END-TO-END ACCENT CONVERSION WITHOUT USING NATIVE UTTERANCES

Songxiang Liu[1]*, Disong Wang*[1]*, Yuewen Cao*[1]*, Lifa Sun*[3]*, Xixin Wu*[1]*, Shiyin Kang*[4]
Zhiyong Wu[1,2,*]*, Xunying Liu*[1]*, Dan Su*[4]*, Dong Yu*[4]*, Helen Meng*[1]

[1]Human-Computer Communications Laboratory,
The Chinese University of Hong Kong, Hong Kong SAR, China
[2]Tsinghua-CUHK Joint Research Center for Media Sciences, Technologies and Systems,
Graduate School at Shenzhen, Tsinghua University, Shenzhen, China
[3]SpeechX Limited, Shenzhen, China
[4]Tencent AI Lab, Tencent, Shenzhen, China
{sxliu,dswang,ywcao,lfsun,wuxx,zywu,xyliu,hmmeng}@se.cuhk.edu.hk,
{shiyinkang,dansu,dyu}@tencent.com

ABSTRACT

Techniques for accent conversion (AC) aim to convert non-native to native accented speech. Conventional AC methods try to convert only the speaker identity of a native speaker's voice to that of the non-native accented target speaker, leaving the underlying content and pronunciations unchanged. This hinders their practical use in real-world applications, because native-accented utterances are required at conversion stage. In this paper, we present an end-to-end framework, which is able to conduct AC from non-native-accented utterances without using any native-accented utterances during online conversion. We achieve this by independently extracting linguistic and speaker representations from non-native accented speech and condition a speech synthesis model on these representations to generate native-accented speech. Experiments on open-source data corpora show that the proposed system can convert Hindi-accented English speech into native American English speech with high naturalness, which is indistinguishable from native-accented recordings in terms of accent.

Index Terms— Accent conversion, speech synthesis, accented speech recognition

1. INTRODUCTION

Accent conversion (AC) [1, 2] aims to transform non-native speech to sound as if the speaker had a native accent. Learners who acquire a second language (L2) usually speak with a non-native accent because of the influence of their mother tongues. It would be beneficial for L2 learners to be able to listen to native-accented speech with their own voices [2].

Conventional AC methods try to convert only the speaker identity of a native speaker's voice to that of the non-native accented target speaker, leaving the underlying content and pronunciations unchanged. This hinders their practical use in real-world applications, because native-accented utterances are required at the conversion stage. To solve this issue, we propose an end-to-end AC approach, which is able to conduct AC from non-native-accented utterances without using any native-accented utterance at the conversion stage. The proposed approach contains four parts: a speaker encoder, a sequence-to-sequence (seq2seq) multi-speaker text-to-speech (TTS) synthesis model, a seq2seq accented automatic speech recognition (ASR) model and a neural vocoder. The speaker encoder, which is trained on a speaker verification (SV) task, generates fixed-dimensional speaker embedding vectors from acoustic features. The multi-speaker TTS model is based on Tacotron2 [3], which generates mel-spectrogram features from a phoneme sequence, conditioned on the speaker embedding. The accented ASR model predicts linguistic representations from acoustic features, conditioned on accent embedding. The TTS model is trained using only native English speech data, while the accented ASR model is trained using both the native and non-native English speech data. During accent conversion, acoustic features from the non-native speech are fed into the accented ASR model and the speaker encoder, generating linguistic representations and the speaker embedding respectively. The generated linguistic representations and the speaker embedding are then used by the TTS decoder to generate the native-accented acoustic features. The neural vocoder finally convert the acoustic features into time-domain waveforms.

The contributions of this paper include: (1) To the best of our knowledge, the proposed approach is the first AC technique which is able to convert non-native-accented into native-accented speech directly during the conversion stage, without using any native utterance. (2) The proposed approach is based on end-to-end seq2seq networks, which has the ability to model prosodic characteristics, e.g., speaking rate and duration, making the converted speech output sound more native. (3) The proposed approach has no requirement for parallel speech data between the native and non-native speakers for training.

