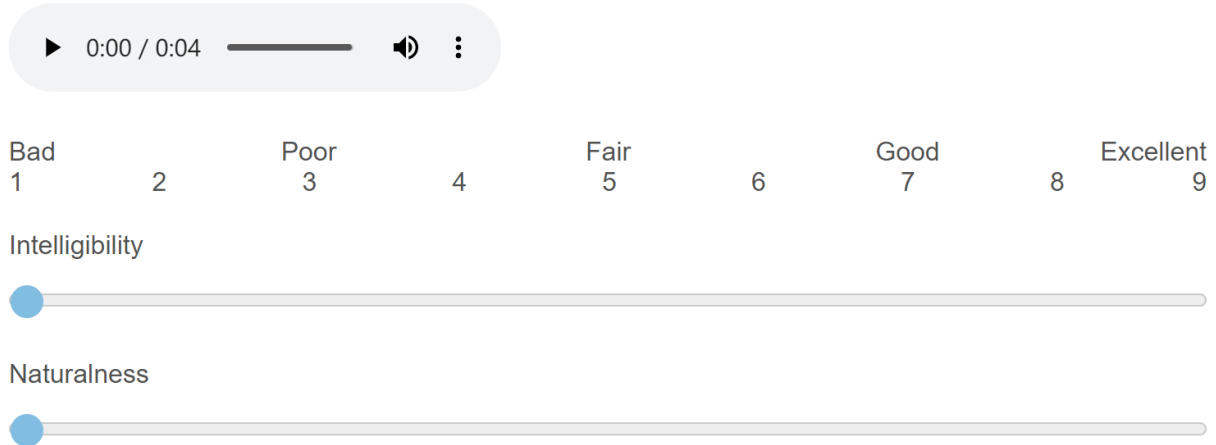


Figure 3.6: Question page for testing intelligibility and naturalness.

Q49. Please listen to the following audios and give your opinion on the similarity of their voices.

▶ 0:00 / 0:04 ———— 🔊 ⋮ ▶ 0:00 / 0:02 ———— 🔊 ⋮

Dissimilar 1 2 Slightly dissimilar 3 4 Slightly Similar 5 6 Very Similar 7 8 Same speaker 9

Speaker Similarity

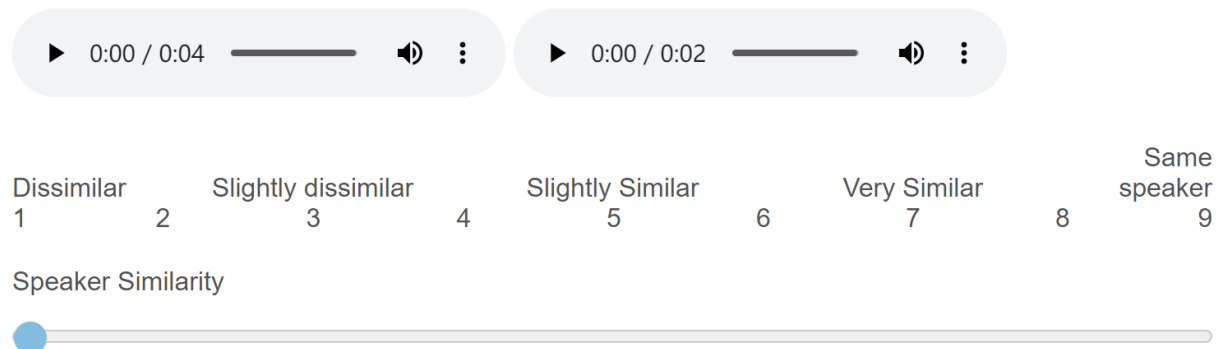


Figure 3.7: Question page for testing speaker similarity.

CHAPTER 4

SYSTEM DESIGN

4.1 Overview

Our proposed system consists of a speaker encoder, a mel-spectrogram synthesizer and a vocoder. All three components are separately trained. The speaker encoder produces x-vector speaker embedding. After it is trained, speaker embedding of new speakers can be extracted by enrolling their speech. The mel-spectrogram synthesizer is trained with x-vectors, and linguistic features and acoustic features derived from text-audio pairs. Linguistic features includes phoneme embedding and tone/stress embedding. Acoustic features are mel scale spectrograms computed from training audio waveforms. Both the speaker encoder and the mel-spectrogram synthesizer are trained with multi-lingual data from various corpora. The vocoder, WaveNet, is trained with mel-spectrograms and audio waveform in CUSENT.

4.2 Speech Corpora

The experiments conducted in this thesis are trained on 4 speech corpora in three languages: (1) Librispeech [20]; (2) SurfingTech [28]; (3) Aishell [5]; (4) CUSENT [16]. Librispeech is an open-source English corpus. There are two open-source Mandarin corpora, SurfingTech and Aishell. The CUSENT corpus is in Cantonese. They are all transcribed multi-speaker speech corpora sampled at 16kHz.

4.2.1 Librispeech

Librispeech is an English speech corpus derived from audio books of LibriVox ¹ corpus. It consists of the 100-hr and 360-hr ‘train-clean’ sets and another 500-hr set named ‘train-other’.

¹<https://librivox.org/>

hr and 360-hr ‘train-clean’ sets are selected with speeches of higher quality and accents closer to American English. On the contrary, the 500-hr set has substantial background noise. The cleaner sets are used as the English training data for all of the experiments. The selected subsets have 564 female speakers and 608 male speakers with around 25 minutes of speech per speaker. Librispeech has a test set which contains 40 speakers with the same amount of data per speaker. Linguistic models, lexicon and orthographic transcriptions are provided by the corpus.

4.2.2 SurfingTech

SurfingTech is a Mandarin corpus consists of 855 speakers, 102600 utterances in total. Each speaker has approximately 10 minutes of data. We further split the corpus to a training and testing set by randomly selecting 800 speakers for training and 55 speakers for testing. Orthographic transcriptions are provided.

4.2.3 Aishell

Aishell is another Mandarin corpus consisting of 400 speakers with around 26 minutes of speech per speaker. The training, development and test sets contain 340, 40 and 20 speakers, respectively. In all of the experiments, only the training and test sets are used.

4.2.4 CUSENT

CUSENT is a Cantonese corpus consist of 68 training speakers and 12 testing speakers with approximately 17 minutes of speech per speaker. CUSENT provides both orthographic and phonetic transcriptions in Jyupting.

4.3 Linguistic and Acoustic Features

Instead of character representation in [26, 33, 24], the input text in any of the three languages are represented by its phoneme sequence which has been proved in [13, 35] to generate more natural

speech. The training set of Librispeech, SurfingTech and CUSENT are used to train the multi-lingual synthesizer that will generate spectrogram of speech in Cantonese, Mandarin or English. They are preprocessed differently to prepare linguistic features.

4.3.1 Denoising

Even though the ‘train-clean’ training sets in Librispeech have relatively smaller background noise compared to the 500-hr ‘train-other’ training set, the background noise can still affect the quality of synthesized speech according to [13] and some of our preliminary experiments. Audio denoising with block thresholding technique [34] is applied to English training data. No denoising is required for the SurfingTech and CUSENT corpus, since they contain no apparent background noise.

4.3.2 Forced Alignment

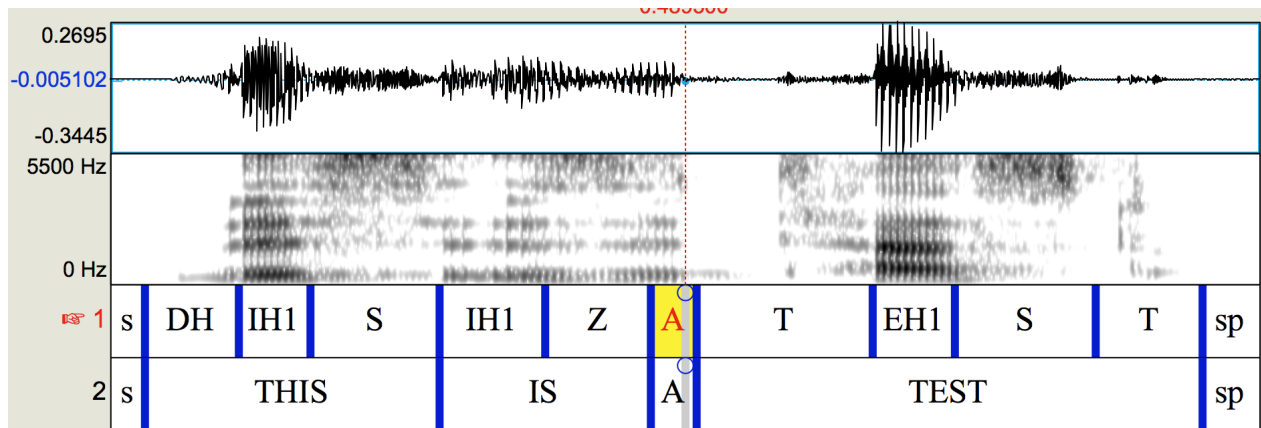


Figure 4.1: An example of forced alignment between the audio waveform and word/phoneme-level transcriptions.

Forced alignment, as shown in Figure 4.1, is a technique to align the orthographic or phonetic transcription with audio waveform along the time axis. As a result, precise time locations of boundaries of both words and phonemes are computed and noted for an audio waveform.

Librispeech and SurfingTech contain significant short pauses in the middle of their utterances. Forced alignment helps to label the silence region in the middle of the utterances where the short

pauses occurs. It helps the synthesizer to capture the boundaries of regular phonemes more precisely to obtain more natural prosody in the synthesized speech.

The Montreal Forced Alignment toolkit [17] is an open-source tool for forced alignment. It provides pretrained acoustic models and grapheme-to-phoneme models for both English and Mandarin. The forced alignment for Librispeech is straightforward since the pretrained models are originally trained on Librispeech where phonemes are represented in ARPABET phoneme set. On the contrary, preprocessing is required for forced aligning SurfingTech utterances since they provide only orthographic transcriptions and the grapheme-to-phoneme model is trained with pinyin. Google translate is applied for the conversion from the Chinese orthographic transcriptions to their pinyin.

