Foreign Accent Conversion by Synthesizing Speech from Phonetic Posteriorgrams

Guanlong Zhao, Shaojin Ding, Ricardo Gutierrez-Osuna[1]

Department of Computer Science and Engineering, Texas A&M University, USA
{gzhao, shjd, rgutier}@tamu.edu

Abstract

Methods for foreign accent conversion (FAC) aim to generate speech that sounds similar to a given non-native speaker but with the accent of a native speaker. Conventional FAC methods borrow excitation information (F0 and aperiodicity; produced by a conventional vocoder) from a reference (i.e., native) utterance during synthesis time. As such, the generated speech retains some aspects of the voice quality of the native speaker. We present a framework for FAC that eliminates the need for conventional vocoders (e.g., STRAIGHT, World) and therefore the need to use the native speaker's excitation. Our approach uses an acoustic model trained on a native speech corpus to extract speaker-independent phonetic posteriorgrams (PPGs), and then train a speech synthesizer to map PPGs from the non-native speaker into the corresponding spectral features, which in turn are converted into the audio waveform using a high-quality neural vocoder. At runtime, we drive the synthesizer with the PPG extracted from a native reference utterance. Listening tests show that the proposed system produces speech that sounds more clear, natural, and similar to the non-native speaker compared with a baseline system, while significantly reducing the perceived foreign accent of non-native utterances.

Index Terms: phonetic posteriorgram, acoustic modeling, speech synthesis, accent conversion

1. Introduction

Foreign accent conversion [1-3] aims to create a new voice that has the voice quality[2] of a given non-native speaker and the pronunciation patterns (e.g., prosody, segmentals) of a native speaker. This can be achieved by combining accent-related cues from a native utterance with the voice quality of the non-native speaker. FAC has potential application in computer-assisted pronunciation training [3-5], where it could be used as a model voice to imitate.

The main challenge in FAC is to divide the speech signal into accent-related cues and voice quality. Multiple solutions have been proposed, including voice morphing [3, 6-8], frame pairing [1, 9], and articulatory synthesis [2, 10-12]. These approaches can reduce the accent of non-native utterances, but have various limitations. Voice morphing often generates voices that sound like a "third" speaker, one who is different from either speaker. Frame-pairing methods can synthesize speech that resembles the non-native speaker's voice but the syntheses retain some aspects of the native speaker's voice quality; this is because excitation information from the native speaker is used to synthesize the speech. Finally, articulatory synthesis needs specialized apparatus to collect articulation data, so they are not practical for real-world applications.

In this work, we propose to perform FAC in a speaker-independent phonetically-rich speech embedding: a phonetic posteriorgram (PPG) [13]. A PPG is defined as the posterior probability that each speech frame belongs to a set of predefined phonetic units (phonemes or triphones/senones), which retain the linguistic and phonetic information of the utterance. Our approach works as follows. In a first step, we generate a PPG for the non-native speaker using a speaker-independent acoustic model that is trained on a large corpus of native speech. Then, we construct a sequence-to-sequence speech synthesizer that captures the voice quality of the non-native speaker. The synthesizer takes a PPG sequence from the non-native speaker as the input and produces the corresponding mel-spectrogram sequence as the output. Finally, we train a neural vocoder, WaveGlow [14], to convert the mel-spectrogram into a raw speech signal. During testing, we feed the synthesizer with a PPG sequence from a native utterance. The resulting output contains the native speaker's pronunciation patterns and the non-native speaker's voice quality. The overall workflow of the proposed system is shown in Figure 1.

The proposed system has three advantages. First, it eliminates the need to borrow excitation information from the native reference speech, which prevents aspects of the native speaker's voice quality from leaking into the synthesized speech. Second, our system does not require any training data from the native reference speaker. Thus, we have the flexibility to use any reference voices during testing. Third, our system captures contextual information by means of a sequence-to-sequence model, which has shown state-of-the-art performance on multiple tasks [15-17], helping produce better audio quality.

