Interspeech 2022
18-22 September 2022, Incheon, Korea

Synthesizing Near Native-accented Speech for a Non-native Speaker by Imitating the Pronunciation and Prosody of a Native Speaker

Brian Mak[1] *and Raymond Chung*[1,2]

[1]1Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong
[2]Logistics and Supply Chain MultiTech R&D Centre, Pok Fu Lam, Hong Kong
rchung@lscm.hk, mak@cse.ust.hk

Abstract

This paper investigates how to reduce foreign accent in the synthesis of native (L1) speech for a non-native (L2) speaker. We focus on two major aspects of foreign accents: mispronunciations and improper prosody (rhythm, phonemes duration, and pauses). Firstly, to reduce mispronunciations, the mel-spectrograms generated by an L2 text-to-speech (TTS) model are fed to a pre-trained speech recognizer and the mispronunciation information is fed back to the TTS model during back-propagation to help the model learn correct native mel-spectrograms. Secondly, to imitate L1 speech prosody, a recent data augmentation (DA) technique originally proposed for speaking style transfer is applied to transfer L1 speaking style to L2 speakers. The DA technique creates additional L2 speeches when L2 speakers try to imitate L1 speeches. Automatic speech recognition on native-accented speeches synthesized from non-native speakers by the proposed method gives a lower word error rate. The speaker embeddings produced by a pre-trained speaker verifier from the original L2 speakers' speech and their synthesized speech are highly similar. Finally, subjective MOS scores on the synthesized speech show that they have good quality and reduced accentedness.

Index Terms: text-to-speech, neural speech synthesis, accent conversion

1. Introduction

In this paper, we study how to reduce foreign accent in the synthesis of native-accented (L1) English speech for a non-native (L2) English speaker. There are many applications for this: for example, (1) in pronunciation learning of a second language by L2 speakers, the current practice is to have the L2 speakers listen to native speech spoken by a golden L1 speaker and ask them to imitate the pronunciation of the latter. However, it is found that it is pedagogically more effective for L2 learners to train their pronunciation skills from self-imitation [1, 2], which can be made possible if we may synthesize L1 speech from the L2 speaker with the correct L1 pronunciation and prosody. (2) the technique can be used in dubbing a film into another language, as it is highly desirable that an actor's voice in the dubbed film sounds like his/her original voice though he/she does not speak in the dubbed language (well). In both examples, the speaking script is given. If the script is not given, an L1 speech recognizer can be added as a frontend to first produce the script before the proposed synthesis, and the whole system becomes a foreign accent conversion (FAC) system. The problem is more difficult in many real situations when there are no parallel data between the L1 and L2 speakers, and it is not realistic to collect too many training data from the L2 users.

The goodness of such synthetic L1 speech for L2 speakers can be measured in the following four aspects: (1) *quality*: is the synthetic speech natural and intelligible? (2) *correctness*: are the words in the synthetic speech pronounced correctly? (3) *voice similarity*: does the voice of the synthetic L1 speech sound like the target L2 speaker? (4) *accentedness*: is the accent of the synthetic L1 speech close to the native accent?

Theoretically, the problem may be treated solely as a speech synthesis problem, and it can be solved by training a good L1 text-to-speech (TTS) model, a good L1 vocoder and a perfect speaker encoder so that such TTS system can generate L1 speech from any voices including L2 voices. With the advance of deep learning, high-quality neural TTS models [3, 4] and vocoders [5, 6, 7] are available nowadays. However, a perfect speaker encoder is more elusive. As shown in [8], even with the use of a speaker encoder that is trained with speech data from 18K speakers, the synthetic speech of *unseen* speakers from a multi-speaker TTS system can only give a speaker similarity score of around 3 (out of a maximum score of 5) in a Mean Opinion Score (MOS) evaluation.

Most related works can be found in the area of FAC instead. In general, FAC systems that use L1 TTS model/vocoder [9] will give better performance in terms of pronunciation correctness and native accentedness, whereas FAC systems that use L2 TTS model [10, 11, 12, 13] usually give better voice similarity for its synthetic speech which, however, retains stronger L2 foreign accent. Moreover, if L2 speech is used as the input [9], the non-native pronunciation errors will be propagated to the output though the problem can be mitigated by training a pronunciation correction model as in [12].