4.3.3 Input Representation

To synthesize speeches in multiple languages, a shared phoneme set is created by mapping all pinyin phonemes in Mandarin and Jyupting phonemes in Cantonese to ARPABET phonemes. Three exceptions of pinyin phonemes ‘j’, ‘q’ and ‘x’ are treated as distinct phonemes as no good ARPABET mappings are found. The mapping table is attached in the appendix. The phonemes in the phoneme set are represented by 512-D vectors learned from one-hot vectors which are referred as phoneme embeddings.

To improve the naturalness of the synthesized speech, tone/stress embedding are input additional to phoneme input. They are represented as 14-D one-hot embeddings concatenated to phoneme embedding.

4.3.4 Acoustic Features

In this thesis, we use mel-frequency spectra as the acoustic features. To be specific, the output of the synthesizer and the conditional input of the WaveNet are sequences of mel-frequency spectra named mel-spectrogram. A linear spectrogram is a sequence of spectra which is a spectral vector in frequency-domain derived by applying short-time Fourier transform (STFT) to speech signal in time-domain. Mel-spectrograms are spectrograms in mel scale, which is introduced based on

Layer	Layer context	Tot. context	In x out
Frame 1	$[t-2, t, t+2]$	5	$5F \times 512$
Frame 2	$\{t-2, t, t+2\}$	9	1536×512
Frame 3	$\{t-3, t, t+3\}$	15	1536×512
Frame 4	$\{t\}$	15	512×512
Frame 5	$\{t\}$	15	512×1500
Stats pooling	$[0, T)$	T	$1500T \times 3000$
Segment 6	$\{0\}$	T	$3000 \times n-D$
Segment 7	$\{0\}$	T	$n-D \times 512$
Softmax	$\{0\}$	T	$512 \times N$

Table 4.1: The detailed configurations of the x-vector network where the x-vector dimension is denoted as $n-D$, F denotes the dimension of filterbanks features of one frame, T denotes the total number of frames of the input utterance, and N denotes the number of training speakers.

human perception of speech pitches, instead of in Hertz scale. In the thesis, the STFT uses 50 ms frame length, 12.5 ms hop size and the Hanning window. The spectral energies are computed over an 80 channel mel filterbanks spanning from 125 Hz to 7600 Hz.

4.4 Speaker Embeddings - x-vectors

X-vector is known as the state-of-the-art method in speaker verification tasks. As depicted in 4.2, the x-vector system we use is a TDNN consisting of 7 hidden layers, a statistics pooling layer and a softmax layer. The first five hidden layers operates on frame level temporal contexts. Assuming that there are T frames input, the first hidden layer captures a context over 5 frames centered at the current frame at time t . The second and third layer each captures 3 previous layer outputs centered at current output with strides of 2 and 3 frames, respectively. The fourth and fifth layers only capture current output. Building up temporal contexts over the first three hidden layers expands the DNN’s frame-level receptive field up to 15 frames before statistics pooling. Statistics pooling layer aggregates outputs from the fifth layer throughout all frames input from time 0 to time T . Their mean and variance are computed as segment-level features capturing information over T frames and are forwarded to the last two hidden layers. All nonlinearities are rectified linear units (ReLU). The entire network is trained to classify the training speakers in the training data. After training, x-vectors are extracted from the sixth hidden layer output before ReLU. In this thesis, 64-D, 128-D

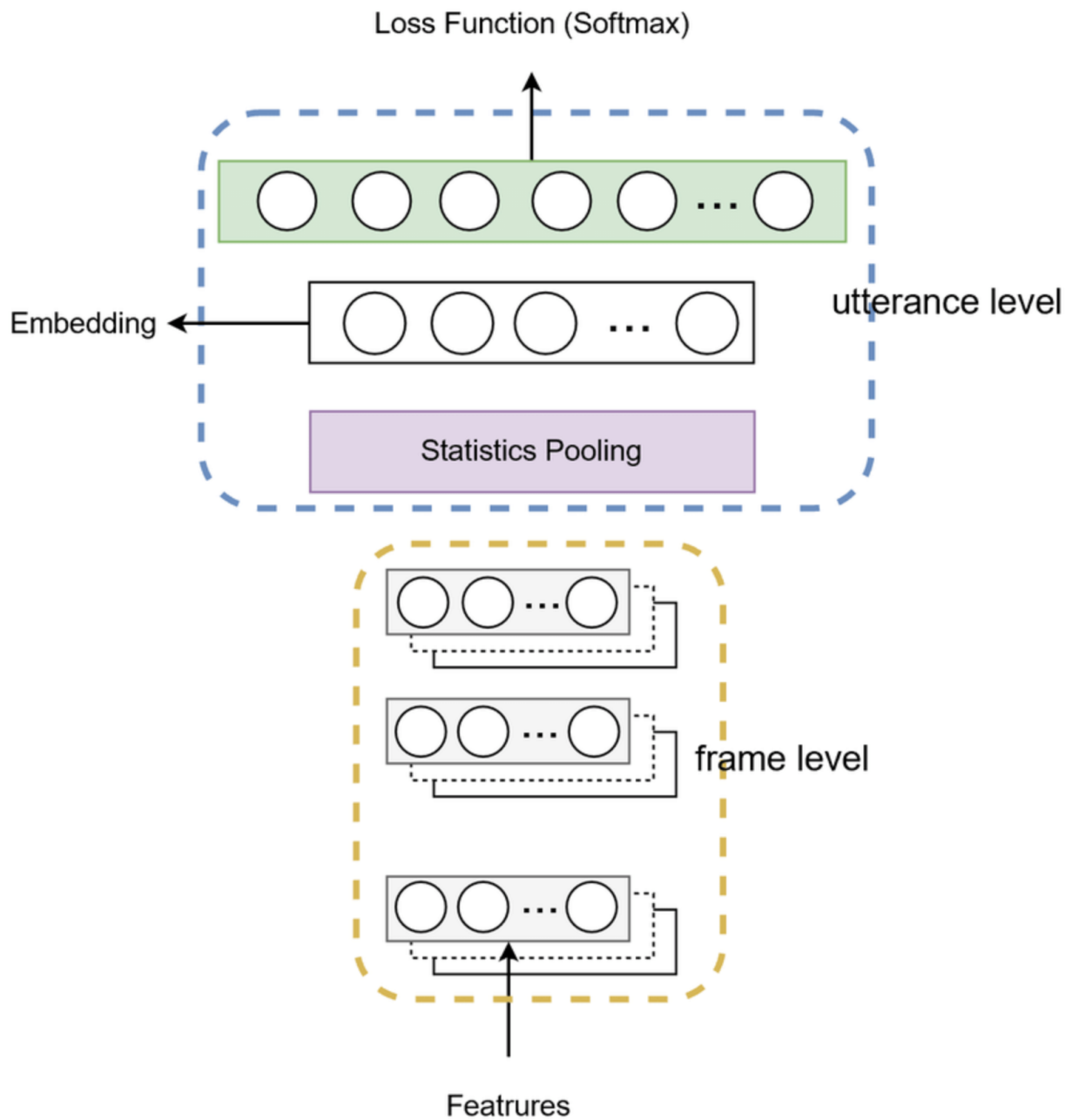


Figure 4.2: The deep neural network architecture of the x-vector system.

and 512-D (n-D) x-vectors are tried.

A Probabilistic Linear Discriminant Analysis (PLDA) backend [12] is used in x-vectors for scoring paired speaker embeddings. First, LDA is applied to reduce speaker embedding dimen-

sions. LDA outputs are length-normalized before PLDA is applied. Finally, PLDA scores are normalized with adaptive s-norm [27].

In the experiments, x-vector system is built with standard parameters mentioned in [25] and the provided recipe in Kaldi toolkit [22]. Speaker embedding dimension might be changed in different experiments by changing the output dimension of the sixth hidden layer.