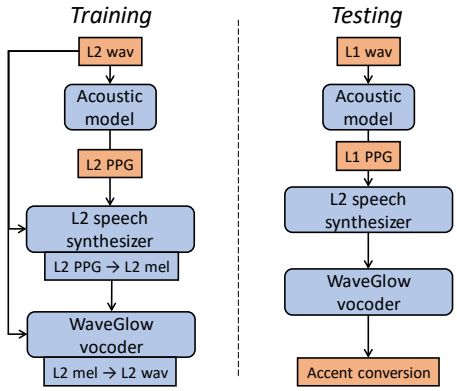

Figure 1: *Overall workflow of the proposed system. L1: native, L2: non-native.*

[1] Work supported by NSF awards 1619212 and 1623750.

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

2. Related work

Early attempts at accent conversion used voice morphing [3, 6-8] to control the degree of accent by blending spectral components from the native and non-native speakers. In [18, 19], the authors used PSOLA to modify the duration and pitch patterns of accented speech. Aryal and Gutierrez-Osuna [1] adapted voice conversion (VC) techniques, replacing Dynamic Time Warping (DTW) with a technique that matched source and target frames based on their MFCC similarity after vocal tract length normalization. Later, Zhao *et al*. [9] used PPG similarity instead of MFCC similarity to pair acoustic frames.

PPGs have been applied to many tasks, e.g., neural-network-based speech recognition [20, 21], spoken term detection [13], mispronunciation detection [22], and personalized TTS [23]. PPGs have also gained much recent attention for VC. Xie *et al.* [24] divided PPGs from a target speaker into clusters and then mapped PPGs from a source speaker into the closest cluster of the target speaker. Sun *et al.* [25] used PPGs for many-to-one voice conversion. Miyoshi *et al*. [26] extended the PPG-based VC framework to include a mapping between source and target PPGs using LSTMs; they obtained better speech individuality ratings but worse audio quality than a baseline that did not include the PPG mapping process. Zhang *et al*. [15] concatenated bottleneck features and mel-spectrograms from a source speaker, then used a sequence-to-sequence model to convert the source mel-spectrograms into those of the target speaker, and finally recovered the speech waveform using a WaveNet [27] vocoder. Their model required parallel recordings and needed to train a new model for each speaker pair. They then applied text supervision [28] to resolve some of the mispronunciations and artifacts in the converted speech. Recently, Zhou *et al.* [29] adopted bilingual PPG for cross-lingual voice conversion.

3. Method

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

3.1. Acoustic modeling and PPG extraction

We use a DNN with multiple hidden layers and the p-norm non-linearity as the AM. We train the AM on a native speech corpus [30] by minimizing the cross-entropy between outputs and senone labels obtained from a pre-trained GMM-HMM forced aligner. Training on native speech is critical for our task because the native and non-native frames have to be matched in a native phonetic space. For more details about the AM, please refer to [31].

3.2. PPG-to-Mel-spectrogram conversion

We convert PPGs from the non-native speaker into their corresponding mel-spectrograms using a modified Tacotron 2 model [32]. The original Tacotron 2 model takes a one-hot vector representation of characters and passes it to an encoder LSTM that converts it into a hidden representation, which is then passed to a decoder LSTM with a location-sensitive attention mechanism [33] that predicts the mel-spectrogram. To improve model performance, the character embedding is passed through multiple convolution layers before being fed to the encoder LSTM. The decoder appends a PreNet (two fully connected layers) before passing the predicted mel-spectrogram to the attention and decoder LSTM to extract structural information. It also applies a PostNet (multiple 1-D convolutional layers) after the decoder to predict spectral details and add them to the raw prediction.