In this work, we propose to solve the problem by building an L2 TTS model by fine-tuning from a well-trained L1 TTS model with a (relatively) small amount of L2 utterances to ensure a good quality of the synthetic L2 speech with a high voice similarity. To reduce foreign accent of the L2 speaker due to non-native prosody, augmented L2 training data are created by imitating L1 training utterances on-the-fly using the technique in [14] which is originally developed for speaking style transfer. Finally, to reduce foreign accent due to mispronunciations, the generated speech is passed to a speech recognizer and the TTS model is trained to reduce the recognition errors as well. All these are done with no parallel data.

2. Related works

Deep learning has been applied successfully to neural speech synthesis (or TTS) [3, 8, 4], voice conversion (VC) [15, 16, 17], and foreign accent conversion (FAC) [10, 9, 11, 12, 13] in recent years to produce high-quality synthetic speech (using a good

10.21437/Interspeech.2022-11124

neural vocoder [5, 6, 7]). These applications use a sequence-to-sequence model, consisting of an encoder-decoder architecture with an intermediate attention layer to convert a word sequence or speech to mel-spectrograms. Tacotron2 [3] is one of the state-of-the-art TTS models that is also the core component in many VC/FAC models. It can be trained as a multi-speaker TTS model with an independently trained speaker encoder [8] so that theoretically it can synthesize any text to speech for any speaker given his/her embedding obtained from the speaker encoder. However, for unseen speakers, the speaker similarity score of its synthesized speech is still unsatisfactory [8].

FAC is a special case of VC; it takes an L1 or L2 speech as input and tries to convert it to an L1 native speech spoken by the L2 target speaker. For example, [10] proposed to extract speaker-independent phonetic posteriorgrams (PPGs) instead of phonemes from an L1 reference speech input to drive an L2 speaker-dependent Tacotron2 to produce native speech for the target L2 speaker. Since the PPG representation is much longer than the original phoneme or character representation used in Tacotron2, methods using PPG representation need extra work to overcome this mismatch. It was also noticed that the synthesized L2 speech contained mispronunciations and had some wrong intonations. [11] assumed text input for its FAC but its L2-TTS model was trained with both phoneme embeddings and PPGs. Thus, it also trained an L1 acoustic model and an L1 TTS model so that during inference, it would first generate L1 synthesized speech from the input text, from which L1 PPGs were extracted for FAC. It used GMM attention to solve the issue of using long PPG vector in training its TTS model. The system in [9] directly took an L2 speech input for FAC. It extracted the phoneme sequence using an accented speech recognizer, and then used an L1 multi-speaker Tacotron2 to synthesize L2 mel-spectrograms by conditioning on the L2 target speaker's embedding given by an independently trained speaker encoder. Though it gave better quality of L2 synthesized speech, it found that the speaker similarity was not satisfactory probably because the speaker encoder was not trained well enough, and it did not deal with mispronounced words in the L2 input speech. [12], like [9] directly converted an L2 speaker's speech to its L1 native-accented speech with an L2 synthesizer but added a pronunciation correction model to correct the mispronounced words in the input. Finally, the recent Accentron [13] claimed to get very good FAC results from both seen or unseen speakers by using bottleneck features extracted from a large speaker-independent acoustic model as the linguistic representation, a separately trained accent encoder and speaker encoder from large amount of data, and a multi-speaker TTS model.

3. Proposed model

In this paper, we design an accent-reduced TTS model to reduce accent in the synthesized L1 speech for a target L2 speaker due to mispronunciations and wrong prosody (such as wrong rhythm, phonemes and pauses durations, stress and intonation). The proposed model in the training phase is illustrated in Fig. 1. It is modified from the standard Tacotron2 [3] with a simple speaker encoder, and they are jointly trained to minimize the Tacotron loss $Loss_{taco}$, which is the sum of the squared errors in predicting the mel-spectrograms and the binary cross entropy in predicting the stop tokens. Tacotron2 is extended with four additional modules[1] to reduce foreign accent:

1. automatic speech recognition (ASR) feedback
2. GST style encoder using audio input
3. TP-GST style predictor from text input
4. L1 speech imitation by L2 speakers via on-the-fly data augmentation

The first ASR module is added to improve output speech quality and reduce mispronunciations, whereas the remaining three additional modules are designed to generate speech with a more native prosody at the utterance and phoneme levels. Note that the first two modules, ASR model and GST style encoder are only used in training and are then discarded during inference.