4.5 Neural Mel-spectrogram Synthesizer

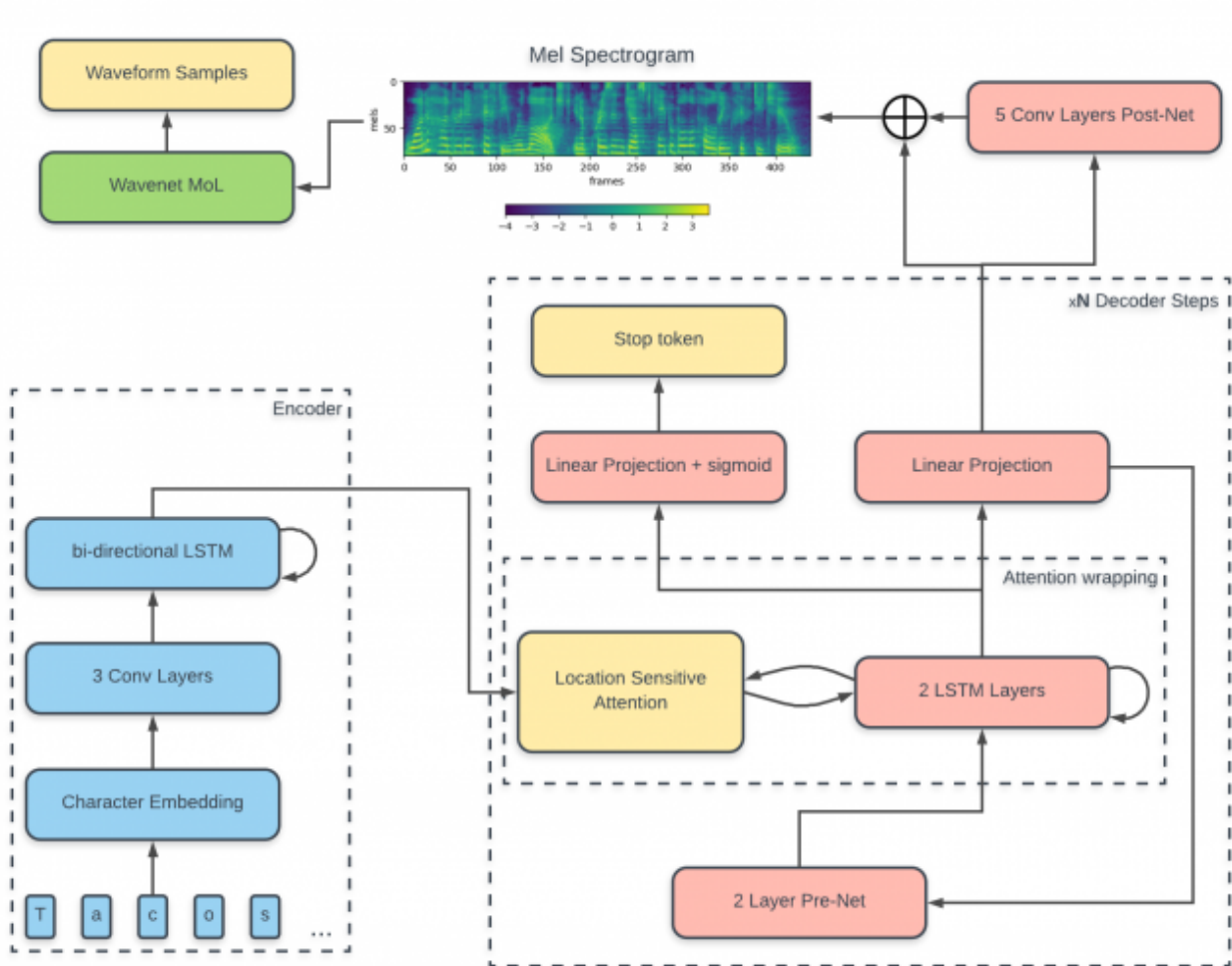


Figure 4.3: The architecture of Tacotron 2 including the encoder and decoder of the synthesizer and the WaveNet.

The mel-spectrogram synthesizer is a modified version of the synthesizer in Tacotron 2 [24]. In [24], it is an attention-based encoder-decoder neural network which is trained with character embeddings and mel-spectrograms. Figure 4.3 describes the entire architecture of Tacotron 2 including the synthesizer and the WaveNet. The encoder of the synthesizer computes a sequence of hidden representation from input character embeddings and inputs them to the attention mechanism. A context vector is summarized by the attention network which is given to the decoder to predict the mel-spectrogram frame by frame. The synthesizer in the proposed system is implemented similarly to that in Tacotron 2 except a few differences: (1) phoneme embedding and tone/stress embedding are used instead of character embedding. The tone/stress embedding is concatenated to the phoneme embedding to produce a 526-D embedding before they are input to the encoder. (2) The speaker embedding is concatenated to the encoder’s output hidden representations same as in [13].

In detail, Tacotron 2 adopts a sequence-to-sequence architecture converting character embedding to mel-spectrograms which can be further input to a vocoder, such as WaveNet, for waveform generation. According to [24], the MOS results show that the synthesized speech is very close to human’s genuine speeches. Instead of using many separate acoustic features, Tacotron 2 uses a lower level acoustic feature, the mel-spectrogram, to bridge the synthesizer and vocoder. The mel-spectrograms can be computed efficiently from audio signals. Using it also enables the separate training of the synthesizer and the vocoder such that it can cooperate with the WaveNet.

4.5.1 The Encoder

The encoder comprises a stack of 3 convolutional layers and one bi-directional LSTM layer. Each of the convolutional layers consists of 512 kernels of shape 5×1 followed by a batch normalization and Relu activations. Each kernel spans 5 characters and the stack of the these convolution layers increases the receptive field to model a longer context. These CNNs computes local deep features from the input embeddings while capturing long-term dependencies. The bi-directional LSTM layer consists of 256 units in each direction. It receives the output from the convolutional layers and predicts a sequence of hidden representations $h_t(h_1, h_2, \dots, h_L)$ over the L-length input utterance as the encoded features.

4.5.2 Attention Mechanism

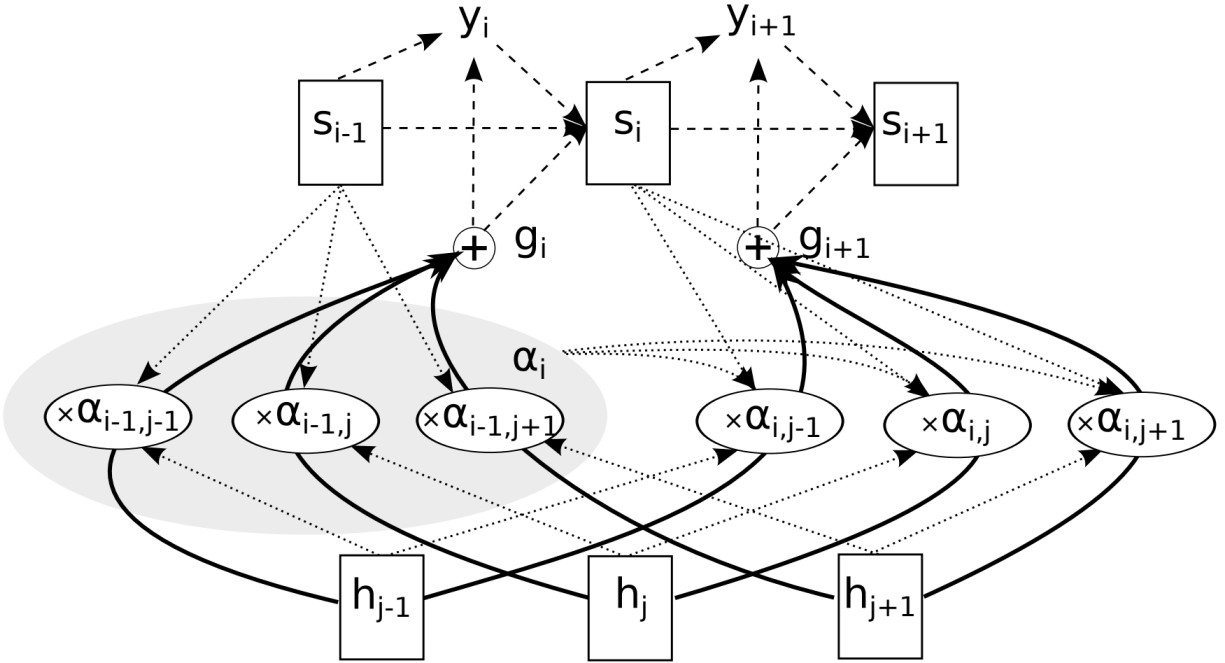


Figure 4.4: A general frame work of attention mechanism.

The attention mechanism used in Tacotron 2 is a location sensitive attention network proposed in [7] for speech recognition. Different from the original attention mechanism proposed in [3], the location sensitive attention is shown to be more adaptive for longer testing inputs than training utterances in speech recognition. The same situation also happens in speech synthesis. Shorter training utterances increases the batch size to stabilize the training process. But it is quite often to synthesize longer utterances in testing. In our experiments, the longest training utterances are 10 seconds. In real life applications, the system is often used to generate longer speech.

$$h_i = \text{Encoder}(x_i, h_{i-1}) \quad (4.1)$$

Figure 4.4 shows the architecture of a generic attention mechanism. The attention mechanism is first introduced to an RNN-based Encoder-Decoder neural network which generates an output sequence $y, (y_1, \dots, y_t)$ from the input $x(x_1, x_2, \dots, x_L)$. In this network, the input x is first processed

by a recurrent neural network encoder which outputs a sequence of hidden representation sequence $h_t(h_1, h_2, \dots, h_L)$ which is at the same length L of the input x . The process is described in Equation 4.1. The decoder predicts the output y step by step at each output time step. At the i -th step, the decoder generates y_i as follows:

$$\alpha_i = \text{Attend}(s_{i-1}, h) \quad (4.2)$$

$$c_i = \sum_{j=1}^L \alpha_{i,j} h_j \quad (4.3)$$

$$y_i = \text{Generate}(s_{i-1}, c_i) \quad (4.4)$$

$$s_i = \text{Decoder}(s_{i-1}, c_i, y_i) \quad (4.5)$$

where s_{i-1} is the $(i - 1)$ -th hidden state of the decoder recurrent neural network and α_i is the attention weights, also known as the alignment, used for the decoder to focus differently on the hidden representations h to predict the output y_i at the i -th step. The attention weights are updated at each time step according to Equation 4.2. The context vector c_i at the i -th step is computed by Equation 4.3. Based on the context vector and previous decoder hidden state, the decoder predicts the output y_i described in Equation 4.4. Finally, the decoder updates its hidden state s_i by Equation 4.5.

$$e_{i,j} = \text{Score}_1(s_{i-1}, h) \quad (4.6)$$

$$e_{i,j} = \text{Score}_2(\alpha_{i-1}, h) \quad (4.7)$$

$$e_{i,j} = \text{Score}_3(s_{i-1}, \alpha_{i-1}, h) \quad (4.8)$$

$$\alpha_{i,j} = \frac{\exp(e_{i,j})}{\sum_{j=1}^L \exp(e_{i,j})} \quad (4.9)$$

The *Attend* function is implemented by a score function. The attention weight $\alpha_{i,j}$ on encoder output h_j at the i -th step is computed by a softmax function over the scores $e_{i,j}$ computed by the *Score* function. Different attention mechanisms are distinguished as content-based attention, location-based attention and hybrid attention based on different *Score* functions. The attention

mechanism which uses the *Score* function in Equation 4.6 which only focuses on the decoder hidden state is called the content-based attention. Equation 4.7 that focuses on the previous attention weights is location-based. The attention mechanism which uses Equation 4.8 is called hybrid attention.

The location sensitive attention in the synthesizer implements a hybrid attention by adding previous alignment in its *Score* function by convolving a matrix F with α_{i-1} . The implementation of the *Score* function is described by Equation 4.11.

$$f_i = F * \alpha_{i-1} \quad (4.10)$$

$$e_{i,j} = w^T \tanh(Ws_{i-1} + Vh_j + Uf_{i,j} + b) \quad (4.11)$$

The attention summarizes the encoded hidden representations into a fixed-length context vector. Attention weights are computed after projecting inputs and location features to 128-dimensional hidden representations. Location features are computed using 32 1-D convolution filters of length 31.