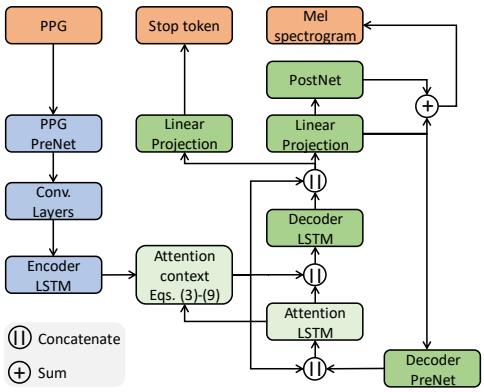

Figure 2: *PPG-to-Mel conversion model.*

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

The original Tacotron 2 was designed to accept character sequences as input, which are significantly shorter than our PPG sequences. For example, each sentence in our speech corpus [34] contains an average of 41 characters, whereas the PPG sequence has a few hundred frames. Therefore, the original Tacotron 2 attention mechanism would be confused by such long input sequences and cause misalignment between the PPG and acoustic sequences, as pointed out in [15]. As a result, the inference would be ill-conditioned and would generate non-intelligible speech. One solution to this issue is to train the PPG-to-Mel model with shorter PPG sequences. For example, one could use word segments instead of sentences. However, this solution has several issues. First, to obtain accurate word boundaries, we need to perform forced alignment on the training sentences, which requires access to the transcription. Second, and more importantly, training with short segments and performing inference with significant longer input sequences leads to model failure, as observed in [33].

We resolve this issue by adding a locality constraint to the attention mechanism. Speech signals have a strong temporal-continuity and progressive nature. To capture the phonetic context, we only need to look at the PPGs in a small local window. Inspired by this, at each decoding step during training we constrain the attention mechanism to look at a window in the hidden state sequence, instead of the full sequence. We formally define this constraint as follows. Let d_i be the output of the decoder LSTM at time step i, y_i be the predicted acoustic features (output after applying a linear projection on d_i), and $h = [h_1, \ldots, h_T]$ be the full sequence of hidden states from the encoder. Applying the location-sensitive attention mechanism, we have,

$$d_i = \text{DecoderLSTM}(s_{i-1}, g_i). \quad (1)$$

where s_{i-1} is the hidden state of the attention LSTM at the $(i-1)$-th time step, and g_i is the attention context,

$$s_i = \text{AttentionLSTM}\big(s_{i-1}, g_i, \text{PreNet}(y_i)\big), \quad (2)$$

$$g_i = \sum_{j=1}^{T} \alpha_i^j h_j \,. \tag{3}$$

and,

$$\alpha_i = \text{Attend}(s_{i-1}, \alpha_{i-1}, h) = [\alpha_i^1, \ldots \alpha_i^T], \tag{4}$$

$$\alpha_i^j = \frac{\exp(e_{ij})}{\sum_{j=1} \exp(e_{ij})}, \tag{5}$$

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

$$e_{ij} = v^T \tanh\left(W s_{i-1} + V h_j + U f_i^j + b\right), \tag{6}$$

$$f_i = F * \alpha_{i-1} = [f_i^1, \ldots, f_i^T], F \in R^{k \times r}, \tag{7}$$

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

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

$$\tilde{h} = [0, \ldots, 0, h_{i-w}, \ldots, h_{i+w}, 0, \ldots, 0], \tag{8}$$

where w is the window size, and let,

$$\alpha_i = \text{Attend}\left(s_{i-1}, \alpha_{i-1}, \tilde{h}\right). \tag{9}$$

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

$$L = \alpha \|G_{mel} - P_{Decoder}\|_2 + \beta \|G_{mel} - P_{PostNet}\|_2 + \gamma \text{CE}(G_{stop}, P_{stop}), \tag{10}$$

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

3.3. Mel-spectrogram to speech

We use a WaveGlow vocoder to convert the output of the speech synthesizer back into a speech waveform. WaveGlow is a flow-based [35] network capable of generating high-quality speech from mel-spectrograms (comparable to WaveNet). It takes samples from a zero mean spherical Gaussian (with variance σ) with the same number of dimensions as the desired output and passes those samples through a series of layers that transform the simple distribution to one that has the desired distribution. In the case of training a vocoder, we use WaveGlow to model the distribution of audio samples conditioned on a mel-spectrogram. WaveGlow can achieve real-time inference speed using only a single neural network, whereas WaveNet takes a long time to synthesize an utterance due to its auto-regressive nature. For more details about the WaveGlow vocoder, we refer readers to [14].

