INTERSPEECH 2019
September 15–19, 2019, Graz, Austria

Cross-lingual, Multi-speaker Text-To-Speech Synthesis Using Neural Speaker Embedding

Mengnan Chen[1], *Minchuan Chen*[2], *Shuang Liang*[2], *Jun Ma*[2], *Lei Chen*[1], *Shaojun Wang*[2], *Jing Xiao*[2]

[1]East China Normal University
[2]Ping An Technology

{chenminchuan109, liangshuang161, wangshaojun851}@pingan.com.cn

Abstract

Neural network-based model for text-to-speech (TTS) synthesis has made significant progress in recent years. In this paper, we present a cross-lingual, multi-speaker neural end-to-end TTS framework which can model speaker characteristics and synthesize speech in different languages. We implement the model by introducing a separately trained neural speaker embedding network, which can represent the latent structure of different speakers and language pronunciations. We train the speech synthesis network bilingually and prove the possibility of synthesizing Chinese speaker's English speech and vice versa. We explore different methods to fit a new speaker using only a few speech samples. The experimental results show that, with only several minutes of audio from a new speaker, the proposed model can synthesize speech bilingually and acquire decent naturalness and similarity for both languages.

Index Terms: neural TTS, multi-speaker modeling, multi-language, speaker embedding

1. Introduction

In the past years, end-to-end speech synthesis system based on deep learning has made great progress such as Tacotron [1], Tacotron2 [2], DeepVoice3 [3], ClariNet [4] , Char2wav [5] and VoiceLoop [6]. Although the end-to-end TTS can generate natural speech which is close to humans, these models require a large amount of speech data from one speaker to obtain good quality. According to [7], it concludes that around 10 hours of speech-transcript pairs from one speaker are needed to get high quality by a neural end-to-end TTS model such as Tacotron. In order to support multiple speakers, we usually have to use tens of minutes of training data for every speaker, which make collecting high quality data a laborious work.

There are some studies focus on multi-speaker neural TTS modeling and most of these methods rely on a speaker embedding. DeepVoice2 [8] introduced a multi-speaker variation based on Tacotron which learned a low-dimensional embedding for each speaker. DeepVoice3 implemented a fully-convolutional sequence-to-sequence architecture and incorporated a position-augmented attention to support more than 2400 speakers with LibriSpeech dataset. However, both of them can only synthesize speech for observed speakers during training stage. Some approaches introduce a separate deep network to encode the speaker characteristics and embed the information into spectrogram predictor or vocoder. VoiceLoop [6] proposed a novel memory buffer mechanism to fit new speakers which are not seen during training, however, it needs tens of minutes of speech and transcript to obtain accepted quality. An extension work [9] of VoiceLoop employed an additional speaker encoding network and trained together with speech synthesis model, which can fit new voice by only using a few audio data. [10] and [11] explore different strategies to build multi-speaker TTS with few-shot adaptation. By comparing speaker adaptation method (fine-tuning a pre-trained multi-speaker model entirely or merely to the speaker embedding) and speaker encoding method (training a seperate model to predict the new speaker embedding with few data), they showed that both approaches can successfully adapt the multi-speaker neural network to a new speaker using just a small amount of speech data without transcript, while speaker adaptation substantially acquires more naturalness speech and better similarity. [12] used a speaker verification network as the speaker encoder, and concatenated the generated speaker embedding to each encoder time step. The speaker embedding is uniformly initialized, and then combined with the other parts of the model to train. The system gets a decent naturalness mean opinion score (MOS) by employing WaveNet as neural vocoder, but the speaker similarity MOS is comparatively low for unseen speaker during evaluation. This makes an obstacle to implement new speaker adaptation with a few utterances.

Cross-lingual TTS aims to build a system which can synthesize speech in a specific language not spoken by the target speaker. The technology can benefit various fields such as the speech translation system. In cross-lingual synthesis, text of one language is synthesized by the TTS system built for another language. There are mainly two ways to achieve this goal. The first is GMM-HMM-based method [13, 14, 15] by training two independent HMM based TTS systems from data recorded by a bilingual speaker, which were then used in a framework where the HMM states were shared across the two languages in decision-tree based clustering. A state mapping process is used to obtain mapping information and then apply to the target speaker for synthesizing speech with the target language. Another approach is based on unit selection such as phoneme-level [16] or frame-level selection [17]. In this approach, the source language frames are mapped to the closest target language frames and the mapping is based on minimizing the distance between speech feature vectors. Due to language mismatch, the main problem for cross-lingual TTS is the quality of synthesized speech is not natural enough.