4.5.3 The Decoder

The decoder autoregressively predicts a mel spectrogram from the attended context vector one frame at a time. The prediction from the previous time step is passed through a pre-net containing 2 fully connected layers of 256 hidden ReLU units. In [24], the pre-net was found essential for learning attention. The pre-net output and attention context vector are concatenated and passed through a stack of 2 uni-directional LSTM layers with 1024 units. The attention context vector is concatenated with the output of the LSTM layers and the concatenation is projected to the target spectrum and a scalar stop token through two independent linear transformations. Finally, the predicted mel-spectrum is passed through a 5-layer convolutional post-net which predicts a residual to add to the prediction to improve the overall reconstruction. Each post-net layer consists of 512 filters with shape 5×1 with batch normalization, followed by tanh activations on all except the final layer. The summed mean squared errors (MSE) from before and after the post-net are minimized in training to aid convergence. In parallel to spectrum prediction, the stop token is passed through

a sigmoid activation to predict the probability that the generation has completed. The stop token is used to dynamically terminate the generation at a threshold of 0.5.

The convolutional layers in the network are regularized using dropout with probability 0.5, and LSTM layers are regularized using zoneout with probability 0.1. In order to introduce output variation at inference time, dropout with probability 0.5 is applied only to layers in the pre-net of the autoregressive decoder.

4.6 WaveNet Vocoder

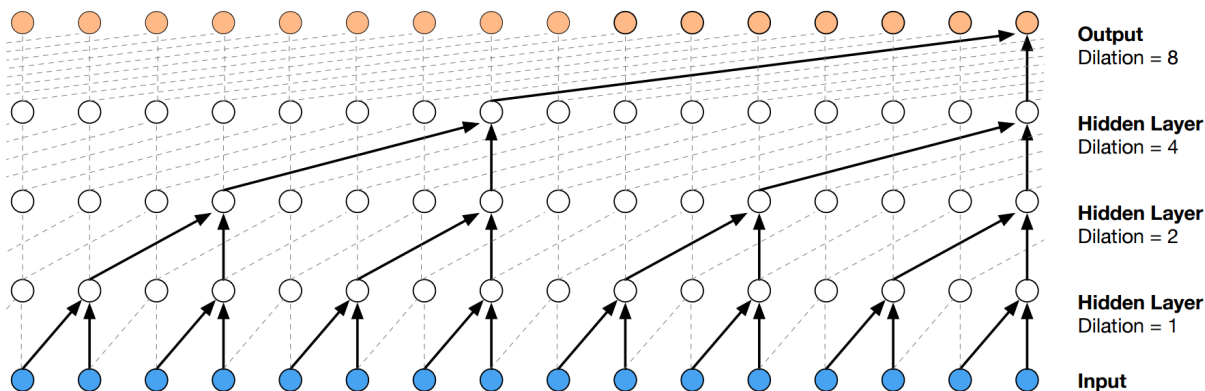


Figure 4.5: The dilated causal convolutions layers in WaveNet.

WaveNet [30] is a raw audio generation model, inspired by the PixelRNN and PixelCNN models, which can predict audio waveform sample by sample autoregressively. It is built with stacks of dilated causal convolutional layers as described in Figure 4.5. Causal convolutional networks predict the next sample at time $t + 1$ only depending on the past samples at time from 1 to t . The convolutional networks use dilations, which skips over some input values, to improve the computation efficiency and reduce the number of layers required for the same size of receptive field. At each step, a new sample is predicted and fed back as the input for the next prediction. Instead of Relu activations, WaveNet uses gated activation units:

$$z = \tanh(W_{f,k} * x + V_{f,k}^T * h) \odot \sigma(W_{g,k} * x + V_{g,k}^T * h) \quad (4.12)$$

where $*$ denotes convolution, \odot denotes element-wise multiplication, $\sigma()$ denotes sigmoid function, k denotes the layer index, f and g denotes the filter and gate, the $W_{f,k}$, $V_{f,k}$, $W_{g,k}$ and $V_{g,k}$ are trainable variables, and h is the conditional input if any. Residual and skip-connections are also used for faster convergence as shown in Figure 4.6.

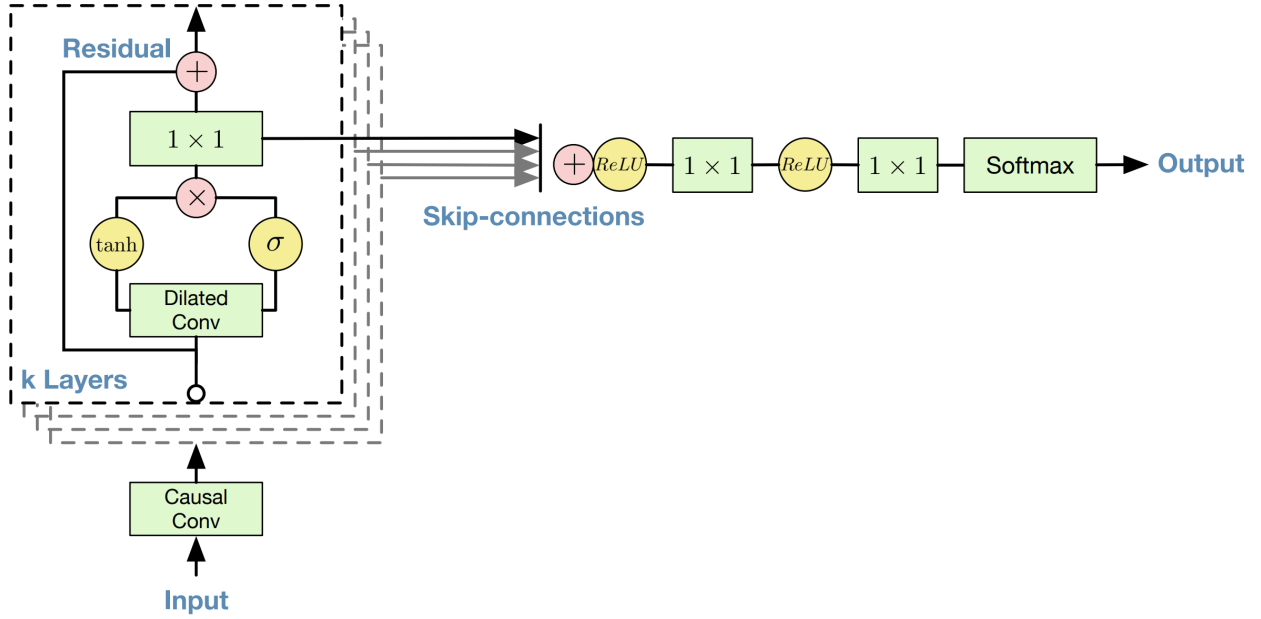


Figure 4.6: Residual block with gated activation units in WaveNet.

The conditional probability $p(x_t|x_1, x_2, \dots, x_{t-1})$ over output audio samples is modeled by a softmax function. To reduce the number of output probabilities in raw audio generation where there are 65536 possible values for a 16-bit sample, μ -law quantization is applied to quantize the 16-bit sample to 8-bit:

$$f(x_t) = \text{sign}(x_t) \frac{\ln(1 + \mu|x_t|)}{\ln(1 + \mu)} \quad (4.13)$$

It has been shown that the WaveNet model can be conditioned on conditional inputs for various applications. For example, the WaveNet model conditioned on mel-spectrograms [24, 2, 10, 21, 13, 35] can serve as a vocoder. It can also be conditioned on speaker embeddings [30] to synthesize speech in various voices. According to these past researches, conditioned on mel-spectrograms,

WaveNet can outperform traditional vocoders such as Griffin & Lim [11] and WORLD [19]. Even though it has been proposed to condition WaveNet on speaker embeddings for synthesizing speeches in their voices, conditioning WaveNet on the mel-spectrogram also works as shown in [24]. Since the spectrograms are computed from waveforms, it implies that they contain speaker information inherently which makes extra speaker embedding conditioning unnecessary.

In the thesis, the WaveNet is trained with training data of the CUSENT corpus. The WaveNet model is merely conditioned on the mel-spectrogram without any extra embeddings. The implementation of the WaveNet follows [13]. It is shown from the results that the WaveNet model trained on data in one language can successfully synthesize high-quality speech in other languages.

CHAPTER 5

PRELIMINARY EXPERIMENTS AND RESULTS

5.1 Overview

In speech synthesis, there are untrainable variables and uncertain factors that should be optimized to improve the overall quality of the system such as,

- **Input representation:** Both character embedding and phoneme embedding were used in speech synthesis. According to Tacotron 2 [24], using character embedding is successful in training an English TTS system. Whereas in the multi-speaker Tacotron 2 [13] and multi-lingual Tacotron 2 [35], it is reported that phoneme embedding has a significant advantage over character embedding. In multi-lingual TTS systems especially when there are over thousands of characters in languages such as Mandarin and Cantonese, phoneme input can reduce domain of the input embedding since there is a large overlap of phonemes between languages but not characters. In mono-lingual English multi-speaker TTS systems, we conduct experiments to verify the performances of phoneme input and character input.
- **Phoneme set:** In multi-lingual speech synthesis, there are different standard phoneme sets in each language and there are international phoneme sets such as International Phonetic Alphabet (IPA) and Extended Speech Assessment Methods Phonetic Alphabet (X-SAMPA). In the thesis, two methods are compared for multi-lingual speech synthesis, using IPA or mapping pinyin and Jyupting phonemes to the closest ARPABET phonemes. One major difference is that the mapping method encourages more sharing of phonemes among languages so as to reduce the number of phonemes in the phoneme set. However, there is hardly a perfect mapping existed among the phoneme sets of different languages. Thus we keep the unique phonemes in Cantonese and Mandarin if no similar phonemes are found in English. It is worth evaluating the performance of this mapping compared to using IPA.