4. Experiments and results

4.1. Experimental setup

We used the Librispeech corpus [30] to train the AM. It contains 960 hours of native English speech, most of which from North America. The AM has five hidden layers and an output layer with 5816 senones. We trained the PPG-to-Mel and WaveGlow models on two non-native speakers, YKWK (native male Korean speaker) and ZHAA (native female Arabic speaker) from the publicly-available L2-ARCTIC corpus [34]. We applied noise reduction on the original L2-ARCTIC recordings using Audacity [36] to remove ambient background noise. For the native reference speech, we used two North American speakers, BDL (M) and CLB (F) from the ARCTIC corpus [37]. Each speaker in L2-ARCTIC and ARCTIC recorded the same set of 1132 sentences, or about an hour of speech. For each L2-ARCTIC speaker, we used the first 1032 sentences for model training, the next 50 sentences for validation, and the remaining 50 sentences for testing. All audio signals were sampled at 16 KHz. We used 80 filter banks to extract mel-spectrograms with a 10ms shift and a 64ms window. The PPG was also extracted with a 10ms shift.

The PPG-to-Mel model parameters are summarized in Table 1. We used a batch size of 6 and a learning rate of 1×10^{-4}. α, β, γ were empirically set to 1.0, 1.0, and 0.005, respectively. The window size w of the locality constraint of the attention mechanism was set to 20. We trained the model until the validation loss reached a plateau (~8h). For the WaveGlow models, we set σ to 0.701 during training and 0.6 during testing, as suggested by [14]. The batch size was 3 and the learning rate was 1×10^{-4}. The models were trained until convergence (~one day). All models were trained on a single Nvidia GTX 1070 GPU.

The AM was trained with Kaldi, and the other models were implemented in PyTorch and trained with the Adam optimizer [38]. For more details and audio samples, please refer to `https://github.com/guanlongzhao/fac-via-ppg`.

We compared our proposed system against a baseline from [9] that worked as follows. First, we computed the PPG for each native and non-native frame. Then, we used the symmetric KL divergence in the PPG space to pair the closest native and non-native frames. In a final step, we extracted Mel-Cepstral Coefficients (MCEPs) from the frame pairs to train a joint-density GMM (JD-GMM) spectral conversion as described in [39]. We then converted the native MCEPs using the JD-GMM to match the non-native speaker's voice quality. Finally, we used the STRAIGHT vocoder [40] to synthesize speech from the converted MCEPs combined with the native speaker's aperiodicity (AP) and F0 (normalized to the non-native speaker's pitch range). We used the same 1032-utterance training set for the baseline system. The GMM contained 128 mixtures and full covariance matrices. We used 24-dim MCEPs (excluding MCEP_0) and the Δ features. All features were extracted by STRAIGHT with a 10ms shift and 25ms window. For each system, we generated accent conversion for speaker pairs BDL-YKWK and CLB-ZHAA.

Table 1: *The model details of the PPG-to-Mel synthesizer.*

Module	***Parameters***
PPG PreNet	Two fully connected (FC) layers; 600 ReLU units; 0.5 dropout rate [41]
Conv. Layers	Three 1-D convolution layers (kernel size 5); batch normalization [42] after each layer
Encoder LSTM	One-layer Bi-LSTM; 300 cells in each direction
Decoder PreNet	Two FC layers; 300 ReLU units; 0.5 dropout rate
Attention LSTM	One-layer LSTM; 300 cells; 0.1 dropout rate
Attention	v in eq. (6) has 150 dims; eq. (7), $k = 32$, $r = 31$
Decoder LSTM	One-layer LSTM; 300 cells; 0.1 dropout rate
PostNet	Five 1-D conv. layers; 512 channels; kernel size 5

4.2. Results

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

For each test, 25 utterances per speaker pair (50 in total) from each system were randomly selected. The presentation order of the samples was randomized in all experiments.