In this paper, we investigate the cross-lingual, multi-speaker neural TTS in two different aspects that has not been fully explored by previous work. One is that we further discuss how to use limited amount of data to achieve multi-speaker TTS. We explore solutions to improve the naturalness and similarity of the generated speech. Secondly, we analyze end-to-end models in cross-lingual setting. In previous work, the multi-speaker neural TTS is mostly concentrated on the same language, and only one language speech could be synthesized since it's not easy to collect hours of speech as well as finding a person who can speak multiple languages. In our work, we extract the speakers' voice characteristics across languages and enable an end-to-end speech synthesis system to support multiple languages. We train a multi-speaker TTS network bilin-

http://dx.doi.org/10.21437/Interspeech.2019-1632

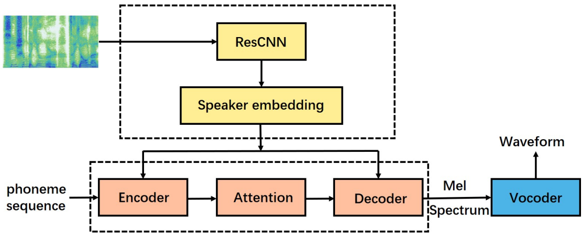

Figure 1: *System architecture.*

gually with monolingual speakers' dataset, although there is no speaker who speaks both English and Chinese, we can synthesize English audio spoken by Chinese speaker and vise versa with the proposed model.

The rest of the paper is organized as follows: Section 2 describes the architecture of cross-lingual multi-speaker neural TTS system. Section 3 presents the experiments and subjective evaluation results. We make a brief conclusion in Section 4.

2. System Architecture

Our system consists of three parts, illustrated in Figure 1: (1) a speaker encoder converts speech into speaker embedding, (2) a sequence-to-sequence network that converts phoneme sequence to mel-spectrogram which is conditioned on speaker embedding and (3) a vocoder that transforms the mel-spectrogram to audio.

2.1. Neural Speaker Encoder

The speaker encoder extracts acoustic features to produce utterance-level speaker embedding, which is used to condition mel-spectrogram generation network on the desired speaker' voice signal. The structure of the speaker encoder follows the ResCNN in [18] as depicted in Figure 2, which captures the characteristics of speech across both speakers and languages. In this work, the filter size of convolution layers is 64, 128, 256, and 512, respectively. Raw audio is first preprocessed and then fed into the ResBlock. The activations with variable length are averaged along time dimension with a temporal average layer. The pooling layer extract remarkable features and then performs an affine transformation. The affine transformation layer is used to project the utterance-level representation into a 256-dimension embedding. The output is regularized by length normalization to generate speaker embedding which captures speaker characteristics and language pronunciations from the latent space.

We firstly train the speaker embedding network separately on a speaker verification task with softmax loss, and then finetune the whole model with triplet loss [19] which maps speech into a feature space where the distances correspond to speaker similarity. The triple loss takes three samples as input: an anchor, a positive example, and a negative example as shown in Figure 3. During training stage, the model operates on pairs of speaker embeddings by maximizing the cosine similarities of embedding pairs from the same speaker (anchor and positive example), and minimizing those from different speakers (anchor and negative example). That is:

$$Loss = \sum_{i=1}^{N} [\|f_i^a - f_i^p\|_2^2 - \|f_i^a - f_i^n\|_2^2 + \alpha]_+ \qquad (1)$$

where f_i^a, f_i^p and f_i^n are anchor, positive and negative speaker embedding, $\alpha \geq 0$ is a constant[19]. As mentioned in [18], the method transfers well across spoken languages which are vastly different, so it's suitable for multiple languages transformation and can model speaker's voice bilingually, such as Chinese and English.

