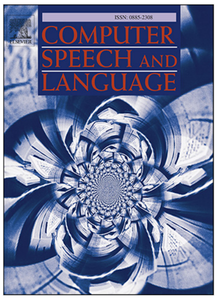


Accentron: Foreign accent conversion to arbitrary non-native speakers using zero-shot learning

Shaojin Ding, Guanlong Zhao, Ricardo Gutierrez-Osuna

PII: S0885-2308(21)00102-9
DOI: https://doi.org/10.1016/j.csl.2021.101302
Reference: YCSLA 101302

To appear in: *Computer Speech & Language*

Received date : 12 April 2021
Revised date : 19 September 2021
Accepted date : 21 September 2021

Accentron: Foreign Accent Conversion to Arbitrary Non-Native Speakers Using Zero-Shot Learning[⋆]

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

Department of Computer Science and Engineering, Texas A&M University, USA

Abstract

Foreign accent conversion (FAC) aims to create a new voice that has the *voice identity* of a given second-language (L2) speaker but with a native (L1) *accent*. Previous FAC approaches usually require training a separate model for each L2 speaker and, more importantly, generally require considerable speech data from each L2 speaker for training. To address these limitations, we propose Accentron, an approach that can generate accent-converted speech for arbitrary L2 speakers unseen during training. In the proposed approach, we first train a speaker-independent acoustic model on L1 corpora to extract bottleneck features that represent the linguistic content of utterances. Then, we develop a speaker encoder and an accent encoder to generate embedding vectors for the desired voice identity (L2 speaker's) and accent (L1 accent), respectively. Lastly, we use a sequence-to-sequence model to transform bottleneck-features to Mel-spectrograms, conditioned on the L2 speaker embedding and the L1 accent embedding. We conducted experiments on the L2-ARCTIC corpus under two testing conditions: the *standard* FAC setting where test L2 speakers were seen during training, and a *zero-shot* FAC setting where test L2 speakers were unseen during training. Accentron achieves over 27% relative improvement in accentedness ratings compared to two state-of-the-art FAC systems in the standard

[⋆] This work was supported by NSF awards 1619212 and 1623750.

[*] Corresponding author

Email addresses: shjd@tamu.edu (Shaojin Ding), gzhao@tamu.edu (Guanlong Zhao), rgutier@cse.tamu.edu (Ricardo Gutierrez-Osuna)

[1] Zhao is now with Google. This work was done solely with TAMU resources.

FAC setting. More importantly, our results show that Accentron generalizes to the zero-shot FAC setting with no performance loss. Therefore, in practical use scenarios (e.g., computer-assisted pronunciation training software), Accentron can effectively avoid the need to adapt or retrain the model, which significantly reduces computations and the users' waiting time.

Keywords: Foreign accent conversion,, Speech synthesis, Computer-assisted pronunciation training

1. Introduction

Foreign accent conversion (FAC) [1] aims to create a new voice that has the voice identity of a given L2 speaker and the accent of an L1 speaker. In pronunciation training, FAC can serve as a "golden speaker" for the L2 speaker to practice with: their own voice, but with a native accent [1, 2, 3, 4]. FAC also finds applications in movie dubbing [5], personalized text-to-speech (TTS) synthesis [6, 7], and improving speech recognition performance [8]. A variety of techniques have been proposed to perform FAC, including voice morphing [1, 9, 10], frame pairing [11, 12], articulatory synthesis [13, 14], and sequence-to-sequence (seq2seq) modeling [15, 16]. However, previous FAC approaches have two major limitations. First, they operate in a one-to-one fashion, i.e., they require training a separate model for each pair of L1 and L2 speakers. Second, they need a considerable amount of speech data (∼1,000 utterances) for each L2 speaker. Thus, when using these conventional FAC methods in real-world applications such as pronunciation training, L2 learners need to record a large number of utterances and then wait for a dedicated model to be trained, which can be tedious and demotivating.