2. RELATED WORK

Various approaches for AC have been proposed. Early approaches combine spectral features from native and non-native speakers to control the degree of accent using voice morphing [4–6]. Voice conversion (VC) techniques [7] are adapted to match native and non-native frames based on their MFCC similarity after vocal tract length normalization (VTLN) [1]. Phonetic posteriorgrams (PPGs), which are successfully used for VC tasks [8–10], have also been used for AC [11, 12]. These approaches can reduce the accent of non-native

*Corresponding author

utterances, but have various limitations. Voice morphing methods tend to generate converted speech perceived as neither the native speaker nor the non-native speaker in timbre. The method proposed in [1] requires relatively large set of parallel recordings from the native and non-native speakers and VLTN only accounts for a subset of the speaker characteristics, which leads to limited AC performance. Since the nature of frame-wise feature mapping in the aforementioned approaches, prosodic patterns, such as duration and speaking rate, are hard to convert. Moreover, all previous AC techniques require native reference utterances during the accent conversion phase.

The work presented here is mostly inspired by the recently proposed non-parallel seq2seq VC technique [13], neural voice cloning technique [14] and accented ASR technique [15]. Zhang et al. [13] proposed a seq2seq-based VC framework using non-parallel training data, where a recognition encoder and a speaker encoder are jointly trained to extracted disentangled linguistic and speaker representations from acoustic features. Different from their method, in this paper we train a stand-alone speaker encoder using a speaker discriminative loss [16] to generate speaker representations. We expect the speaker encoder to learn a representation which captures speaker characteristics relevant to speech synthesis. Inspired by [15] and [17], we adopt an end-to-end accented ASR model trained using a multi-task loss to extract accent-agnostic linguistic representations from the non-native-accented speech. Following the setting in [14], we extend the Tacotron2 architecture [3] to support multiple speakers.

3. BASELINE APPROACH

During training, we first train a speaker-independent ASR (SI-ASR) model using a native English multi-speaker corpus. We then use the SI-ASR model to compute L1 PPGs from speech utterances of native-accented speakers and L2 PPGs from speech utterances of the non-native-accented speaker, respectively. Dynamic Time Warping (DTW) is used to align the paralleled L2 PPGs and L1 PPGs. As shown in Fig. 1, we then train two transform models, where transform 1 maps L2 PPGs to L1 PPGs while transform 2 maps L2 PPGs to mel spectrograms. During accent conversion, as shown in Fig. 2, we first compute L2 PPGs from the non-native-accented utterance, and then feed the L2 PPGs into transform 1 and transform 2 to get the converted mel sepctrograms. A WaveRNN-based neural vocoder [18] is used to convert the mel spectrograms into waveforms.

4. PROPOSED APPROACH

The proposed end-to-end AC approach is composed of four independently trained neural networks: a speaker encoder, a multi-speaker TTS model, an accented ASR model (as shown in Fig. 3) and a neural vocoder. (The neural vocoder is not shown because of space limitation.)

4.1. Speaker encoder

The speaker encoder model used in this paper follows [16], which is a scalable and accurate neural network framework for speaker verification. It generates a fixed-dimensional speaker embedding vector from a sequence of acoustic frames computed from a speech utterance of arbitrary length. The speaker embedding vector is used to condition the TTS model on a reference speech signal from a desired target speaker, so that the generated speech has speaker identity of that target speaker. The speaker encoder is trained to optimize a generalized end-to-end (GE2E) speaker verification loss [16], so

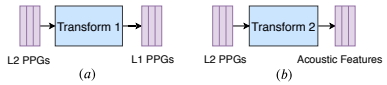

Fig. 1. Training stage of the baseline approach.

Fig. 2. Conversion stage of the baseline approach.

that embeddings of utterances from the same speaker have high cosine similarity, while those of utterances from different speakers are far apart in the embedding space. We expect the speaker encoder to learn a representation relevant to speech synthesis, which captures speaker characteristics of a non-native-accented speaker who is unseen during training.