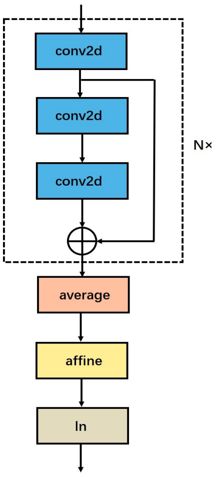

Figure 2: *Speaker encoder. "average" denotes average the variable length tensor along the time dimension, "ln" denotes length normalization.*

2.2. Mel-Spectrogram Generation Network

The mel-spectrogram generation network is similar to Tacotron2 [2], a sequence-to-sequence model with attention mechanism which predicts the corresponding mel-spectrogram with a sequence of character/phoneme as an input. The speaker embedding is served as a condition to the spectrogram predictor with specific speaker characteristics. An embedding vector of the speaker is concatenated with the mel-spectrogram generation network by different strategies (described in Section 3.3.2). In our work, four approaches were experimented to embed the speaker feature vectors into the neural network.

2.3. Vocoder

We used Griffin-Lim [20] to transform the predicted mel-spectrogram into the corresponding audio. It's an inversion algorithm converts spectrograms to time-domain waveforms by iteratively estimating the unknown phases. In our work, the number of iterations is set to 60 to obtain acceptable quality.

3. Experiments

We first describe the datasets used in our experiment. Then we introduce each part of the proposed model seperately and discuss different training stratagies. The subjective evaluation result is presented at last.

3.1. Dataset

We use two public datasets of English and Chinese for training the mel-spectrogram generation network. For English, we use VCTK corpus [21] which contains 44 hours of clean audio from 109 speakers with various accents. Each speaker recorded 400 sentences. A subset of Free ST Chinese Mandarin Corpus [22] is used for Chinese dataset, which contains 855 speakers and recorded in silent environment, each speaker includes 120 sentences. We also use the CMU Arctic dataset [23] which includes 7 speakers with different accents and an internal Chinese dataset includes 7 speakers with high quality speech of Mandarine for validation and testing. We downsample the audio to 16kHz and trim leading and tailing silence.

We split these datasets for training and testing: 337 Chinese speakers and 109 English speakers are used for training the multi-speaker model, 8 Chinese speakers and 8 English speakers are used for validation. We choose 2 Chinese, 2 English speakers for testing (Seen) and 3 Chinese, 3 English speakers for new speaker adaptation (Unseen), all have similar distribution with training dataset in terms of gender and accent.

We use the epitran IPA library [24] to convert English and Chinese transcripts to International Phonetic Alphabet (IPA), which improves the pronunciation accuracy and unifies phonetic transcriptions of different languages.

3.2. Training Cross-lingual, Multi-speaker TTS

In our proposed method, the speaker encoder and the mel-spectrogram generation network can be trained in parallel. Our speaker encoder follows the model structure in [18] and the two monolingual datasets are used for training. The speaker encoder is trained in two stages: Firstly, we pre-train the speaker encoder for 10 epochs with softmax loss and using a minibatch size of 32 as it converge to an approximate local optimal point, then the model is fine-tuned with triple loss for 10 epochs using a minibatch size of 64. The loss of the pre-trained model reduced substantially which improves the similarity MOS of the entire multi-speaker TTS.

For mel-spectrogram generation network, we follow the specifications mentioned in Tacotron2 [2] and minimize the ground truth mel-spectrogram and the predicted mel-spectrogram with L2 loss. We normalize the mel-spectrograms to [-4, 4] in preprocess in order to reduce blurriness in synthesized audio. In order to increase the accuracy of stop-token prediction, we assign a weighted penalty 20 for misclassification. We use the Adam optimizer with learning rate decay, which starts from 1e-3 and is reduced to 1e-5 after 50k steps. The network is trained with a batch size of 32 with an Nvidia V100 GPU. We uses the L2 regularization to improve the model's generalization ability.

We use grapheme-to-phoneme (G2P) library to convert transcripts to the corresponding phoneme sequence. The Chinese and English pronunciations can be identified with a unified phonetic system. For Chinese text, we first perform word segmentation which separates words and phrases with specific symbols in order to improve speech fluency. The phoneme sequences are fed to the encoder of mel-spectrogram generation network as input. The mel-spectrogram generation network is conditioned on the speaker embedding produced by the speaker encoder with different strategies.

3.3. Model Implementation

3.3.1. Speaker encoder

We first train the model with English (VCTK corpus) dataset only. In order to enable the speaker encoder to capture the speaker characteristics across languages, then we mix Chinese (ST corpus) with English (VCTK corpus) data to build a bilingual dataset for training. The selected audios which are all clean enough and have many variants of accent.