- **Speaker Embedding Dimensions:** In [13], different speaker embedding dimensions have been used when the number of training speakers is increasing. It is to our interest what is the optimal speaker embedding dimension for our system. In our thesis, we compare x-vectors in 64, 128 and 512 dimensions.
- **Speaker Embedding Normalization:** Normalization is a technique used in data preprocessing before they are fed into the system which often improves the system robustness and training stability. In the thesis, l2-norm normalization and whitening are applied to x-vectors before the training of the synthesizer. Their performances in speech synthesis are compared with no normalization.

We conducted several preliminary experiments to optimize aforementioned factors one by one. Experiments setup and their results are presented in the following subsections. Short conclusions are made based on the results. To synthesize speech faster, we use Griffin & Lim vocoder in these experiments.

5.2 Input Representation: Character/Phoneme

The experiment is conducted with a pair of mono-lingual English multi-speaker TTS systems trained on Librispeech data with character embedding and phoneme embedding respectively. The speaker encoder is a 128-D x-vector system. The synthesizer is as described in Chapter 3.

The speeches synthesized by the model with phoneme embedding input sounds more natural than that with character embedding. Sometimes synthesized speeches with character embedding have pronunciation mistakes. The prosody is also less natural because of the segmentation mistakes. The synthesized speech with character input have unexpected pauses. In conclusion, phoneme embedding can perform better than character embedding in mono-lingual English multi-speaker speech synthesis.

5.3 X-vector Dimension

In this subsection, we present the speaker verification EER results for x-vectors in 64, 128 and 512 dimensions. Enrollment utterances are 3 minutes long and test utterances vary from 5 seconds to 12 seconds. Table 5.1 shows the speaker verification EER of these x-vectors. Even though the 128-D x-vector system has the highest EER, we found that the TTS system can synthesize better quality speeches when trained with 128-D x-vector. In contrast, although 64-D x-vectors give the best SV EER, they produce audios of poorer quality in our TTS system. We also found the synthesized speeches using the TTS system with 128-D x-vectors have higher speaker similarity than 64-D x-vectors. It seems that speaker embeddings that give better SV EER is no guarantee for better synthesized audios. At the end, we choose the 128-D x-vectors for our speaker embeddings.

We further investigate the x-vector embedding with more training speakers. Table 5.2 shows speaker verification EER on a separate test set for systems trained with increasing number of speakers from various corpora. Results show that adding more speakers from other datasets does not further reduce the speaker verification EER. We use the x-vectors trained with the most number of speakers as our speaker embeddings.

System	Dim	Train set	Speakers	SV-EER
x-vector	64	LS	1172	1.00
x-vector	128	LS	1172	1.50
x-vector	512	LS	1172	1.25

Table 5.1: Librispeech SV EER (%) for x-vectors in different dimensions.

Train set	Speakers	LS	CU	ST	AI
LS, CU	1240	0.75	0	–	–
LS, ST, AI	2312	0.75	–	0	0.5
LS, CU, ST, AI	2380	0.75	0	0	0.5

Table 5.2: Effect of increasing number of training speakers on Librispeech SV EER (%) using 128-D x-vector.

5.4 IPA vs ARPABET

We implement three multi-lingual Cantonese and English TTS systems, one using IPA and language embedding, one using IPA and no language embedding and one using the ARPABET mapping with language embedding. The speaker encoder is the 128-D x-vector system trained on data from all corpora. The synthesizer is trained with English and Cantonese data with no tone/stress embedding. Speaker embedding and language embedding, if any, are input to the synthesizer through a concatenation with the encoder’s output.

From the experiment results, although the synthesized Cantonese speech sounds less natural when there is no tone/stress embedding, it appears that no one model has significant advantages over the others in the aspect of audio quality of both English and Cantonese speech. It is expected that the synthesized speeches could be sound subtle accented in the other language when there is no language embedding. However, it is not obvious in the real synthesized speeches. The ARPABET mapping and IPA perform equally in both English and Cantonese. It seems the mapping method does not compromise the audio quality while reducing the input domain. And the results also show that the mapping method makes the training be more stable when the amount of English speech to Cantonese speech is very unbalanced with a ratio of 20:1 (400 hours to 20 hours). We found the system using the mapping method has less difficulty to stop during the inference.

5.5 Speaker Embedding Normalization

Experiments are conducted to compare L2-norm normalization, ZCA-whitening and no normalization on speaker embeddings. The speaker encoder is the same as above. Before training the synthesizer, the x-vectors of training speakers are normalized with different normalization techniques. Different from above experiments, the synthesizer is trained with English and Mandarin corpora with a 8-D one-hot tone/stress embedding which only represents English stresses and Mandarin tones.

Figures 5.1, 5.2 and 5.3 show the t-SNE visualization of x-vectors of training speakers and enrollment speakers after normalization. In these figures, ‘-’ denotes the speaker embeddings of

female speakers and ‘|’ denotes the speaker embeddings of male speakers. The numbers in the figures are the speaker identities which denote the speaker embeddings of the selected speakers for synthesizing speeches used in MOS tests. These speaker embeddings are also denoted in the legends of the figures in format ‘MOS-(Uns/S)eenSpk’. The legends of the figures in formats ‘corpus name-train’ and ‘corpus name-enroll’ denote the training and enrollment speaker embeddings in the four corpora. From the results, it is observed that the speaker embeddings form clusters separable by gender and corpus. We hypothesize that these x-vectors, no matter normalized or not, contain language information and gender information. The x-vectors after whitening in Figure 5.3 seem to form a big cluster but they still form their clusters separable from each other in Figure 5.1.

Tables 5.4, 5.5 and 5.6 show the MOS results mean and standard deviation of intelligibility, naturalness and speaker similarity on unseen speakers. We use these notations shown in Table 5.3 for different types of synthesized speech. From the results, both L2-norm and whitening normalization improve the intelligibility of native and foreign native speeches. It is observed that whitening helps generate very natural native English and foreign native Mandarin for Librispeech speakers. L2-norm helps produce better accented speeches. Whitening helps generate speeches slightly more similar to target speaker’s voice than L2-norm normalization.

Notation	Meaning
EAM	Accented English spoken by Mandarin speakers
EAC	Accented English spoken by Cantonese speakers
ENM	Native English spoken by Mandarin speakers
ENC	Native English spoken by Cantonese speakers
EN	Native English.
CAE	Accented Cantonese spoken by English speakers
CAM	Accented Cantonese spoken by Mandarin speakers
CNE	Native Cantonese spoken by English speakers
CNM	Native Cantonese spoken by Mandarin speakers
CN	Native Cantonese.
MAC	Accented Mandarin spoken by Cantonese speakers
MAE	Accented Mandarin spoken by English speakers
MNC	Native Mandarin spoken by Cantonese speakers
MNE	Native Mandarin spoken by English speakers
MN	Native Mandarin.

Table 5.3: Notations used in MOS results.



Figure 5.1: t-SNE visualization of x-vectors.

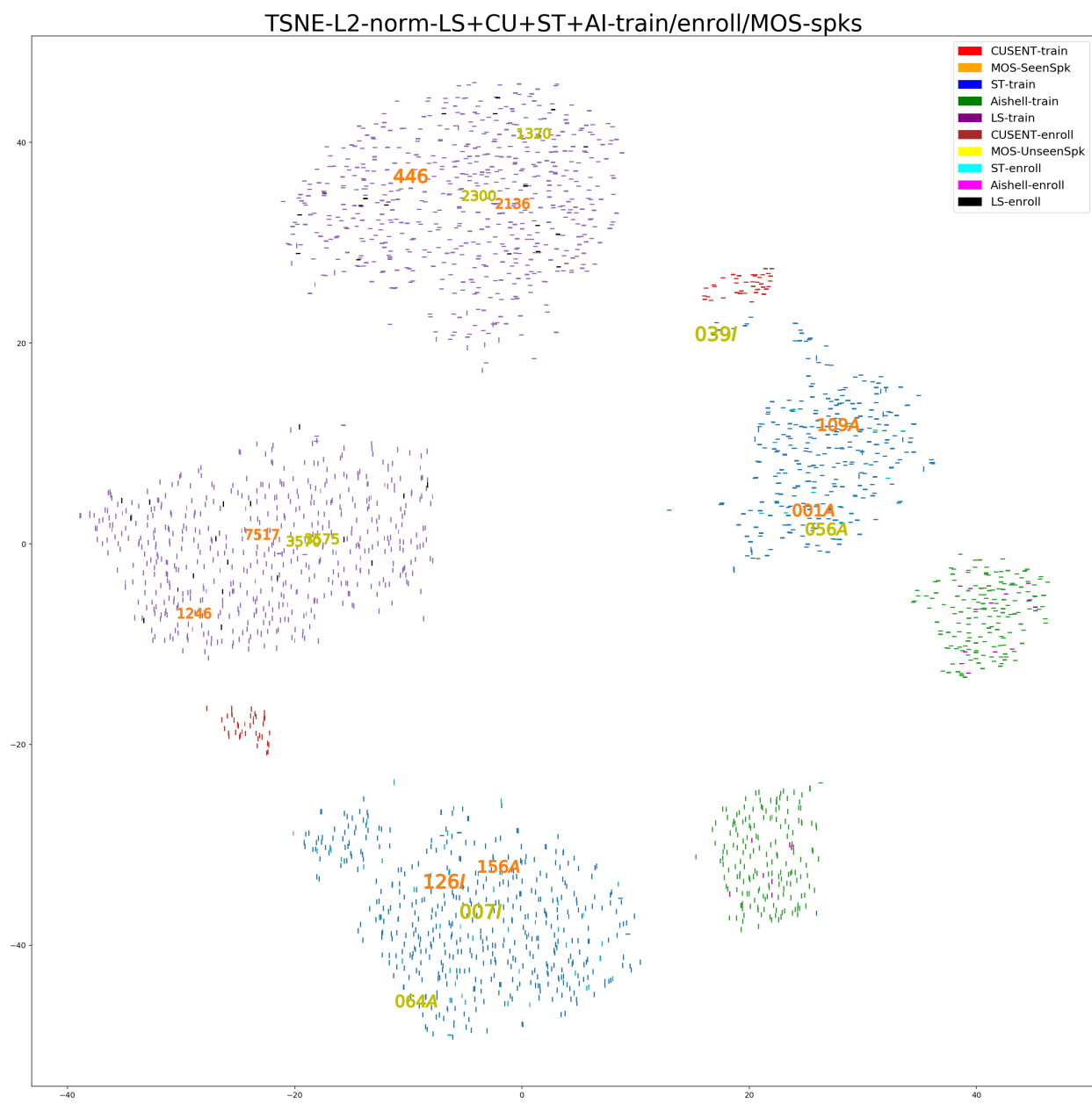


Figure 5.2: t-SNE visualization of x-vectors after L2-norm normalization.