3.1. ASR model

To reduce mispronunciations from TTS outputs, during TTS training, they are passed to a native ASR model, and the TTS model is trained to minimize the recognized phoneme errors $Loss_{asr}$. By minimizing $Loss_{asr}$, the TTS model is "forced" to generate "more" native mel-spectrograms from the ASR perspective. Thus, we have

$$\hat{X_{seq}} = ASR(MEL)\ . \tag{1}$$

where MEL is the synthesized mel-spectrogram and $\hat{X_{seq}}$ is the predicted phoneme sequence. Connectionist temporal classification (CTC) [18] is used without a language model to compute the ASR loss as follows:

$$Loss_{asr} = CTC(\hat{X_{seq}}, X_{seq})\ . \tag{2}$$

where X_{seq} is the target phoneme sequence.

The use of an additional ASR module in a TTS system has been studied in [19] to improve unsupervised TTS style transfer, while [20] jointly trained the two models resulting in better performance of both models. Here, we use it for a different purpose: to reduce foreign accent due to L2 pronunciation errors.

3.2. GST style encoder

During training, a speaking style encoder is employed to extract the style embedding from a reference audio, which should capture its prosody at the utterance level. It is implemented using the global style tokens (GST) as in [21], which has been found effective for same-text prosody transfer (pitch and rhythm) between speakers [22, 23, 14].

3.3. TP-GST style predictor

Since during inference, there will be no reference audio but only text input, we train a GST style predictor, called text-predicted GST (TP-GST) [24], to predict from the input text the same GST style embedding extracted by the GST style encoder from its corresponding audio during model training. The TP-GST module in [24] is modified to condition its prediction on whether the input audio is native or non-native speech as our Tacotron2 will be trained with both native and non-native speech. During inference, this TP-GST style predictor will produce the required native style embedding based only on the input text and by setting the condition to native.

3.4. L1 speech imitation by L2 speakers via on-the-fly data augmentation (DA)

Recently, we proposed an on-the-fly DA scheme for speaking style transfer among speakers for TTS without the use of parallel data nor in-domain data [14]. In this paper, we treat native speaking and non-native speaking as two styles, and apply our

[1] The standard Tacotron2 components are connected with red arrows, while additional modules are connected to Tacotron2 with blue arrows.

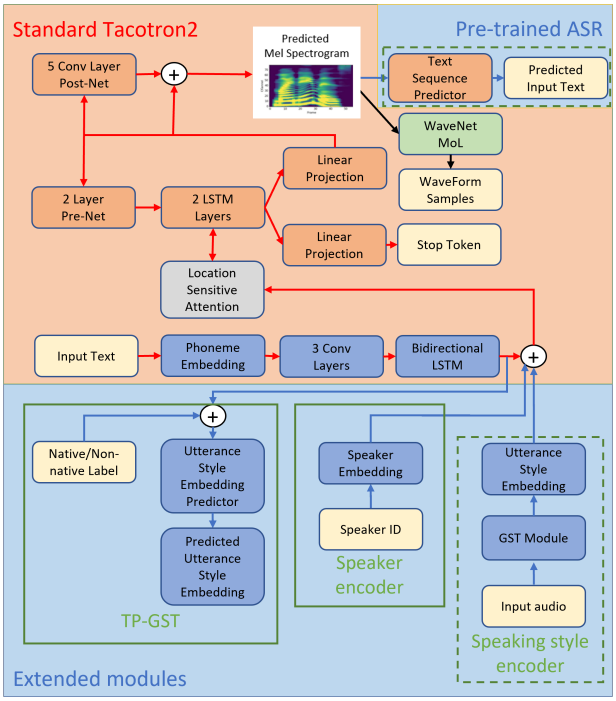

Figure 1: *Proposed model in the training phase.*

DA method to transfer the native prosody to L2 speakers at the finer phoneme level.