As shown in Figure 4, the different training sets have significant impact on speaker embedding. In our experiment, speakers of different languages can be magnificently separated in latent space after 65k training steps. We observe that the learned speaker embedding can represent the relation between pronunciations across the two languages, although English and Chinese have different phoneme sets and English is a stress language while Mandarin is a tonal language. We perceive the sounds were properly approximated from one language to another by means of the similarity between the articulatory features of the two languages. We also find that for the bilingual speaker voice generated by our model, it imposes the effect of his/her mother tongue while speaking another language, when synthesizing an English speaker's Chinese, the speech has an English accent as called first language accent effect. We observe the phonemes with similar pronunciation are inclined to stay closer than the others across the two languages. Although some phoneme-to-phoneme mappings are not exactly match between English and Chinese phoneme sets in terms of IPA, phonemes of one language are mapped to the closest sounding of the primary language depending on the training dataset.

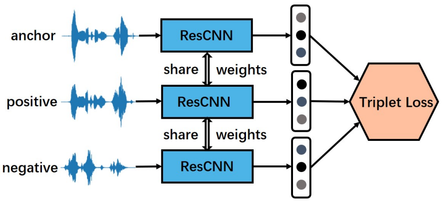

Figure 3: *Triplet Loss. The training goal is to maximize the distance of embeddings from the same speaker, and pull away the speaker embeddings of different speakers as far as possible.*

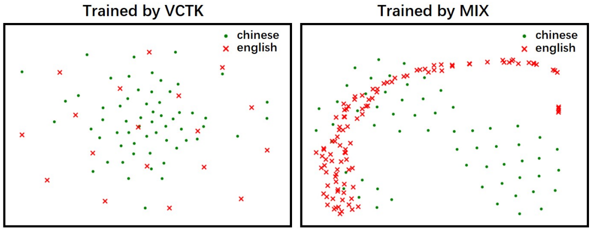

Figure 4: *t-SNE visualization of speaker embeddings extracted by speaker encoder with respect to VCTK and MIX.*

3.3.2. Condition on speaker embedding

We try four ways to condition the mel-spectrogram generation model on speaker embedding: (1). Concatenate speaker embedding to each time step of the encoder; (2). Add an affine transformation to speaker embedding, then splicing to each time step of the encoder; (3). Initialize the encoder with the speaker embedding; (4). Initialize the decoder with speaker embedding.

Our experiments show that the timbre of the synthesized speech is inconsistent if the speaker embedding is simply concatenated to each time step of the encoder. The voice generated by the same speaker embedding is sometimes changeable, but it's significantly improved by the other three methods. We then initialize the encoder with speaker embedding, the synthesized audio has obvious background noise, which leads to bad voice quality. This may be caused by the mismatch of speaker embedding which represents acoustic feature while the encoder of Tacotron is fed linguistic sequences as input. Next, we add an affine transformation to the speaker embedding and concatenated to every time step of the encoder, it really stabilizes the voice and make the speech more fluent. Lastly, we initialize the decoder with speaker embedding which improves generalization and acquires better audio quality. In our final model, we combine the method (2) and (4): the speaker embedding after an affine transformation then concatenated to each time step of the encoder as well as initialize the decoder.

3.3.3. Pre-training decoder

In training stage, the amount of training data for each speaker is comparatively small in our multi-speaker TTS. In order to improve the audio quality and ease the requirement for paired (audio, text) training dataset, we follow the training strategy as mentioned in [7], which pre-trains the Tacotron2 decoder seperately. During pre-training, the goal of the decoder is to predict the next frame for learning the acoustic information of mel-spectrogram with teacher forcing. This alleviates the workload of the decoder and leads to faster convergence for the entire model. In our experiment, after pre-training the decoder with the mixed data, we trained the whole model with VCTK corpus and the mixed dataset, respectively.

3.4. Evaluation of Synthesized Speech

We evaluate the quality of the generated samples by conducting MOS text using the crowdMOS framework [25]. In our test, we use 48 utterances to evaluate naturalness and 40 utterances to evaluate speaker similarity[1].