The MOS test rated the audio quality and naturalness of audio samples on a five-point scale (1-bad, 2-poor, 3-fair, 4-good, 5-excellent). Audio quality and naturalness MOS described how clear and human-like the speech was, respectively. The two measures were obtained from non-overlapping groups of listeners to avoid bias. Each audio sample received at least 17 ratings. Listeners also rated the same set of original ARCTIC and L2-ARCTIC recordings as a reference. Results are summarized in Table 2 and Table 3. It should be noted that in [9], we established that the baseline system's audio quality MOS is around 0.4 higher than a conventional JD-GMM system that uses DTW for frame pairing. Therefore, our baseline is a stronger system than the conventional JD-GMM.

In all cases, our system outperformed the baseline significantly in both audio quality and naturalness. Although the two systems have lower audio quality MOS than the original recordings, there is *no* significant difference between the proposed system and either the ARCTIC ($p = 0.35$) or L2-ARCTIC ($p = 0.54$) recordings on the naturalness MOS, using a two-tail two-sample t-test.

In the voice similarity test, listeners were provided with three utterances, the original non-native utterance and syntheses from the two systems, and were asked to choose which of the two syntheses sounded more like the non-native speaker. Participants were also asked to rate their confidence level on a 7-point scale (1-not at all confident, 7-extremely confident) when making a choice. Participants were instructed to ignore accent when performing the task. Presentation order of samples from the two systems was counter-balanced in each trial, and 17 participants rated the audio samples. Results are presented in Table 4. In 72.47% of the cases, listeners preferred the proposed system with a 3.4 confidence level (above "somewhat confident"), whereas in the remaining 27.53% of the cases, listeners chose the baseline with a much lower confidence level (1.05, or "not at all confident.")

Table 2: *MOS results with 95% confidence intervals. Q: audio quality; N: naturalness.*

Conversion		***Baseline***	***Proposed***
BDL-YKWK	*Q*	3.23±0.11	**3.48**±0.12
	N	3.18±0.15	**3.59**±0.15
CLB-ZHAA	*Q*	2.86±0.15	**3.58**±0.14
	N	2.66±0.13	**3.32**±0.20
All pairs	*Q*	3.04±0.10	**3.53**±0.09
	N	2.92±0.12	**3.46**±0.13

Table 3: *MOS ratings for original recordings.*

Real speech		***Rating***
ARCTIC	*Q*	4.40±0.08
	N	3.54±0.11
L2-ARCTIC	*Q*	3.98±0.09
	N	3.50±0.08

Table 4: *Voice similarity test results.*

Measure	***Baseline***	***Proposed***
Preference	27.53±5.00%	**72.47**±5.00%
Confidence	1.05±0.21	**3.40**±0.32

Table 5: *Accentedness ratings.*

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

In the accentedness test, participants were asked to rate the degree of foreign accent in a nine-point scale (1-no foreign accent, 9-very strong foreign accent), which is commonly used in the pronunciation literature [43]. Each audio sample was rated by 18 individuals. Results are summarized in Table 5. Original utterances from ARCTIC speakers were rated as "no foreign accent" (1.20), whereas original utterances from the L2-ARCTIC speakers were rated as heavily accented (7.17). Both the baseline (2.94) and proposed (3.93) systems reduced the foreign accent significantly compared with the L2-ARCTIC speech but were rated more accented than the native speech. Surprisingly, speech generated from our system was rated as more accented than that of the baseline system; see discussion section for a potential explanation of this result.