During Tacotron2 training, the decoder takes the encoder output and generates the corresponding mel-spectrograms using autoregression with an attention mechanism. A by-product of decoding is the attention alignment matrix, which is usually discarded afterwards. This alignment matrix actually encapsulates the prosody of the training utterance. We make use of the alignment matrix and create augmented data on-the-fly as follows: Suppose there is a training sample $\{text_p, audio_p\}$ from an L1 speaker p. We first pass the sample through Tacotron2 as usual, but then we retain its attention alignment matrix $\boldsymbol{A_p}$. We then ask an L2 speaker q to imitate L1's sample by passing the same sample to Tacotron2 but with L2's speaker embedding, resulting in L2's alignment matrix $\boldsymbol{A_q}$. For the augmented imitation data, we introduce the alignment loss $Loss_{align} = |\boldsymbol{A_p} - \boldsymbol{A_q}|_2$ requiring the two alignment matrices as similar as possible as depicted in Fig. 2. By minimizing the alignment loss, non-native speaker q will try to imitate the prosody and other speaking characteristics of native speaker p.

The overall training loss function for our TTS model is:

$$Loss = Loss_{taco} + Loss_{asr} + Loss_{tpgst} + Loss_{align} \quad (3)$$

where $Loss_{tpgst}$ is the TP-GST estimation loss. We will call the sum of $Loss_{taco}$ and $Loss_{tpgst}$ the TTS loss, $Loss_{asr}$ the ASR loss, and $Loss_{align}$ the DA loss.

4. Experiments and results

To evaluate our proposed model, one native English speaker was chosen as the "golden speaker", and all other non-native English speakers have to imitate the golden speaker's native accent (pronunciation and prosody) to reduce their own accent in speaking English. An L1 TTS model was first built for the golden speaker and then it was adapted in various ways to each individual L2 speaker in the evaluation. Both objective and subjective evaluations were performed to study the effectiveness of our model.

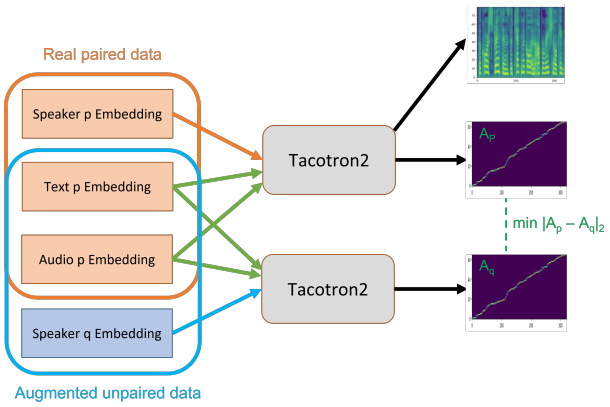

Figure 2: *Proposed on-the-fly data augmentation scheme for imitating L1 speech by an L2 speaker.*

4.1. Data

The female native English speaker in the LJS corpus [25] was chosen as the "golden speaker"; there are ∼24 hours of her speech in the corpus. Three non-native speakers from the L2-ARCTIC corpus [26] were selected to imitate the golden speaker: the female Arabic speaker ZHAA, the female Chinese speaker LXC and the male Korean speaker YKWK. Each L2 speaker recorded 1132 utterances, which are equivalent to roughly one hour of data. We used the first 1032 utterances for model training, and selected 25 utterances from the remaining data for testing. The split is the same as in [11]. We compared our synthesized speeches with those from [11] simply because they are publicly available and are reasonably good.

All speech data were sampled or re-sampled at 22.05kHz, and their mel-spectrograms were computed with a FFT size of 1024, a hop length of 12.5 ms and a window size of 50 ms.

4.2. Experimental setup

We used a simplified version of the Deep Speech model [27] as our ASR model, which consisted of layers of linear projections, a Bi-RNN layer, and a final softmax classifier to predict the phoneme sequence of the synthesized speech. It was separately trained using the LJS corpus for 300 epochs, and its weights were then frozen in subsequent training of our TTS system.