4.2. Multi-speaker TTS model

Inspired by [14], we modify the Tacotron2 model, which is an attention-based encoder-decoder model, to support multiple speakers. As shown in Fig. 3, a speaker embedding vector computed by the speaker encoder for a desired target speaker is concatenated with the TTS encoder output at each time step. The TTS decoder with attention takes these as additional inputs to generate mel spectrograms. The TTS model is trained with the native-accented speech and corresponding text transcripts by using the mean square error (MSE) loss L_{TTS}, such that it is expected to only generate native-accented speech with the identity/timbre determined by the speaker embedding. We map the text transcripts into phoneme sequences as the input of the TTS model, since [14] has shown that using phoneme sequences leads to faster convergence and improved pronunciation of rare words and proper nouns.

4.3. Multi-task accented ASR model

We use an accented ASR model to learn accent-agnostic linguistic representations from acoustic features. The ASR model applies an end-to-end attention-based encoder-decoder framework [19]. Given a pair of audio and its phoneme transcriptions, we compute acoustic features from the audio, and linguistic representations from the phoneme sequence by the TTS encoder. Inspired by [17], we add a fully connected (FC) transform layer on top of the ASR encoder and compute a connectionist temporal classification (CTC) loss L_{CTC} to stabilize the training process. Since the training data of the ASR model includes accented utterances, following [15], we concatenated an accent embedding with acoustic features at each frame as inputs to the ASR model and add an accent classifier on top of the ASR encoder to make it more robust for accented speech recognition. We postulate that different accents are associated with different speaker. In this paper, the accent embedding of a speaker is obtained by averaging all his/her speaker embeddings. The output of the accent classifier is used to compute a cross-entropy loss L_{ACC}. The attention-based ASR decoder is adopted to predict phoneme labels and linguistic representations in two streams. A cross-entropy loss L_{CE} is used for phoneme label prediction while a MSE loss L_{TTSE} which measures the linguistic difference between TTS-encoder output $\mathbf{H}^l$ and ASR-decoder output $\widehat{\mathbf{H}^l}$ is used for linguistic representation pre-

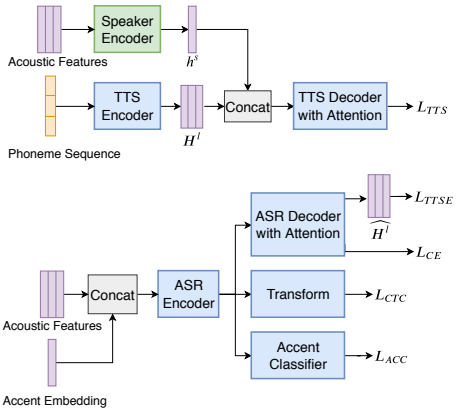

Fig. 3. Training stage of the proposed approach. $\mathbf{h}^s$ is speaker representation. $\mathbf{H}^l$ is TTS-encoder linguistic representation while $\widehat{\mathbf{H}}^l$ is ASR-decoder linguistic representation.

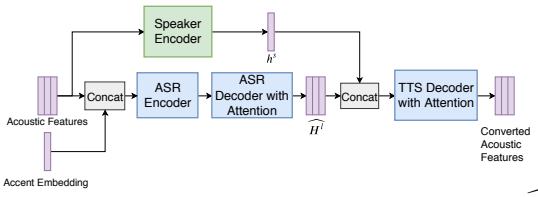

Fig. 4. Conversion stage of the proposed approach. $\mathbf{h}^s$ and $\widehat{\mathbf{H}}^l$ are speaker and linguistic representations, respectively.

diction. L_{TTSE} has the following form:

$$L_{TTSE} = \frac{1}{N}\sum_{i=1}^{N} ||\mathbf{H}_i^l - \widehat{\mathbf{H}}_i^l||_2, \quad (1)$$

where N is the number of training samples. The ASR model is trained with a multi-task loss:

$$L_{ASR} = \lambda_1 L_{CE} + \lambda_2 L_{TTSE} + \lambda_3 L_{CTC} + \lambda_4 L_{ACC}, \quad (2)$$

where the λ's are hyper-parameters, weighting the four losses.

4.4. Neural vocoder

In this paper, the WaveRNN network is used as the neural vocoder. We use the open-sourced Pytorch implementation[1]. Since the mel spectrogram captures all of the relevant details needed for high quality speech synthesis, we simply use ground-truth mel spectrograms from multiple speakers to train the WaveRNN, without adding any speaker embedding.