To address this issue, we propose Accentron, a zero-shot learning [17] approach to FAC that can synthesize speech for arbitrary L2 speakers who were unseen during training. Accentron consists of four independent models: (1) a speaker-independent acoustic model that captures the linguistic content of an L1 utterance as a sequence of *bottleneck feature vectors*, (2) a speaker encoder that captures the voice identity of the L2 speaker, denoted as a *speaker embedding*, (3) an accent encoder that captures the desired L1 accent, denoted as an *accent embedding*, and (4) a seq2seq model that generates a Mel-spectrogram from the sequence of bottleneck features, conditioned on the desired speaker and accent embeddings. These components can be trained independently, at which point the system can generate accent conversions to arbitrary L2 speakers given a few seconds of audio (i.e., enough speech to compute a speaker embedding), without the need to have any model re-training or adaptation process.

To our knowledge, ours is the first work to apply zero-shot learning for the task of FAC. Though zero-shot learning has been used for voice conversion [18, 19, 20, 21] and voice cloning [22, 23], previous studies [18, 19, 20, 21, 22, 23] have focused exclusively on manipulating voice identity, ignoring the speaker's accent, which holds important cues to speaker recognition [24] and speech perception [25, 26, 27, 28]. Incorporating accent into the conversion process requires changes to the conventional encoder-decoder structure of sequence-to-sequence (seq2seq) models for voice conversion. Our encoder takes a sequence of L1 bottleneck feature vectors as the input, and produces a hidden representation sequence. In a conventional voice conversion system [18, 19, 29, 30, 31, 32, 33, 34], this hidden representation sequence is then concatenated with the speaker embedding of the target speaker. In our case, however, the system also concatenates the accent embedding, which is treated as an additional independent and controllable factor during synthesis. The combined bottleneck/speaker/accent embedding is consumed by a decoder coupled with a location-sensitive attention mechanism [35]. During each decoding step, the decoder autoregressively predicts a Mel-spectrogram frame based on the output from the previous decoding step and a context vector produced by the attention mechanism. Finally, the output Mel-spectrogram is converted back into a waveform through either the Griffin-Lim algorithm [36] or a separately trained vocoder (e.g., WaveNet [37], WaveRNN [38]).

We thoroughly evaluated Accentron on the L2-ARCTIC corpus [39]. First, we visualized the speaker and accent embedding distributions for the accent-converted speech and natural speech, and the results show that Accentron syntheses can successfully capture the L2 voice identity along with an L1 accent. Second, we conducted a series of listening tests under two different settings: (1) a *standard* FAC setting, where the test L2 speakers were available during training, and (2) a *zero-shot* FAC setting, which assumes that the test L2 speakers were not available during training. Accentron achieves 27% relative improvement in accentedness while retaining the acoustic quality and voice identity, compared to two state-of-the-art FAC systems in standard FAC settings. In addition, Accentron showed no performance degradation when tested under zero-shot FAC setting.

The manuscript is organized as follows. Section 2 reviews prior approaches to foreign accent conversion, many-to-many voice conversion, and sequence-to-sequence models. Section 3 describes the architecture of Accentron in detail. Section 4 provides the experimental setup, including the corpora and implementation details. Section 5 presents the objective and subjective evaluations of Accentron and analyzes the results. We discuss the implications of the results in Section 6. Lastly, we conclude the findings of this work and point out potential future directions in Section 7.