The official codebase of Mellotron [23] was modified to implement our proposed TTS model. The dimension of the speaker embeddings was 128, and that of the style embeddings was 256. It was firstly trained with LJS speeches for 150 epochs. The resulting model serves as the pre-trained model to create L2 TTS model for each of the three chosen L2-ARCTIC speakers. The following four adaptation methods were studied with the use of L2 training data:

1. **Fine-tuning (FT)**: fine-tuned for 100 epochs by minimizing only the TTS loss.
2. **FT + ASR**: fine-tuned for 100 epochs by minimizing only the TTS and ASR losses.
3. **FT + DA**: fine-tuned for 40 epochs by minimizing only the TTS loss, and then fine-tuned with the augmented data by minimizing both the TTS and DA losses.

Table 1: *Word error rate (WER %) and voice similarity (VS) of the L2 original and synthesized audios.*

Speaker	ZHAA		LXC		YKWK	
Utterances	WER	VS	WER	VS	WER	VS
Original L2	26.5	1.00	52.2	1.00	46.9	1.00
FT	23.1	0.86	43.4	0.86	38.8	0.90
FT+DA	20.5	0.86	35.4	0.80	36.6	0.86
FT+ASR	16.1	0.83	19.9	0.77	18.4	0.84
FT +ASR+DA	**14.4**	0.82	**15.2**	0.71	**14.1**	0.83
[11] Samples	18.3	0.74	NA	NA	NA	NA

4. **FT + ASR + DA**: fine-tuned using the same procedure in **FT + DA** except with the additional ASR loss.

Since each L2 speaker had less speech data than the L1 speaker's, the data of an L2 speaker were duplicated in each epoch to around the same amount of the L1 speaker's data to mitigate data imbalance. Finally, we used a pre-trained WaveGlow [6] vocoder to convert the mel-spectograms generated by our model to their audios[2].

4.3. Objective evaluations

The pronunciation accuracy and voice similarity of our synthesized L2 speeches are assessed by an independently trained ASR model and speaker verifier, respectively.

4.3.1. Pronunciation accuracy evaluation

We used Mozilla's Deep Speech model [28] (together with its default language model) as a proxy of a native English listener to check the pronunciation accuracy. The model was trained with thousands of hours of speech from multiple corpora, with a bias towards the US accents. It achieves a word error rate (WER) of 7.06% on the LibriSpeech clean test corpus. From Table 1, we observe that, compared with the WER on the original L2 speeches across the three L2 speakers, all the four adaptation schemes: **FT**, **FT + DA**, **FT + ASR**, and **FT + ASR + DA**, can reduce mispronunciations in the L2 synthesized speech effectively by a relative of 13-17%, 22-32%, 39-62%, and 46-71%, respectively. The improvement achieved by the **FT + ASR** scheme is particularly drastic.

4.3.2. Voice similarity evaluation

The independently trained Resemblyzer model [29] was used to judge voice similarity. It is a deep learning-based speaker verifier [30] that was trained on speech data from 8K speakers. It achieved an equal error rate of around 4% on 9 enrollment utterances. The Resemblyzer model was used to extract speaker embeddings from all original or synthesized utterances. We compared the similarity between the speaker embedding extracted from a synthesized utterances to its original L2 utterances (of the same text) by computing their cosine similarity. From Table 1, it is interesting to see that although the pronunciation accuracy of an L2 voice can be improved by TTS training with data augmentation and ASR feedback, the synthesized voice becomes less similar to the original voice by a relative 17-29%. For a comparison, it is worth noting that most similar works [19, 10, 13] obtained voice similarity in low 70s%.

In both objective measures, our proposed model **FT + ASR + DA** performed much better than the one in [11].