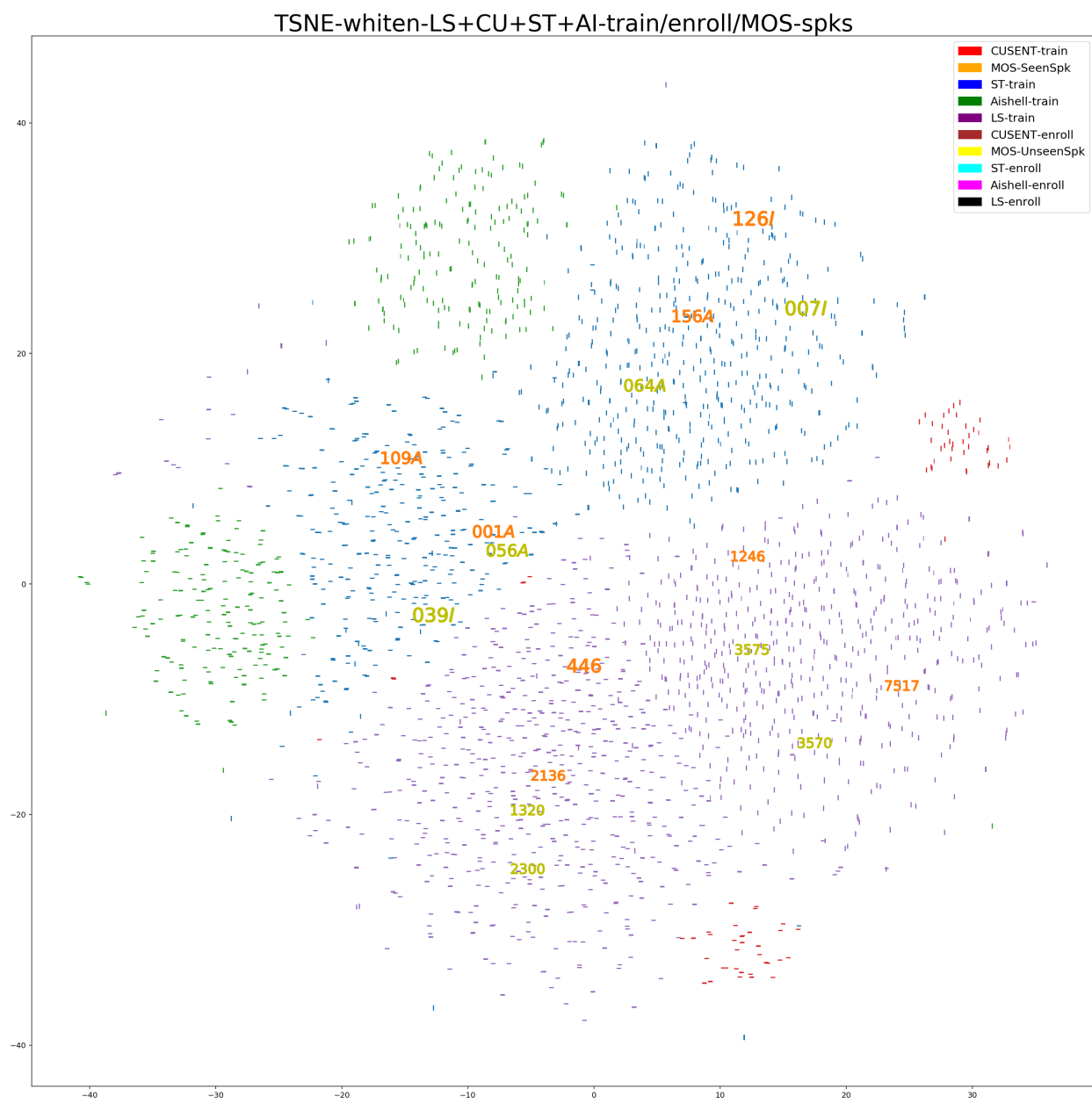


Figure 5.3: t-SNE visualization of x-vectors after whitening.

Model\Cases	EAM	ENM	EN
No-norm	2.08 ± 0.23	3.50 ± 0.52	4.67 ± 0.18
L2-norm	2.29 ± 0.35	4.92 ± 0.10	4.54 ± 0.21
Whitening	1.92 ± 0.27	4.25 ± 0.24	4.92 ± 0.07
Model\Cases	MAE	MNE	MN
No-norm	1.75 ± 0.23	2.83 ± 0.25	4.77 ± 0.14
L2-norm	1.88 ± 0.29	4.46 ± 0.17	4.71 ± 0.15
Whitening	1.42 ± 0.19	4.08 ± 0.44	4.67 ± 0.13

Table 5.4: Intelligibility MOS (mean \pm standard deviation) on unseen speakers.

Model\Cases	EAM	ENM	EN
No-norm	1.88 ± 0.22	3.79 ± 0.28	4.33 ± 0.22
L2-norm	1.96 ± 0.09	4.08 ± 0.19	4.08 ± 0.23
Whitening	1.47 ± 0.16	3.46 ± 0.26	4.58 ± 0.19
Model\Cases	MAE	MNE	MN
No-norm	1.88 ± 0.38	1.92 ± 0.26	4.18 ± 0.23
L2-norm	2.25 ± 0.36	3.17 ± 0.48	4.21 ± 0.24
Whitening	1.75 ± 0.28	3.58 ± 0.33	4.33 ± 0.20

Table 5.5: Naturalness MOS (mean \pm standard deviation) on unseen speakers.

Model\Cases	EAM	ENM	EN
No-norm	2.69 ± 0.37	2.62 ± 0.32	3.46 ± 0.30
L2-norm	2.46 ± 0.34	2.29 ± 0.22	3.37 ± 0.32
Whitening	2.19 ± 0.34	2.08 ± 0.35	3.50 ± 0.23
Model\Cases	MAE	MNE	MN
No-norm	3.21 ± 0.25	1.25 ± 0.14	3.32 ± 0.41
L2-norm	3.75 ± 0.21	2.08 ± 0.27	3.17 ± 0.29
Whitening	3.75 ± 0.14	1.67 ± 0.15	3.33 ± 0.32

Table 5.6: Speaker similarity MOS (mean \pm standard deviation) on unseen speakers.

CHAPTER 6

EVALUATION & RESULTS

6.1 Baseline

We build a mono-lingual English multi-speaker TTS system as the baseline model. It uses phone-me input and whitened 128-D LS+CU+ST+AI x-vectors (i.e. the speaker encoder is trained using LibriSpeech, CUSENT, SurfingTech and Aishell corpora). The synthesizer is trained on Librispeech. The WaveNet is trained on CUSENT.

6.2 Proposed

The proposed system is built with a synthesizer trained with Cantonese, English and Mandarin data. Pinyin and Jyupting phonemes of Mandarin and Cantonese training utterances are mapped to ARPABET. Tone/stress embedding is the 14-D one-hot embedding introduced in Chapter 4. Speaker embeddings are whitened 128-D LS+CU+ST+AI x-vectors. The synthesizer is trained on Librispeech, SurfingTech and CUSENT. The WaveNet vocoder is trained on CUSENT.

6.3 Experiments and Results

We conduct MOS tests to test the quality of generated speech in intelligibility, naturalness and speaker similarity. Ground truth speech from 2 seen and 2 unseen English speakers, 2 unseen Cantonese speakers and 2 unseen Mandarin speakers are tested as the reference. Each pair of speakers contains one male and one female speaker. Using the baseline English TTS system, one English utterance ‘Through out the centuries people have explained the rainbow in various ways.’ is synthesized for the four English speakers. Using the proposed multi-lingual TTS system, the same English utterance, one Cantonese utterance in Figure 3.2 and one Mandarin utterance in Figure 3.3

are synthesized for the 6 unseen speakers in three languages. For each utterance, we synthesize native speech, foreign native speech and foreign accented speech for corresponding speakers, respectively. We have also synthesized such speeches for seen speakers in all the languages. To reduce the duration of the MOS tests to an acceptable range, we didn't test those synthesized speeches for seen speakers. However, we listened carefully to those speeches and observed that they are slightly better than synthesized speeches for unseen speakers where similar behaviour can be observed in the baseline results.

In total, 20 responses are collected from raters from Guangdong province in mainland who are at least fluent in all the three languages. MOS results are shown in Table 6.1, 6.2 and 6.3.

6.4 Discussions

6.4.1 Intelligibility and Naturalness

Comparing the MOS results between the ground truth English speeches and the synthesized speeches using the baseline model (case 1 and case 4/5), the synthesized speeches are highly intelligible even though they are not as good as the ground truth speeches. The ratings of speeches for both seen and unseen speakers are slightly better than Good but below Excellent. The intelligibility scores are very close between seen and unseen speakers (case 4 and case 5). It means that our baseline English model can produce almost equally highly intelligible speeches for seen and unseen speeches.

The synthesized foreign accented speeches (case 6, 7, 11, 12, 16 and 17) are low as expected. They are mostly rated between Poor and Fair. Only the synthesized accented English speeches spoken by Cantonese speakers (case 11) are rated slightly better than Fair but still not as good as native speeches. The synthesized foreign accented speech can easily confuse the meaning of the synthesized speeches with its accent.