2. Related work

2.1. Foreign accent conversion

The problem of foreign accent conversion was first formulated by Felps *et al.* [1] as the means to provide implicit feedback in computer assisted pronunciation training. Early approaches [14, 40, 41, 42] involved building an articulatory synthesizer for the L2 speaker. The articulatory synthesizer was trained to map the L2 speaker's articulatory trajectories (e.g., tongue and lip movements) into his or her acoustics features (e.g., Mel Cepstra) using GMMs [14], unit-selection models [40], and DNNs [41]. Once the synthesizer was built, it could be driven with articulatory trajectories from an L1 speaker to synthesize FAC speech. However, these approaches were impractical for pronunciation training since collecting articulatory data is expensive and requires specialized equipment[2]. As a result, later work on FAC has focused on acoustic methods, since they only require recording speech with a microphone. Previous acoustic methods can be grouped into two categories: frame-pairing methods [11, 12] and seq2seq methods [15, 16]. As the name suggests, frame-pairing methods builds a lookup table where L1 and L2 speech frames are paired based on their similarity, and then use a statistical model (e.g., a GMM) to convert from L1 frames to their corresponding L2 frames in the lookup table. Aryal and Gutierrez-Osuna [11] first proposed a technique to pair L1-L2 frames based on their acoustic similarity (in MFCC space), after applying vocal tract length normalization to reduce global differences between the L1 and L2 spectra. Following this, Zhao *et al.* [12] argued that the L1 and L2 frames should be paired based on their linguistic content, and consequently, they used Phonetic-PosteriorGram (PPG) similarity instead of MFCC similarity. More recently, methods based on seq2seq models have been shown to significantly improve synthesis quality. In a previous study [15], we proposed a seq2seq PPG-to-Mel synthesizer for FAC. During training, the system learns a seq2seq model to convert PPGs to Mel-spectra extracted from utterances of an L2 speaker. During inference, the model is driven by PPGs extracted from a reference L1 utterance, which then produces FAC synthesis. In related work, Liu *et al.* [16] proposed a novel recognizer-synthesizer framework to remove the need for a reference L1 utterance. Their system trained a speaker recognizer, a multi-speaker text-to-speech (TTS) model, and an accent-sensitive automatic speech recognition (ASR) system. During inference, they feed L2 Mel-spectra to the ASR system with the corresponding accent, and then feed the output of the ASR system and the L2 speaker embedding to the multi-speaker TTS model to generate accent-converted utterances. These seq2seq model based FAC approaches can convert segmental and prosody features simultaneously, producing syntheses with higher speech naturalness and acoustic

[2] Articulatory measurements can be performed via electromagnetic articulography [40], ultrasound imaging [43], palatography [44], and more recently real-time MRI [45].

quality.

2.2. *Many-to-many voice conversion*

Foreign accent conversion is related to the more general problem of voice conversion (VC) [46, 47], which aims to synthesize a voice that has the linguistic content of an utterance from a source speaker and the voice identity of a target speaker. Traditional VC approaches use GMMs [48, 49], sparse representations [50, 51], and DNNs [52, 53, 54, 55, 56, 57, 21, 58] to transform the spectra from a source speaker to that of the target speaker. These methods require training a separate model for each pair of source-target speakers. More recently, several studies have proposed many-to-many VC approaches based on Variational Autoencoders (VAE) [19, 29, 30, 31, 20, 59] and the PPG-to-speech synthesizer [18, 32, 33, 60, 61]. Hsu *et al.* [19, 62] first proposed to use a VAE for many-to-many VC. During training, the encoder learns a speaker-independent latent embedding from input speech signals, and the decoder reconstructs the input speech signals given the latent embedding and the corresponding speaker embedding. During inference, the speaker embedding is replaced with that of a target speaker to produce a VC synthesis. A number of subsequent studies have been conducted to improve performance through various techniques, such as auxiliary classifiers [20], WaveNet vocoder adaption [59], and discrete latent space [29, 63]. Other studies [18, 32, 33] have used a PPG-to-speech synthesizer approach to perform many-to-many VC. The PPG-to-speech synthesizer is a neural network that takes PPGs as an input, and predict spectra conditioned on the speaker embedding of the target speaker. Early many-to-many VC models used one-hot vectors as the speaker embedding due to its simplicity, but recent studies [30, 18, 32, 33] have used learned speaker embeddings (e.g., i-vector [64], d-vector [65]) to generalize to unseen speakers, which make it possible to perform VC in a zero-shot fashion.