5. Discussion and conclusion

The proposed accent-conversion system produces speech with better quality than the baseline system because it uses a state of-the-art sequence-to-sequence model (a modified Tacotron 2) to convert PPGs into mel-spectrograms, and then utilizes a neural vocoder to generate audio directly from the mel-spectrogram. This process takes advantage of the temporal-dependent natural of speech signals and avoids the use of conventional signal-processing based vocoders, which generally degrade the synthesis quality. We have also proposed an easy-to-implement locality constraint on the attention mechanism to make the PPG-to-Mel model trainable on utterance-level samples. Note that our MOS ratings are lower than those in the original Tacotron 2 and WaveGlow paper, largely because their systems were trained with 24× more data. One future direction for improving the MOS ratings of the proposed system is training the PPG-to-Mel and WaveGlow models jointly.

In contrast with the baseline, which borrows excitation information (F0, AP) from the native speaker, our system generates the non-native speaker's excitation directly from the synthesized mel-spectrogram. This prevents the voice quality of the native speaker from "leaking" into the synthesis, making it more similar to the voice quality of the non-native speaker.

Our system extracts native pronunciation patterns from the native PPG sequence, and therefore makes the synthesized speech significantly less accented than the non-native speech. The slight increase in accentedness rating compared to the baseline system could be the result of two factors. First, the AM inevitably produces recognition errors when extracting the PPG and these errors will be reflected as mispronunciations in the synthesis. Second, the proposed model does not explicitly model stress and intonation patterns; as such, we find that some of synthesis results have unexpected intonations. Therefore, in future work we plan to incorporate intonation information into the modeling process; one possible solution is to condition the PPG sequence on a normalized F0 contour when training and testing the PPG-to-Mel model.

Currently, the PPG-to-Mel and WaveGlow models need at least one hour of speech from the non-native speaker. This requirement may be relaxed by following the transfer-learning paradigm from multi-speaker TTS [44]. The ultimate goal of accent conversion is to eliminate the need for a reference utterance at synthesis time, i.e., to take a non-native utterance and automatically reduce its accent. This may be accomplished by learning a sequence-to-sequence mapping from the non-native speaker's PPG sequence to a native PPG sequence, and then driving the PPG-to-Mel synthesizer with this accent-reduced PPG sequence.

6. References

[1] S. Aryal and R. Gutierrez-Osuna, "Can Voice Conversion Be Used to Reduce Non-Native Accents?," in *ICASSP*, 2014, pp. 7879-7883.

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

[3] 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.

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

[5] S. Ding *et al.*, "Golden Speaker Builder: an interactive online tool for L2 learners to build pronunciation models," in *PSLLT*, 2017, pp. 25-26.

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

[7] 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.

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

[9] G. Zhao, S. Sonsaat, J. Levis, E. Chukharev-Hudilainen, and R. Gutierrez-Osuna, "Accent Conversion Using Phonetic Posteriorgrams," in *ICASSP*, 2018, pp. 5314-5318.

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

[11] S. Aryal and R. Gutierrez-Osuna, "Articulatory inversion and synthesis: towards articulatory-based modification of speech," in *ICASSP*, 2013, pp. 7952-7956.

[12] S. Aryal and R. Gutierrez-Osuna, "Accent conversion through cross-speaker articulatory synthesis," in *ICASSP*, 2014, pp. 7694-7698.

[13] T. Hazen, W. Shen, and C. White, "Query-by-example spoken term detection using phonetic posteriorgram templates," in *ASRU*, 2009, pp. 421-426.

[14] R. Prenger, R. Valle, and B. Catanzaro, "WaveGlow: A Flow-based Generative Network for Speech Synthesis," in *ICASSP*, 2019.

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

[16] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *NIPS*, 2014, pp. 3104-3112.