4.5. Training and conversion

At training stage, as shown in Fig. 3, we first train the speaker encoder model. We then train the multi-speaker TTS model using only native English speech data with the loss L_{TTS}. After that, the accented ASR model is first pre-trained using speech data from multiple native-accented speakers and one non-native-accented target speaker. Then the ASR model is fine-tuned using speech data from only the non-native-accented target speaker. We use the loss L_{ASR} in Eq. 2 during these two stages. We train a WaveRNN using speech data from only native-accented speakers.

[1]https://github.com/fatchord/WaveRNN

At accent conversion stage, as shown in Fig. 4, acoustic features are first computed from the non-native accented utterances. The speaker encoder then takes in the acoustic features and output the speaker embedding vectors representing the identity of the non-native-accented speaker. Accent embedding is the averaged speaker embeddings of the non-native-accented speaker. We concatenate the accent embedding with acoustic features at each frame and then feed them into the ASR model to generate linguistic representations $\widehat{\mathbf{H}}^l$. The attention-based TTS decoder then takes in the linguistic representations and speaker embedding to generate native-accented acoustic features. Finally, we use the WaveRNN model to convert the acoustic features into time domain waveform, which is expected to be more native-accented.

5. EXPERIMENTS

5.1. Experimental setup

We use LibriSpeech (train-other-500) [20], VoxCeleb1 [21] and VoxCeleb2 [22] datasets to train the speaker encoder. In total, there are 8.4K speakers. The inputs to the speaker encoder are 40-channel log mel spectrograms with 25ms window width and 10ms frame-shift.

The VCTK dataset [23] contains 44 hours of clean speech from 109 speakers. In this paper, we choose the Hindi-accented speaker p248 as the target speaker. We regard the speakers having no Hindi accent as native English speakers. Audios are re-sampled to 22.05 kHz. For training of the TTS model and the WaveRNN model, we only use speech data from 105 native speakers. The mel spectrograms have 80 channels, computed using a 50ms window width and 12.5ms frame-shift. We randomly select 1000 samples as validation set and the remaining samples are used for training. For training of the accented ASR model, speech data of the native speakers and p248 is used. 40-channel mel spectrograms computed using 25ms window width and 10ms shift as well as their delta and delta-delta features are used as acoustic features. The number of utterances from p248 used for training, validation and testing are 326, 25 and 25, respectively. The SI-ASR model used to extract PPGs is trained with the TIMIT dataset [24] as in [25].

The speaker encoder is a 3-layer LSTM with 256 hidden nodes followed by a projection layer of 256 units. The output is the L2-normalized hidden state of the last layer, which is a vector of 256 elements. The TTS model employs the same architecture as in [3]. The ASR encoder is a 5-layer bidirectional LSTM (BLSTM) with 320 units per direction. 300-dimensional location-aware attention [26] is used in the attention layer. The ASR decoder is a single layer LSTM with 320 units. The accent classifier is a 2-layer 1D convolution network with 128 channels and 3 kernel size, followed by an average pooling layer and a final FC output layer. In Eq. 2, $\lambda_1 = 0.5, \lambda_2 = 0.1, \lambda_3 = 0.5$ and $\lambda_4 = 0.1$, heuristically making the four loss terms to be at similar numerical scale. The transform 1 and 2 in the baseline approach are 2-layer and 4-layer BLSTMs with 128 units per direction, respectively.

The speaker encoder model is trained for 1000k steps with the Adam optimizer using batch size of 640 and learning rate of 0.0001. The TTS model is trained for 100k steps with the Adam optimizer using batch size of 16 and learning rate of 0.001. The accented ASR model is first pre-trained for 160k steps with the Adadelta optimizer using batch size of 16 and learning rate 1 on speech data from the native speakers and p248. Then it is fine-tuned on speech data from only p248 for another 5k steps with unchanged batch size and learning rate.