It is observed that the synthesized foreign native speeches (case 8, 9, 13, 14, 18 and 19) are more intelligible and natural than the corresponding accented speeches spoken by the same speaker. However, most of them are not as good as the native speeches spoken by native speakers (case 10, 15 and 20) except that the case 18 and case 20 are rated very close to each other but case 20 are

Case No.	Case	Intelligibility		Naturalness		Speaker Similarity	
		Mean	95% CI	Mean	95% CI	Mean	95% CI
1	English	4.34	0.08	4.45	0.07	4.45	0.06
2	Cantonese	4.45	0.08	4.05	0.11	4.55	0.06
3	Mandarin	4.60	0.07	4.03	0.13	4.98	0.01

Table 6.1: The MOS mean and 95% confidence interval using t-distribution for ground truth utterances in three languages.

Case No.	Case	Intelligibility		Naturalness		Speaker Similarity	
		Mean	95% CI	Mean	95% CI	Mean	95% CI
4	Seen	4.07	0.23	3.80	0.18	3.78	0.38
5	Unseen	4.03	0.28	3.83	0.26	3.49	0.46

Table 6.2: The MOS mean and 95% confidence interval using t-distribution for English synthesized utterances for seen/unseen speakers using the mono-lingual baseline English model.

Case No.	Case	Intelligibility		Naturalness		Speaker Similarity	
		Mean	95% CI	Mean	95% CI	Mean	95% CI
6	CAE	2.05	0.27	2.13	0.17	3.07	0.39
7	CAM	1.70	0.15	1.54	0.09	2.96	0.69
8	CNE	3.28	0.23	2.92	0.23	2.91	0.38
9	CNM	3.84	0.20	3.51	0.21	3.00	0.51
10	CN	4.50	0.11	4.28	0.18	3.82	0.37
11	EAC	3.09	0.14	2.91	0.20	2.53	0.39
12	EAM	2.14	0.17	2.05	0.17	3.01	0.53
13	ENC	4.07	0.15	3.79	0.20	3.57	0.32
14	ENM	3.78	0.14	3.45	0.20	3.30	0.40
15	EN	4.54	0.07	4.30	0.13	3.84	0.37
16	MAC	2.26	0.21	2.12	0.35	2.45	0.44
17	MAE	2.57	0.20	2.30	0.19	3.39	0.35
18	MNC	4.43	0.09	4.24	0.12	3.28	0.36
19	MNE	4.17	0.15	3.70	0.15	3.20	0.33
20	MN	4.42	0.15	4.32	0.18	4.49	0.18

Table 6.3: MOS mean and 95% confidence interval using t-distribution for synthesized utterances in 5 accents in three languages using the proposed multi-lingual model for unseen speakers.

more natural than case 18. The reason may be explained in the t-SNE visualization of whitened x-vectors shown in Figure 5.3. We notice that even though the whitening may help to generate speeches with better speaker similarity, it doesn't change the fact that speaker embeddings of the speakers from different languages are in different clusters. It clearly indicates that the trained x-

vectors contain some language information which may perform as a certain bias during the training of the synthesizer. This phenomenon may suggest that the synthesizer is able to distinguish speakers from different languages.

Comparing the synthesized English speeches for unseen speakers between using the mono-lingual baseline English model and the proposed multi-lingual model (case 5 and case 15), we observe the synthesized speeches in case 15 using the proposed model is rated higher than those synthesized using the mono-lingual baseline English model. It shows that the proposed model can successfully generate high quality native speeches. The WaveNet paper [30] shows that the WaveNet trained with multi-speaker corpora can generate more natural speech than the WaveNet trained with single speaker corpus. We observe similar behaviour of the synthesizer: the proposed multi-lingual TTS system can outperform the baseline English TTS system when it is trained on multi-lingual corpora.

It is very interesting to observe that the synthesized native Cantonese speeches spoken by Mandarin speakers (case 9) are more intelligible and natural than those spoken by English speakers (case 8). Case 18 is also more intelligible and natural than case 19. The results seem to suggest that the Mandarin accents and Cantonese accents are closer to each other than they are to English accents. We hypothesize that both Mandarin and Cantonese are tonal languages, their prosody are closer than a non-tonal language English. We also observe that the case 6 is more intelligible and natural than case 7. Comparing the foreign accented and foreign native speeches, case 8 and case 9 are only different in speaker embedding inputs but case 6 and case 7 are also different in the tone/stress embedding inputs. Assuming English speakers will speak Cantonese or Mandarin as foreign languages with no stress, we input the tone/stress embedding representing ‘no stress’ in English when we synthesize speeches in case 6 and case 17. Similarly, we input the tone/stress embedding representing ‘tone one’ in Mandarin for synthesizing speeches in case 7 and case 12 for Mandarin speakers and the embedding representing ‘tone one’ in Cantonese for synthesizing speeches in case 11 and case 16 for Cantonese speakers. Since case 6 is better than case 7 in all measurements, we hypothesize that it is better to synthesize tonal Cantonese speech with no stress no tone than a wrong tone in Mandarin. We may derive the same hypothesis by comparing the synthesized Mandarin speeches in case 16, 17, 18 and 19. When we further compare the synthesized

English speech, we found case 13 is better than case 14. We hypothesize that Cantonese accents can be closer to English accents compared with Mandarin accents. Even though case 11 is better than case 12, the differences between them are larger than the differences between case 13 and 14. It might suggest that the Cantonese tone one is more closer to English pronunciations compared with Mandarin tone one.

It is also observed that some of the synthesized native speech are even rated higher than the ground truth speech such as case 10 and 15 in the table. One possible reason can be that the raters' opinion may be affected by the order of the presented samples.

6.4.2 Speaker Similarity

We first evaluate the speaker similarity among the ground truth utterances of the same speaker. The results show that the speaker similarity of the Mandarin ground truth speeches is much higher than that of Cantonese and English ground truth speeches. It shows that the speaker identity is more consistent among utterances of the same speaker in the Surfingtech corpus than the other corpora. It may explain why the speaker similarity of case 20 is much higher than those of case 10 and 15. However, this behaviour are not such significant in other synthesized speeches. It seems the cross-lingual speech synthesis will compromise the speaker identity because the MOS evaluation of the speaker similarity requires raters to compare two utterances in different languages. To verify this hypothesis, bi-lingual speech is required where the speaker similarity of speeches of same speaker in different languages can be evaluated.

Comparing synthesized native speeches (case 10, 15 and 20) and foreign native speeches (case 8, 9, 13, 14, 18 and 19), the speaker similarity of native speeches is much higher. We also observe similar patterns between the speaker similarity scores and the other two scores. When we compare two classes of speeches, the more intelligible and natural speeches usually perform better in speaker similarity. However, the differences between the pair of scores in speaker similarity are much smaller than intelligibility and naturalness scores. There are also a few exceptions. For example, the foreign accented speeches (case 6, 7 and 17) have very close or even better speaker similarity than the foreign native speech (case 8, 9 and 19) where the later should be more intelligible and natural. Higher intelligibility and naturalness MOSs do not always suggest that the speaker similarity MOS

is also higher (case 11 and 12).

CHAPTER 7

CONCLUSIONS

In conclusion, the proposed system can synthesize both intelligible and natural native speeches and foreign native speeches for unseen speakers in Cantonese, English and Mandarin with MOS results better than mono-lingual English baseline system. It proves that the transfer learning of x-vectors to TTS system is successful. It can also synthesize foreign accented speeches with foreign accents by manipulating the tone/stress embedding inputs. These accented speeches can confuse the meaning of the input utterance with foreign accents. The MOS results indicate that the speaker identity is well captured in the synthesized native speech. The slight differences between the foreign native speech and the native speech in MOS results seem to suggest that the TTS model captures the language information from speaker embeddings and generates speeches with the speaker's native accent. We hypothesize that the pretrained x-vectors on multi-lingual corpora contain language information. Although it may depend on the application whether language information should be kept in speaker embeddings, in our thesis, the language information in the speaker embeddings can differentiate the native speeches from native speakers and foreign speakers.

REFERENCES

- [1] Sercan Arik, Jitong Chen, Kainan Peng, Wei Ping, and Yanqi Zhou. Neural voice cloning with a few samples. In *Advances in Neural Information Processing Systems*, pages 10019–10029, 2018.
- [2] Sercan Ömer Arik, Mike Chrzanowski, Adam Coates, Greg Diamos, Andrew Gibiansky, Yongguo Kang, Xian Li, John Miller, Jonathan Raiman, Shubho Sengupta, and Mohammad Shoeybi. Deep Voice: Real-time neural text-to-speech. *CoRR*, abs/1702.07825, 2017.
- [3] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.
- [4] Niko Brümmer and Edward De Villiers. The speaker partitioning problem. In *Odyssey*, page 34, 2010.
- [5] Hui Bu, Jiayu Du, Xingyu Na, Bengu Wu, and Hao Zheng. Aishell-1: An open-source Mandarin speech corpus and a speech recognition baseline. In *the 20th Conference of the Oriental Chapter of the International Coordinating Committee on Speech Databases and Speech I/O Systems and Assessment (O-COCOSDA)*, pages 1–5. IEEE, 2017.
- [6] Mengnan Chen, Minchuan Chen, Shuang Liang, Jun Ma, Lei Chen, Shaojun Wang, and Jing Xiao. Cross-lingual, multi-speaker text-to-speech synthesis using neural speaker embedding. *Proc. Interspeech 2019*, pages 2105–2109, 2019.
- [7] Jan K Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio. Attention-based models for speech recognition. In *Advances in neural information processing systems*, pages 577–585, 2015.
- [8] Y. Fan, Y. Qian, F. K. Soong, and L. He. Multi-speaker modeling and speaker adaptation for DNN-based TTS synthesis, 2015. ID: 1.