2.3. *Seq2seq models*

The seq2seq model was originally proposed by Sutskever *et al.* [66] for machine translation. The seq2seq model usually has an encoder-decoder architecture. The encoder learns a hidden representation sequence from an input sequence, and the decoder learns to autoregressively generate the output sequence given the hidden representation. To capture contextual information and handle length mismatches between the input and output sequences, an attention mechanism [67] is added between the encoder and the decoder. In recent years, there has been growing interest in applying seq2seq model to speech synthesis. Wang *et al.* [68] first proposed a seq2seq based TTS synthesizer (Tacotron), which significantly improved the acoustic quality of the syntheses over previous methods. Following this, Shen *et al.* [69] proposed Tacotron2, which further improved the acoustic quality of Tacotron by using a novel model architecture and a WaveNet vocoder. Jia *et al.* [22] extended Tacotron2 to voice cloning by conditioning a speaker embedding on the decoder. Besides, Biadsy *et al.* [8] and Jia *et al.* [70] also explored the use of seq2seq model in end-to-end speech-to-speech transformation, which avoids the need of multi-stage recognizer-synthesizer framework in related tasks, such as hearing-impaired speech synthesis and speech-to-speech translation. Seq2seq model has also been applied to voice conversion [56, 60, 61] and foreign accent conversion [15, 16], which significantly improved the performance on these tasks compared to conventional approaches.

3. Methods

Accentron consists of four modules: (1) a speaker-independent acoustic model that generates a linguistic representation of an utterance, (2) a speaker encoder that captures the voice identity of the desired speaker, (3) an accent encoder that captures the desired accent, and (4) a seq2seq model that consumes the previous three representations to synthesize Mel-spectrogram for an arbitrary L2 speaker.

The workflow for training Accentron is shown in Figure 1(a). The acoustic model, speaker encoder, and accent encoder are trained separately, and then are used as feature extractors for the seq2seq model. The seq2seq model is trained on a parallel corpus with multiple L1 and L2 speakers, capturing the

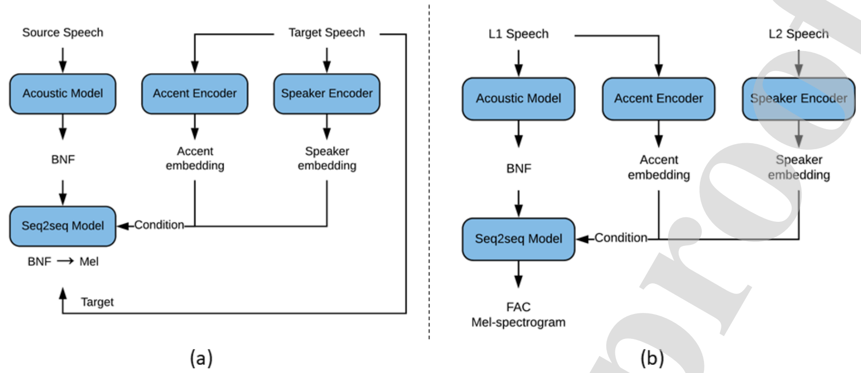


Figure 1: (a): Overall training workflow of Accentron. (b): Overall inference workflow of Accentron. Source: a selected reference L1 speaker, Target: any L1/L2 speaker, BNF: bottleneck feature, L1: native, L2: non-native. Each of the modules is trained independently.

voice characteristics of different speakers and accents. In what follows, we define a *"source"* speaker to be a selected reference L1 speaker, and a *"target"* speaker to be any L1/L2 speaker. To train the seq2seq model, we pair the source speaker with each target speaker. Then, for each pair of speakers, we feed source utterances to the speaker-independent acoustic model to extract bottleneck features (BNFs), which we assume capture only the linguistic content. Next, we feed an utterance from the target speaker to the speaker encoder and the accent encoder, which extract their speaker embedding and accent embedding, respectively. Finally, we train the seq2seq model to convert the source BNFs to the target Mel-spectrogram, conditioning on the target speaker's speaker and accent embeddings.

The workflow during inference is illustrated in Figure 1(b). Accentron requires a source utterance from an L1 speaker and an utterance from the L2 speaker. First, we extract BNFs and accent embedding from the L1 utterance, which encode the desired linguistic content and native accent, and a speaker embedding from the L2 utterance, which encodes the desired voice identity of the L2 speaker. Then, we pass the L1 BNFs, L1 accent embedding, and L2 speaker embedding to the seq2seq model, which generates the accent-converted Mel-spectrogram. Finally, the Mel-spectrogram is converted back to a waveform, in our case using a separately trained WaveRNN [38], though other vocoders or the Griffin-Lim algorithm[36] can also be used.