[17] Y. Wang *et al.*, "Tacotron: Towards end-to-end speech synthesis," *arXiv preprint arXiv:1703.10135,* 2017.

[18] S. Zhao, S. N. Koh, S. I. Yann, and K. K. Luke, "Feedback utterances for computer-adied language learning using accent reduction and voice conversion method," in *ICASSP*, 2013, pp. 8208-8212.

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

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

[21] Y. Miao, M. Gowayyed, and F. Metze, "EESEN: End-to-end speech recognition using deep RNN models and WFST-based decoding," in *ASRU*, 2015, pp. 167-174: IEEE.

[22] A. Lee, Y. Zhang, and J. Glass, "Mispronunciation detection via dynamic time warping on deep belief network-based posteriorgrams," in *ICASSP*, 2013, pp. 8227-8231.

[23] L. Sun, H. Wang, S. Kang, K. Li, and H. M. Meng, "Personalized, Cross-Lingual TTS Using Phonetic Posteriorgrams," in *Interspeech*, 2016, pp. 322-326.

[24] F.-L. Xie, F. K. Soong, and H. Li, "A KL Divergence and DNN-Based Approach to Voice Conversion without Parallel Training Sentences," in *Interspeech*, 2016, pp. 287-291.

[25] L. Sun, K. Li, H. Wang, S. Kang, and H. Meng, "Phonetic posteriorgrams for many-to-one voice conversion without parallel data training," in *ICME*, 2016, pp. 1-6.

[26] H. Miyoshi, Y. Saito, S. Takamichi, and H. Saruwatari, "Voice Conversion Using Sequence-to-Sequence Learning of Context Posterior Probabilities," in *Interspeech*, 2017, pp. 1268-1272.

[27] A. v. d. Oord *et al.*, "WaveNet: A Generative Model for Raw Audio," *arXiv preprint arXiv:1609.03499,* 2016.

[28] J.-X. Zhang, Z.-H. Ling, Y. Jiang, L.-J. Liu, C. Liang, and L.-R. Dai, "Improving Sequence-to-Sequence Acoustic Modeling by Adding Text-Supervision," in *ICASSP*, 2019.

[29] Y. Zhou, X. Tian, H. Xu, R. K. Das, and H. Li, "Cross-lingual voice conversion with bilingual phonetic posteriorgram and average modeling," in *ICASSP*, 2019.

[30] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an ASR corpus based on public domain audio books," in *ICASSP*, 2015, pp. 5206-5210.

[31] X. Zhang, J. Trmal, D. Povey, and S. Khudanpur, "Improving deep neural network acoustic models using generalized maxout networks," in *ICASSP*, 2014, pp. 215-219.

[32] J. Shen *et al.*, "Natural TTS synthesis by conditioning wavenet on mel spectrogram predictions," in *ICASSP*, 2018, pp. 4779-4783.

[33] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *NIPS*, 2015, pp. 577-585.

[34] G. Zhao *et al.*, "L2-ARCTIC: A Non-Native English Speech Corpus," in *Interspeech*, 2018, pp. 2783-2787.

[35] D. P. Kingma and P. Dhariwal, "Glow: Generative flow with invertible 1x1 convolutions," in *NIPS*, 2018, pp. 10236-10245.

[36] *Audacity*®. Available: http://www.audacityteam.org/

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

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

[39] 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.

[40] H. Kawahara, M. Morise, T. Takahashi, R. Nisimura, T. Irino, and H. Banno, "TANDEM-STRAIGHT: A temporally stable power spectral representation for periodic signals and applications to interference-free spectrum, F0, and aperiodicity estimation," in *ICASSP*, 2008, pp. 3933-3936.

[41] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research,* vol. 15, no. 1, pp. 1929-1958, 2014.

[42] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167,* 2015.

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

[44] Y. Jia *et al.*, "Transfer learning from speaker verification to multispeaker text-to-speech synthesis," in *NIPS*, 2018, pp. 4485-4495.