3.4.1. Speech naturalness

Each utterance is listened by at least 20 subjects whose first language is Chinese and are well educated in English. The subjects are asked to rate the naturalness of generated utterances on a five-point Likert scale (1:Bad, 2:Poor, 3:Fair, 4:Good, 5:Excellent). We construct an evaluation set of 800 phrases which do not appear in the training sets and randomly select sentences for testing. This testing set is divided to two subsets: one consists of speakers included in the training set (Seen), the other consists of those held out (Unseen). The sets are constructed of 4 Seen (2 English, 2 Chinese) and 6 Unseen (3 English, 3 Chinese) speakers. For Seen speakers, we randomly choose one utterance to compute the speaker embedding. For Unseen speakers, we use about 3 minutes (30 utterances) audio for prediction. The results are shown in Table 1, the MOS score of the multi-speaker TTS model is 3.76 for Seen speakers and 3.60 for Unseen speakers. Since we used Griffin-Lim vocoder, the naturalness MOS is relatively low comparing with other work [12]. It's helpful to get more natural synthesized voice with neural vocoder such as WaveNet, WaveGlow etc.

Table 1: *Speech naturalness Mean Opinion Score (MOS) with 95% confidence intervals.*

Evaluation Type	**MOS**
Ground truth	4.511± 0.33
Seen	3.762± 0.458
Unseen	3.601± 0.427

3.4.2. Speech similarity

In similarity test, a subject is presented with a pair of utterances comprises a real utterance recorded by a speaker and another real or synthesized utterance from the same speaker. The similarity MOS test uses five-scale-score for evaluation (1: Not at all similar, 2: Slightly similar, 3: Moderately similar, 4: Very similar, 5: Extremely similar). As shown in Table 2, the similarity MOS for Seen and Unseen speakers are both above 3.4 and very closely, which demonstrates the model can primely generalize to the new speakers.

Table 2: *Speech similarity Mean Opinion Score (MOS) with 95% confidence intervals.*

Evaluation Type	**MOS**
Ground truth	4.788± 0.254
Seen	3.418± 0.449
Unseen	3.453± 0.39

3.4.3. Cross-lingual synthesis

To evaluate the ability of Cross-lingual synthesizing, we choose 4 speakers (2 English and 2 Chinese) in training data and 4 new speakers (2 English and 2 Chinese) for validation. English sentences were synthesized by the speaker who spoken Chinese and vice versa. Table 3 shows the result, which indicates that the speaker embbedings properly represent the pronunciations across the two languages.

Table 3: *Speech naturalness and similarity Mean Opinion Score (MOS) of cross-lingual speakers with 95% confidence intervals.*

Evaluation Type	**MOS**
Naturalness-Seen	3.590± 0.432
Similarity-Seen	3.418± 0.389
Naturalness-Unseen	3.312± 0.363
Similarity-Unseen	3.165± 0.459

As each speaker has extremely limited monolingual data and with a simple vocoder, the generated audio results lower voice quality comparing with single speaker model. It can be improved by conditioning the synthesizer on independent speaker with more high quality data and using neural vocoder. The accent of different language speakers has a certain impact and one solution is to add accent embeddings for languages. We leave this as our future work.

4. Conclusions

In this paper, we present a cross-lingual multi-speaker TTS to model speech across languages. This model combines a seperately trained speaker encoder network with a neural TTS synthesis network and a Griffin-Lim vocoder. The model is able to generate decent quality speech for both speakers seen during training and speakers never seen before. Our result shows that the multi-speaker TTS model can extract the speaker characteristics as well as language pronunciations with speaker embbeding from the latent space. This model can generate speech of arbitrary utterance in various speaker's voice. We also verified that with small amount of audio data, our proposed approach can well handle cross-lingual tasks.

[1] Audio samples: `https://cnlinxi.github.io/speech_demo/publications/cross_multi_tts`

5. References

[1] Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio *et al.*, "Tacotron: Towards end-to-end speech synthesis," *Interspeech*, pp. 4006–4010, 2017.

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

[3] W. Ping, K. Peng, A. Gibiansky, S. O. Arik, A. Kannan, S. Narang, J. Raiman, and J. Miller, "Deep voice 3: Scaling text-to-speech with convolutional sequence learning," in *International Conference of Learning Representation (ICLR)*, 2018.

[4] W. Ping, K. Peng, and J. Chen, "Clarinet: Parallel wave generation in end-to-end text-to-speech," in *International Conference of Learning Representation (ICLR)*, 2019.