3.1. Speaker-independent acoustic model

To capture the linguistic content of an utterance, we use the output of the last hidden layer of a speaker-independent acoustic model (AM) as BNFs, rather than the output of the final layer of the AM, which represent the PPGs (i.e., the probabilities of each senone/tri-phone). BNFs contain similar linguistic information as PPGs but have much lower dimensionality (e.g., Senone-PPG: 6,024 dimensions; BNF: 256 dimensions), which avoids the need to perform dimensionality reduction in the seq2seq model.

Our AM is based on a Factorized Time Delayed Neural Network (TDNN-F) [71, 72], a feed-forward neural network acting as a sequential classifier. Given an input acoustic feature vector (i.e., 40-dimensional MFCCs), the TDNN-F produces the probabilities of the vector belonging to each senone/triphone (6,024 senones). The TDNN-F takes time-delayed input frames as side inputs to its hidden layers to model long-term temporal dependencies, concatenated with a 100-dimensional i-vector [64] of the corresponding speaker[3]. Additionally, the TDNN-F uses factorized layers with semi-orthogonal constraints as hidden layers and dilated connections between hidden layers, which are more efficient during training and inference than recurrent layers due to their feed-forward nature [71]. The TDNN-F model is composed of five hidden layers. Each of the first four hidden layers has 1,280 neurons, followed by ReLU activation and batch normalization [73], whereas the last hidden layer has 256 neurons, corresponding to the dimensionality of the BNFs. We train the model through a supervised 6,024-way senone classification task. To promote that the AM

[3] As noted by Peddinti *et al.* [72], this allows the model to capture both speaker and environment specific information, which is useful for neural network adaption.

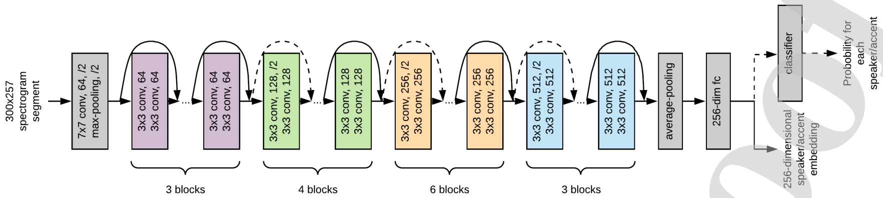

300x257 spectrogram segment
7x7 conv, 64, /2 max-pooling, /2
3x3 conv, 64 3x3 conv, 64
3x3 conv, 64 3x3 conv, 64
3x3 conv, 128, /2 3x3 conv, 128
3x3 conv, 128 3x3 conv, 128
3x3 conv, 256, /2 3x3 conv, 256
3x3 conv, 256 3x3 conv, 256
3x3 conv, 512, /2 3x3 conv, 512
3x3 conv, 512 3x3 conv, 512
average-pooling
256-dim fc
classifier
Probability for each speaker/accent
256-dimensional speaker/accent embedding
3 blocks
4 blocks
6 blocks
3 blocks

Figure 2: Speaker/accent encoder model architecture. The model is based on ResNet-34 [75]. Each convolution block is illustrated as the kernel size and channel numbers. "/2" means the layer divides the spatial resolution by 2.

produces speaker-independent BNFs, we train the model on speech data from several thousands of speakers (Librispeech corpus [74], 2,484 native English speakers; see Section 4.1).

3.2. Speaker and accent encoders

We use two independent encoders to compute the desired voice identity and accent. The speaker encoder is built as a speaker recognition model, which is trained to determine the identity of a speaker from an input utterance, whereas the accent encoder is based on an accent recognition model, which is trained to recognize accent/dialect patterns (e.g., pronunciation and prosody). For this work, we use a convolutional neural network (CNN) based on ResNet-34 [75] for both the speaker encoder and the accent encoder, following a previous speaker-recognition study [76]. We use the same CNN architecture for both models, so here we provide a detailed workflow only for the speaker encoder; the training and inference workflows of the accent encoder can be derived similarly.