[2]Samples are available at https://raymond00000.github.io/attsdemo.html

Table 2: *ABX preference test results on voice similarity.*

Speaker	Proposed Model	Samples of [11]	No Preference
ZHAA	51%	27%	22%

Table 3: *MOS results at 95% confidence level.*

Audio	Naturalness	Accentedness
Original L2 utterances	3.59±0.09	6.19±0.25
Native TTS	3.95±0.13	2.23±0.26
Proposed TTS	3.54±0.12	4.93±0.41

4.4. Subjective evaluation

We utilized Amazon Mechanical Turk to perform a subjective evaluation on the naturalness, accentedness, and voice similarity of the following speech samples of the same content: (a) ZHAA's original L2 utterances; (b) utterances synthesized with the LJS speaker; (c) utterances synthesized by our **FT+AST+DA** model with the L2 speaker ZHAA. Ten listeners who declared themselves native American English speakers were recruited for the evaluation.

An ABX test was conducted for voice similarity evaluation. For each of the 25 ZHAA's test utterances, the listeners were asked to choose between the same utterance synthesized by our method or by [11] that sounded most similar to ZHAA's original sample; they were also allowed to give no preference. The result shown in Table 2 confirms the results in our objective voice similarity evaluation that our model gives a much higher speaker's voice similarity than [11].

Table 3 summarizes the raters' mean opinion scores (MOS) on the naturalness and accentedness of the synthesized speeches. Following the common convention, naturalness MOS is on a scale of 1–5 whereas accentedness MOS is on a scale of 1–9. As expected, the native TTS speech is the most natural and has the least accent. Speech synthesized from our proposed model is as natural as the original L2 utterances but with a much lower accentedness (which drops from 6.19 to 4.93).

5. Conclusion

In synthesizing near-native speech from a non-native speaker, one has to trade off among the quality, correctness, accentedness, and voice similarity of the synthesized speech. In this paper, we proposed a TTS model that can reduce the foreign accent in the synthesized speech of a non-native speaker, while maintaining a high voice similarity of the speaker. Accent is reduced by (1) forcing the TTS model to generate mel-spectrogram with fewer pronunciation errors, and (2) imitating the native prosody at the utterance level (by estimating the native GST style embedding) as well as at the phoneme level (by minimizing the L2 attention alignment loss). Voice similarity and quality is enhanced by fine-tuning on a pre-trained L1 TTS model with augmented imitation speech. We believe voice similarity is a critical objective for accent conversion, otherwise, one could simply synthesize the input text with a native voice.

6. Acknowledgements

This work was supported by grants from the RGC and ITF of the Hong Kong SAR, China (Project No. HKUST16200118, ITP/052/19LP).

7. References

[1] 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.

[2] D. Vigliano, K. Yoshimoto, and E. Pellegrino, "A self-imitation technique for the improvement of prosody in L2 Italian." in *Proceedings of the 22nd Annual Meeting of the The Association for Natural Language Processing*. IEEE, 2016, pp. 1189–1192.

[3] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Ryan, R. A. Saurous, Y. Agiomyrgiannakis, and Y. Wu, "Natural TTS synthesis by conditioning WaveNet on Mel spectrogram predictions," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2018, pp. 4779–4783.

[4] S. Ö. Arik, M. Chrzanowski, A. Coates, G. Diamos, A. Gibiansky, Y. Kang, X. Li, J. Miller, J. Raiman, S. Sengupta, and M. Shoeybi, "Deep Voice: Real-time neural text-to-speech," *CoRR*, vol. abs/1702.07825, 2017.

[5] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior, and K. Kavukcuoglu, "WaveNet: A generative model for raw audio," *CoRR*, vol. abs/1609.03499, 2016.

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

[7] N. Kalchbrenner, E. Elsen, K. Simonyan, S. Noury, N. Casagrande, E. Lockhart, F. Stimberg, A. van den Oord, S. Dieleman, and K. Kavukcuoglu, "Efficient neural audio synthesis," in *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, ser. Proceedings of Machine Learning Research, J. G. Dy and A. Krause, Eds., vol. 80. PMLR, 2018, pp. 2415–2424.

[8] Y. Jia, Y. Zhang, R. Weiss, Q. Wang, J. Shen, F. Ren, P. Nguyen, R. Pang, I. L. Moreno, and Y. Wu, "Transfer learning from speaker verification to multispeaker text-to-speech synthesis," in *Advances in neural information processing systems*, 2018, pp. 4480–4490.