In our experiments, we also conduct an ablation study, where we

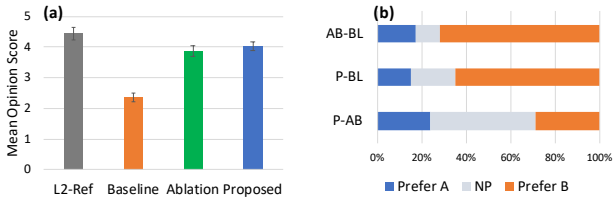

Fig. 5. (a) Mean opinion score results with 95% confidence interval. (b) Speaker similarity preference test results, where "AB-BL", "P-BL" and "P-AB" represent the comparison of "Ablation vs. Baseline", "Proposed vs. Baseline", "Proposed vs. Ablation", respectively.

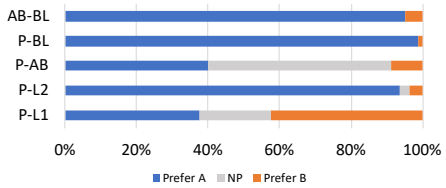

Fig. 6. Accentedness preference test results. "AB-BL", "P-BL", "P-AB", "P-L2" and "P-L1" represent comparison of "Ablation vs. Baseline", "Proposed vs. Baseline", "Proposed vs. Ablation", "Proposed vs. non-native-accented recording" and "Proposed v.s native-accented recording", respectively.

drop the accent embedding and accent classifier from the proposed approach. In total, we compare three systems in terms of their accent conversion performance: (1) the baseline system; (2) the proposed system and (3) the ablation system.

5.2. Experimental results

Three perceptual listening tests are conducted to evaluate the accent conversion performance of the baseline (BL), proposed (P) and ablation (AB) systems: a mean opinion score (MOS) test of audio naturalness, a speaker similarity XAB test and an accentedness AB test. We randomly choose 20 utterances from the test utterances of speaker p248 for evaluation. 10 Chinese speakers who are proficient in English have participated in these tests[2].

Audio naturalness. The audio naturalness is rated on a five-point scale (1-bad, 2-poor, 3-fair, 4-good, 5-excellent) in the MOS test. Audios generated from the three systems as well as non-native-accented reference recordings ("L2-Ref") are randomly shuffled before presenting to the listeners. Each group of audio corresponds to the same text content. Listeners are allowed to replay audio samples as many times as necessary. The MOS results are shown in Fig. 5(a). The proposed approach receives a 4.0 MOS (L2-Ref receives MOS of 4.4), which is much higher in statistical significance than that of the baseline approach. The fact that the proposed approach achieves slightly higher MOS than the ablation system illustrates that adding the accent embedding and classifier is helpful for extracting accent-agnostic linguistic content beneficial for speech synthesis from accented speech. Since we use seq2seq-based ASR and TTS models, the converted speech has more native-like pronunciation patterns such as duration and speaking rate, which are very different from those of the source accented utterances.

Speaker similarity. We compare all the three systems in terms of their speaker similarity of the converted speech to the non-native-accented source speech. In the XAB test, X indicates the non-native reference sample. Paired speech samples (A and B) with the same text content as the reference are presented and the listeners are asked to determine which one has closer timbre to the reference. Listeners are also allowed to replay audio samples and choose "no preference (NP)" if they cannot distinguish the difference. Audios are played in reverse to avoid influence from underlying contents. The similarity test results are shown in Fig. 5(b). We can see that the baseline system achieves significantly better similarity performance than the proposed system. The results are reasonable since the speech synthesis model in the proposed approach never sees speech data from the non-native accented speaker. We expect the synthesis model to infer the speaker timbre from the speaker embedding generated by the speaker encoder with only one utterance (i.e., voice cloning). Access to speech data from more speakers can be helpful for training a more generalizable speaker encoder. [14] achieves very good voice cloning performance by training a speaker encoder using speech data from 18K speakers; however, we cannot access such a large data corpus. We find no statistical differences between the proposed and the ablation systems (p = 0.36).

Accentedness. In the AB test on accentedness, we first let participants listen to native and non-native reference audios. Then paired speech samples (A and B) with the same textual content are presented and the listeners are asked to choose the more native-like samples. The results are shown in Fig. 6. According to the preference tests between "P-BL" and "P-AB", the listeners are very confident that the proposed approach can generate more native-accented utterances than the baseline and ablation approaches ($p << 0.001$). Even the ablation approach is able to achieve significantly better accentedness performance ($p << 0.001$) than the baseline approach. The DTW process in the baseline approach may introduce alignment errors and mapping from L2 PPGs to L1 PPGs using a neural network may not be effective. According to the results of "P-L2" and "P-L1", we can conclude that the proposed approach can remove non-native accented pronunciation patterns from the L2 speech and make the converted speech indistinguishable in accent from the native-accented speech, the p values are $2.3 \times e^{-8}$ and 0.06, respectively.