The architecture of the speaker encoder is shown in Figure 2. The model takes 300×257 in time×frequency magnitude spectrogram segments as inputs. The inputs are first fed to a convolution layer containing 64 7×7 kernels with 2×2 stride, followed by a 2×2 max-pooling layer. These layers decrease the spatial resolution of the feature maps, reducing model complexity and improving training speed. On top of them, there are 16 convolution residual blocks, which extract more abstract features. Each convolution block consists of two convolution layers with 3×3 kernels. The first convolution layer in each block has 2×2 stride to further decrease the spatial resolution of the feature maps. More importantly, each block has a skip connection as an alternative path to avoid gradient vanishing in a very deep model. The 16 convolution blocks have different numbers of kernels, as highlighted in different colors in Figure 2 (Purple: 64 kernels; Green: 128 kernels; Orange: 256 kernels; Blue: 512 kernels). Next, there is an average pooling layer that produces a 512-dimensional vector, followed by a 256-dimensional fully-connected layer. All the layers are followed by ReLU activations and batch-normalization [73].

The model is trained through a supervised speaker-classification task. During training, a classifier on top of the 256-dimensional fully-connected layer produces the probabilities that the segment belongs to each speaker. The network is then optimized by minimizing the cross-entropy loss between the prediction and the target speaker label. During inference, we discard the final classifier layer and directly use the 256-dimensional bottleneck feature as the segment-wise speaker embedding. To obtain utterance-level speaker embeddings for a speaker that does not appear during training, we divide each test utterance into 300-frame segments with a 150-frame overlap using a sliding window, and then we compute the average of these segment-wise embeddings as the utterance-level speaker embedding (i.e., d-vectors [65]).

3.3. Seq2seq foreign accent conversion model

Our seq2seq model is inspired by the text-to-speech Tacotron2 model [69]. As shown in Figure 3, the seq2seq model has an encoder-decoder architecture. During training, inputs to the network consist of a triplet: (1) a sequence of BNFs from the source (L1) speaker, $\mathbf{x} \in \mathbb{R}^{T_i \times D_{BNF}}$, (2) a speaker embedding of the target (L1 or L2) speaker $\mathbf{s} \in \mathbb{R}^{D_{speaker}}$ extracted from the speaker encoder, and (3) the accent embedding of the target speaker $\mathbf{a} \in \mathbb{R}^{D_{accent}}$ extracted from the accent encoder. T_i is the length of the sequence $\mathbf{x}$. D_{BNF} is the dimensionality of the BNFs (e.g., 256 in this work). $D_{speaker}$ and D_{accent} are the

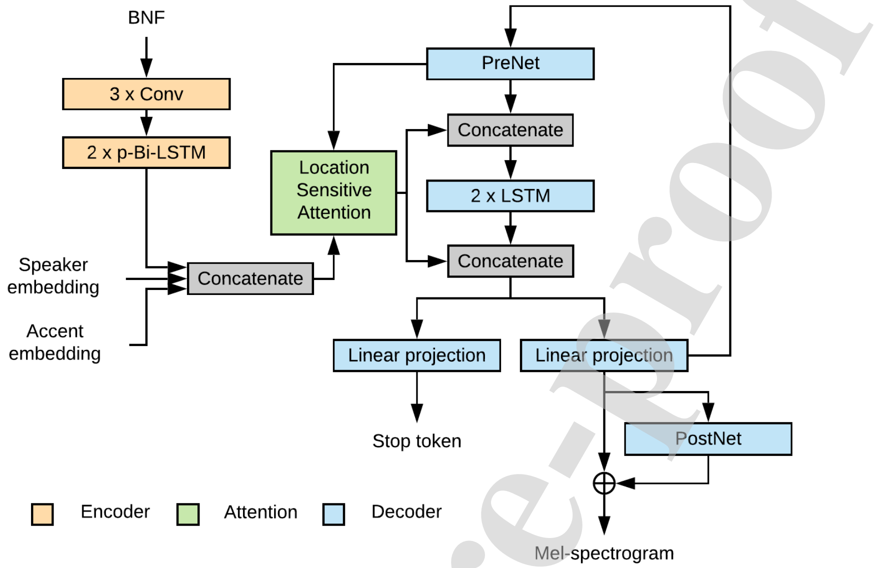


Figure 3: The seq2seq model in Accentron.