[9] S. Liu, D. Wang, Y. Cao, L. Sun, X. Wu, S. Kang, Z. Wu, X. Liu, D. Su, D. Yu *et al.*, "End-to-end accent conversion without using native utterances," in *2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6289–6293.

[10] G. Zhao, S. Ding, and R. Gutierrez-Osuna, "Foreign accent conversion by synthesizing speech from phonetic posteriorgrams," in *Interspeech 2019, Graz, Austria*, G. Kubin and Z. Kacic, Eds., September 2019, pp. 2843–2847.

[11] W. Li, B. Tang, X. Yin, Y. Zhao, W. Li, K. Wang, H. Huang, Y. Wang, and Z. Ma, "Improving accent conversion with reference encoder and end-to-end text-to-speech," *arXiv preprint arXiv:2005.09271*, 2020.

[12] G. Zhao, S. Ding, and R. Gutierrez-Osuna, "Converting foreign accent speech without a reference," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 2367–2381, 2021.

[13] S. Ding, G. Zhao, and R. Gutierrez-Osuna, "Accentron: Foreign accent conversion to arbitrary non-native speakers using zero-shot learning," *Comput. Speech Lang.*, vol. 72, no. 101302, pp. 1–17, 2022.

[14] R. Chung and B. Mak, "On-the-fly data augmentation for text-to-speech style transfer," in *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2021.

[15] K. Qian, Y. Zhang, S. Chang, X. Yang, and M. Hasegawa-Johnson, "AutoVC: Zero-shot voice style transfer with only autoencoder loss," in *Proceedings of the International Conference on Machine Learning*, 2019, pp. 5210–5219.

[16] H. Lu, Z. Wu, R. Li, S. Kang, J. Jia, and H. Meng, "A compact framework for voice conversion using WaveNet conditioned on phonetic posteriorgrams," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2019, pp. 6810–6814.

[17] K. Tanaka, H. Kameoka, T. Kaneko, and N. Hojo, "ATTS2S-VC: Sequence-to-sequence voice conversion with attention and context preservation mechanisms," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2019, pp. 6805–6809.

[18] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd International Conference on Machine Learning*, 2006, pp. 369–376.

[19] D.-R. Liu, C.-Y. Yang, S.-L. Wu, and H.-Y. Lee, "Improving unsupervised style transfer in end-to-end speech synthesis with end-to-end speech recognition," in *2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2018, pp. 640–647.

[20] A. Tjandra, S. Sakti, and S. Nakamura, "Listening while speaking: Speech chain by deep learning," in *2017 IEEE Automatic Speech Recognition and Understanding Workshop, ASRU 2017, Okinawa, Japan, December 16-20, 2017*. IEEE, 2017, pp. 301–308.

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

[22] R. Skerry-Ryan, E. Battenberg, Y. Xiao, Y. Wang, D. Stanton, J. Shor, R. Weiss, R. Clark, and R. A. Saurous, "Towards end-to-end prosody transfer for expressive speech synthesis with tacotron," in *International Conference on Machine Learning*. PMLR, 2018, pp. 4693–4702.

[23] R. Valle, J. Li, R. Prenger, and B. Catanzaro, "Mellotron: Multispeaker expressive voice synthesis by conditioning on rhythm, pitch and global style tokens," in *2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6189–6193.

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

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

[26] G. Zhao, S. Sonsaat, A. O. Silpachai, I. Lucic, E. Chukharev-Hudilainen, J. Levis, and R. Gutierrez-Osuna, "L2-ARCTIC: A non-native English speech corpus," *Perception Sensing Instrumentation Lab*, 2018.

[27] A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates *et al.*, "Deep Speech: Scaling up end-to-end speech recognition," *arXiv preprint arXiv:1412.5567*, 2014.

[28] Mozilla, "Deep Speech, an open source python package for a Speech-To-Text engine," https://github.com/mozilla/DeepSpeech/releases, 2020.

[29] Resemble.AI, "Resemblyzer, an open source python package for a voice encoder," https://github.com/resemble-ai/Resemblyzer, 2019.

[30] 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.

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