6. CONCLUSION

In this paper, we have presented an end-to-end accent conversion approach, which is the first model that is able to convert non-native accented into native-accented speech without any guidance from native reference audio during conversion phase. The system is composed of four independently trained neural networks: a speaker encoder, a multi-speaker TTS model, an accented ASR model and a neural vocoder. Experimental results show that the proposed approach can convert Hindi-accented English speech into native American English speech with high naturalness, which is indistinguishable from natural native speech. We expect the synthesis model to generate the desired target speaker timbre from the speaker embedding obtained from the speaker encoder. But the voice cloning performance is constrained by the amount of training data. The speaker similarity of the converted speech needs further research effort to improve, which will be our future work.

7. ACKNOWLEDGEMENT

This work is supported by Joint Research Scheme of National Natural Science Foundation of China - Research Grant Council of Hong Kong (NSFC-RGC) (61531166002, N_CUHK404/15).

[2]Audios can be found in "https://liusongxiang.github.io/end2endAC/"

References

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

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

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

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

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

[6] M. Huckvale and K. Yanagisawa, "Spoken language conversion with accent morphing," 2007.

[7] T. Toda, A. W. Black, and K. Tokuda, "Voice conversion based on maximum-likelihood estimation of spectral parameter trajectory," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 8, pp. 2222–2235, 2007.

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

[9] S. Liu, J. Zhong, L. Sun, X. Wu, X. Liu, and H. Meng, "Voice conversion across arbitrary speakers based on a single target-speaker utterance.," in *Proc. Interspeech*, 2018, pp. 496–500.

[10] S. Liu, Y. Cao, X. Wu, L. Sun, X. Liu, and H. Meng, "Jointly trained conversion model and wavenet vocoder for non-parallel voice conversion using mel-spectrograms and phonetic posteriorgrams," *Proc. Interspeech 2019*, pp. 714–718, 2019.

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

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

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

[14] 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. 4480–4490.

[15] T. Viglino, P. Motlicek, and M. Cernak, "End-to-end accented speech recognition," *Proc. Interspeech*, pp. 2140–2144, 2019.

[16] L. Wan, Q. Wang, A. Papir, and I. L. Moreno, "Generalized end-to-end loss for speaker verification," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4879–4883.

[17] S. Kim, T. Hori, and S. Watanabe, "Joint ctc-attention based end-to-end speech recognition using multi-task learning," in *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2017, pp. 4835–4839.

[18] N. Kalchbrenner, E. Elsen, K. Simonyan, S. Noury, N. Casagrande, E. Lockhart, F. Stimberg, A. v. d. Oord, S. Dieleman, and K. Kavukcuoglu, "Efficient neural audio synthesis," *arXiv preprint arXiv:1802.08435*, 2018.

[19] W. Chan, N. Jaitly, Q. V. Le, and O. Vinyals, "Listen, attend and spell," *arXiv preprint arXiv:1508.01211*, 2015.

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

[21] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: a large-scale speaker identification dataset," in *Proc. Interspeech*, 2017.

[22] J. S. Chung, A. Nagrani, and A. Zisserman, "Voxceleb2: Deep speaker recognition," in *Proc. Interspeech*, 2018.

[23] C. Veaux, J. Yamagishi, K. MacDonald, et al., "Cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit," *University of Edinburgh. The Centre for Speech Technology Research (CSTR)*, 2017.

[24] J. S. Garofolo, "Timit acoustic phonetic continuous speech corpus," *Linguistic Data Consortium, 1993*, 1993.

[25] S. Liu, L. Sun, X. Wu, X. Liu, and H. Meng, "The hccl-cuhk system for the voice conversion challenge 2018.," in *Odyssey*, 2018, pp. 248–254.

[26] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *Advances in neural information processing systems*, 2015, pp. 577–585.