- [9] Daniel Garcia-Romero and Carol Y Espy-Wilson. Analysis of i-vector length normalization in speaker recognition systems. In *Twelfth annual conference of the international speech communication association*, 2011.
- [10] Andrew Gibiansky, Sercan Arik, Gregory Diamos, John Miller, Kainan Peng, Wei Ping, Jonathan Raiman, and Yanqi Zhou. Deep Voice 2: Multi-speaker neural text-to-speech. In *Advances in neural information processing systems*, pages 2962–2970, 2017.
- [11] Daniel Griffin and Jae Lim. Signal estimation from modified short-time Fourier transform. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 32(2):236–243, 1984.
- [12] Sergey Ioffe. Probabilistic linear discriminant analysis. In *European Conference on Computer Vision*, pages 531–542. Springer, 2006.
- [13] Ye Jia, Yu Zhang, Ron Weiss, Quan Wang, Jonathan Shen, Fei Ren, Patrick Nguyen, Ruoming Pang, Ignacio Lopez Moreno, and Yonghui Wu. Transfer learning from speaker verification to multispeaker text-to-speech synthesis. In *Advances in neural information processing systems*, pages 4480–4490, 2018.
- [14] Patrick Kenny. Joint factor analysis of speaker and session variability: Theory and algorithms. *CRIM, Montreal, (Report) CRIM-06/08-13*, 14:28–29, 2005.
- [15] Patrick Kenny. Bayesian speaker verification with heavy-tailed priors. In *Odyssey*, page 14, 2010.
- [16] Tan Lee, Wai Kit Lo, P. C. Ching, and Helen Meng. Spoken language resources for Cantonese speech processing. *Speech Communication*, 36(3-4):327–342, 2002.
- [17] Michael McAuliffe, Michaela Socolof, Sarah Mihuc, Michael Wagner, and Morgan Sonderegger. Montreal forced aligner: Trainable text-speech alignment using kaldi. In *Interspeech*, pages 498–502, 2017.
- [18] Soroush Mehri, Kundan Kumar, Ishaan Gulrajani, Rithesh Kumar, Shubham Jain, Jose Sotelo, Aaron Courville, and Yoshua Bengio. SampleRNN: An unconditional end-to-end neural audio generation model. *arXiv preprint arXiv:1612.07837*, 2016.

- [19] Masanori Morise, Fumiya Yokomori, and Kenji Ozawa. WORLD: A vocoder-based high-quality speech synthesis system for real-time applications. *IEICE Transactions*, 99-D(7):1877–1884, 2016.
- [20] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: an ASR corpus based on public domain audio books. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5206–5210. IEEE, 2015.
- [21] Wei Ping, Kainan Peng, Andrew Gibiansky, Serkan O. Arik, Ajay Kannan, Sharan Narang, Jonathan Raiman, and John Miller. Deep Voice 3: Scaling text-to-speech with convolutional sequence learning. *arXiv preprint arXiv:1710.07654*, 2017.
- [22] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, and Petr Schwarz. The Kaldi speech recognition toolkit. In *IEEE 2011 workshop on automatic speech recognition and understanding*. IEEE Signal Processing Society, 2011.
- [23] Douglas A Reynolds, Thomas F Quatieri, and Robert B Dunn. Speaker verification using adapted Gaussian mixture models. *Digital signal processing*, 10(1-3):19–41, 2000.
- [24] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan, R. A. Saurous, Y. Agiomvrgiannakis, and Y. Wu. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4779–4783, April 2018.
- [25] David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur. X-vectors: Robust DNN embeddings for speaker recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5329–5333. IEEE, 2018.
- [26] Jose Sotelo, Soroush Mehri, Kundan Kumar, João Felipe Santos, Kyle Kastner, Aaron C. Courville, and Yoshua Bengio. Char2wav: End-to-end speech synthesis. In *the 5th International Conference on Learning Representations, ICLR 2017. Workshop Track Proceedings*, 2017.

- [27] D. E. Sturim and D. A. Reynolds. Speaker adaptive cohort selection for tnorm in text-independent speaker verification. In *Proceedings. (ICASSP '05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.*, volume 1, pages I/741–I/744 Vol. 1, March 2005.
- [28] ST-CMDS-20170001_1, Free ST Chinese Mandarin Corpus.
- [29] Yaniv Taigman, Lior Wolf, Adam Polyak, and Eliya Nachmani. Voiceloop: Voice fitting and synthesis via a phonological loop. *arXiv preprint arXiv:1707.06588*, 2017.
- [30] Aäron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew W. Senior, and Koray Kavukcuoglu. WaveNet: A generative model for raw audio. *CoRR*, abs/1609.03499, 2016.
- [31] E. Variani, X. Lei, E. McDermott, I. L. Moreno, and J. Gonzalez-Dominguez. Deep neural networks for small footprint text-dependent speaker verification. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4052–4056, May 2014.
- [32] Li Wan, Quan Wang, Alan Papir, and Ignacio Lopez Moreno. Generalized end-to-end loss for speaker verification. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4879–4883. IEEE, 2018.
- [33] Yuxuan Wang, R.J. Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J. Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, Quoc Le, Yannis Agiomyrgianakis, Rob Clark, and Rif A. Saurous. Tacotron: Towards end-to-end speech synthesis. In *Proc. Interspeech 2017*, pages 4006–4010, 2017.
- [34] Guoshen Yu, Stéphane Mallat, and Emmanuel Bacry. Audio denoising by time-frequency block thresholding. *IEEE Transactions on Signal processing*, 56(5):1830–1839, 2008.
- [35] Yu Zhang, Ron J. Weiss, Heiga Zen, Yonghui Wu, Zhifeng Chen, R. J. Skerry-Ryan, Ye Jia, Andrew Rosenberg, and Bhuvana Ramabhadran. Learning to speak fluently in a foreign

language: Multilingual speech synthesis and cross-language voice cloning. *arXiv preprint arXiv:1907.04448*, 2019.

APPENDIX A

PHONEME MAPPINGS

A.1 Pinyin to ARPABET

The Table A.1 shows the mappings from pinyin phonemes to ARPABET phonemes we have used in the thesis. We use the plus sign (+) to denote the delimiter between phonemes in a phoneme sequence.

Pinyin	ARPABET	Pinyin	ARPABET	Pinyin	ARPABET	Pinyin	ARPABET
b	B	ch	CH	ou	OW	iong	IY+OW+NG
p	P	sh	SH	an	AA+N	u	UW
m	M	r	ZH	en	EH+N	ua	UW+AA
f	F	z	Z	ang	AA+NG	uo	UW+AO
d	D	c	TH	eng	EH+NG	uai	UW+AY
t	T	s	S	ong	OW+NG	uei	UW+EY
n	N	y	Y	i	IY	uan	UW+AA+N
l	L	w	W	ia	IY+AA	uen	UW+EH+N
g	G	a	AA	ie	IY+EH	uang	UW+AA+NG
k	K	o	AO	iao	IY+AW	ueng	UW+EH+NG
h	H	e	EH	iou	IY+OW	ü	Y+UW
j	J_M	er	ER	ian	IY+AA+N	üe	Y+UW+EH
q	Q_M	ai	AY	in	IY+N	üan	Y+UW+AA+N
x	X_M	ei	EY	iang	IY+AA+NG	ün	Y+UW+N
zh	JH	ao	AW	ing	IY+NG		

Table A.1: The mapping table which maps the pinyin phonemes to the ARPABET phonemes where ‘j’, ‘q’ and ‘x’ are mapped to separate phonemes ‘J_M’, ‘Q_M’ and ‘X_M’ which are then concatenated to the ARPABET phoneme set.

A.2 Jyupting to ARPABET

The Table A.2 shows the mappings from Jyupting phonemes to ARPABET phonemes we have used in the thesis. Phonemes with the following finals: ‘m’, ‘n’, ‘ng’, ‘p’, ‘t’ and ‘k’, are treated as a sequence of vowels and finals and they are mapped to a sequence of ARPABET phonemes. For example, ‘aam’, ‘aan’, ‘aang’, ‘aap’, ‘aat’ and ‘aak’ phonemes are mapped to sequences ‘AA+M’, ‘AA+N’, ‘AA+NG’, ‘AA+P’, ‘AA+T’ and ‘AA+K’, respectively.

Jyupting	ARPABET	Jyupting	ARPABET
b	B	aa	AA
p	P	aai	AA+Y
m	M	aau	AA+W
f	F	ai	AH+Y
d	D	au	AH+W
t	T	e	EH
n	N	ei	EH+Y
l	L	eu	EH+W
g	G	i	IY
k	K	iu	IY+UW
ng	NG	o	AO
h	HH	oi	AO+Y
gw	G+W	ou	OW
kw	K+W	oe	ER
w	W	eo	ER+Y
z	JH	u	UW
c	CH	ui	UW+Y
s	S	yu	Y+UW
j	Y		

Table A.2: The mapping table which maps the Jyupting phonemes to the ARPABET phonemes.

A.3 ARPABET Phoneme set

The Table A.3 shows the 39 phonemes in ARPABET used by the CMU pronouncing dictionary ¹. We use the same ARPABET phoneme set in the thesis.

Phoneme	Example	Translation	Phoneme	Example	Translation
AA	odd	AA D	L	lee	L IY
AE	at	AE T	M	me	M IY
AH	hut	HH AH T	N	knee	N IY
AO	ought	AO T	NG	ping	P IH NG
AW	cow	K AW	OW	oat	OW T
AY	hide	HH AY D	OY	toy	T OY
B	be	B IY	P	pee	P IY
CH	cheese	CH IY Z	R	read	R IY D
D	dee	D IY	S	sea	S IY
DH	thee	DH IY	SH	she	SH IY
EH	Ed	EH D	T	tea	T IY
ER	hurt	HH ER T	TH	theta	TH EY T AH
EY	ate	EY T	UH	hood	HH UH D
F	fee	F IY	UW	two	T UW
G	green	G R IY N	V	vee	V IY
HH	he	HH IY	W	we	W IY
IH	it	IH T	Y	yield	Y IY L D
IY	eat	IY T	Z	zee	Z IY
JH	gee	JH IY	ZH	seizure	S IY ZH ER
K	key	K IY			

Table A.3: ARPABET phoneme set including 39 phonemes used by the CMU pronouncing dictionary.

¹<http://www.speech.cs.cmu.edu/cgi-bin/cmudict>