[5] J. Sotelo, S. Mehri, K. Kumar, J. F. Santos, K. Kastner, A. Courville, and Y. Bengio, "Char2wav: End-to-end speech synthesis," 2017.

[6] Y. Taigman, L. Wolf, A. Polyak, and E. Nachmani, "Voiceloop: Voice fitting and synthesis via a phonological loop," *arXiv preprint arXiv:1707.06588*, 2017.

[7] Y. Chung, Y. Wang, W. Hsu, Y. Zhang, and R. Skerry Ryan, "Semi-supervised training for improving data efficiency in end-to-end speech synthesis," *arXiv preprint arXiv:1808.10128*, 2018.

[8] A. Gibiansky, S. Arik, G. Diamos, J. Miller, K. Peng, W. Ping, J. Raiman, and Y. Zhou, "Deep voice 2: Multi-speaker neural text-to-speech," in *Advances in Neural Information Processing Systems*, 2017, pp. 2962–2970.

[9] E. Nachmani, A. Polyak, Y. Taigman, and L. Wolf, "Fitting new speakers based on a short untranscribed sample," in *International Conference on Machine Learning*, 2018, pp. 3680–3688.

[10] S. Arik, J. Chen, K. Peng, W. Ping, and Y. Zhou, "Neural voice cloning with a few samples," in *Advances in Neural Information Processing Systems*, 2018, pp. 10 040–10 050.

[11] Y. Chen, Y. Assael, B. Shillingford, D. Budden, S. Reed, H. Zen, Q. Wang, L. C. Cobo, A. Trask, B. Laurie *et al.*, "Sample efficient adaptive text-to-speech," *arXiv preprint arXiv:1809.10460*, 2018.

[12] Y. Jia, Y. Zhang, R. Weiss, Q. Wang, J. Shen, F. Ren, P. Nguyen, R. Pang, I. L. Moreno, Y. Wu *et al.*, "Transfer learning from speaker verification to multispeaker text-to-speech synthesis," in *Advances in Neural Information Processing Systems*, 2018, pp. 4485–4495.

[13] Y. Qian, H. Liang, and F. K. Soong, "A cross-language state sharing and mapping approach to bilingual (Mandarin–English) TTS," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 6, pp. 1231–1239, 2009.

[14] F. Xie, F. K. Soong, and H. Li, "A KL divergence and DNN approach to cross-lingual TTS," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 5515–5519.

[15] H. Wang, F. Soong, and H. Meng, "A spectral space warping approach to cross-lingual voice transformation in HMM-based TTS," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 4874–4878.

[16] B. Leonardo, B. Claudia, and Q. Silvia, "Language independent phoneme mapping for foreign TTS," in *Proc. of 5th ISCA Tutorial and Research Workshop on Speech Synthesis*, 2004.

[17] Y. Qian, J. Xu, and F. K. Soong, "A frame mapping based hmm approach to cross-lingual voice transformation," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2011, pp. 5120–5123.

[18] C. Li, X. Ma, B. Jiang, X. Li, X. Zhang, X. Liu, Y. Cao, A. Kannan, and Z. Zhu, "Deep speaker: an end-to-end neural speaker embedding system," *arXiv preprint arXiv:1705.02304*, 2017.

[19] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 815–823.

[20] D. Griffin and J. Lim, "Signal estimation from modified short-time fourier transform," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 32, no. 2, pp. 236–243, 1984.

[21] C. Veaux, J. Yamagishi, K. MacDonald *et al.*, "SUPERSEDED-CSTR VCTK corpus: English multi-speaker corpus for cstr voice cloning toolkit," 2016.

[22] surfing.ai, "St-cmds-20170001_1, free ST Chinese Mandarin corpus," 2017.

[23] J. Kominek and A. W. Black, "The CMU Arctic speech databases," in *Fifth ISCA workshop on speech synthesis*, 2004.

[24] D. R. Mortensen, S. Dalmia, and P. Littell, "Epitran: Precision G2P for many languages," in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*, 2018.

[25] F. Ribeiro, D. Florêncio, C. Zhang, and M. Seltzer, "Crowdmos: An approach for crowdsourcing mean opinion score studies," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2011, pp. 2416–2419.