dimensionalities of the speaker embedding **s** and accent embedding **a**, respectively (both of them are 256 in this work). The ground-truth target of the model is a sequence of Mel-spectrogram frames $\mathbf{y} \in \mathbb{R}^{T_o \times D_{Mel}}$, where T_o is the length of the sequence and D_{Mel} is the number of Mel-filterbanks (e.g., 80 in this work). First, the encoder accepts a BNF sequence **x** and produces a hidden representation **h**:

$$\mathbf{h} = \text{Encoder}(\mathbf{x}) \tag{1}$$

Then, to condition the decoder on the voice identity and the accent of the target speaker, we concatenate the target speaker's speaker embedding and accent embedding to the hidden representation:

$$\mathbf{h_{concat}} = [\mathbf{h}, \mathbf{s}, \mathbf{a}] \tag{2}$$

Finally, the decoder autoregressively predicts the Mel-spectrogram of the target speech using the attention context computed based on the concatenated hidden

Table 1: Hyperparameters of the seq2seq foreign accent conversion model.

Block	Component	Parameters
Inputs	BNF	256-dimensional
	Speaker d-vector	256-dimensional
	Accent d-vector	256-dimensional
Encoder	3 × Conv layers	512 5×1 kernel with 1×1 stride; ReLU; batch norm
	2 × p-Bi-LSTM	256 cells per direction;
Attention	Location-sensitive attention	128-dim attention context; 32 31×1 attention conv kernels
Decoder	PreNet	2 × Fully-connected layer ; 256 neurons; ReLU
	2 × LSTM	1024 cells
	Linear (Mel)	1 × Fully-connect layer; 80 units; no activation
	Linear (stop token)	1 × Fully-connect layer; 1 unit; no activation

	PostNet	5 × Conv layers;
		512 5×1 kernel with 1×1 stride;
		tanh; batch norm

Outputs	Mel-spectrogram	80-dimensional
	Stop token	2-dimensional

representation:

$$\hat{\mathbf{y}}^t = \text{Decoder}(\hat{\mathbf{y}}^{t-1}, \mathbf{h}_{\mathbf{concat}}) \tag{3}$$

where $\hat{\mathbf{y}}^t$ is the t-th frame of the predicted Mel-spectrogram.

During inference, the inputs to the network are also a triplet: (1) a sequence of BNFs from an L1 speaker, $\mathbf{x}$, (2) a speaker embedding of an L2 speaker, $\mathbf{s_{L2}}$, and (3) an accent embedding of an L1 speaker, $\mathbf{a_{L1}}$. The network produces a hidden representation $\mathbf{h}$, concatenates it with $\mathbf{s_{L2}}$ and $\mathbf{a_{L1}}$, and feeds it to the decoder to produce the predicted FAC Mel-spectrogram. We describe each component in the following subsections. The hyper-parameters of each component are summarized in Table 1.

3.3.1. Encoder

The encoder converts a BNF sequence to a hidden representation sequence. The original text-to-speech Tacotron2 encoder contains three 1-dimensional convolution layers and one Bidirectional Long Short-Term Memory (Bi-LSTM) layer. However, in our case, the inputs of the seq2seq model are BNF sequences instead of text embeddings, which are usually significantly longer. To capture the high-level phonetic and contextual information in an input BNF sequence, we replace the LSTM layer in the encoder with two pyramidal Bidirectional LSTM (p-Bi-LSTM) layers [77]. Each p-Bi-LSTM reduces the time resolution by a factor of two, and therefore our encoder produces four times shorter hidden representation sequences compared with the input sequences. A convolution layer has 512 kernels with 5×1 shape in time×frequency and 1×1 stride, followed by ReLU activation and batch normalization. Each convolution kernel spans five BNF frames, which models the local context information. A p-Bi-LSTM layer has 256 cells in each direction, followed by ReLU activation and batch normalization, producing a 512-dimensional hidden representation sequence.

3.3.2. Decoder

The decoder is an autoregressive recurrent neural network coupled with a local sensitive attention mechanism [35]. The decoder accepts the concatenated hidden representation sequences as inputs, and produces an 80-dimensional Mel-spectrogram as the prediction of the L2 speech. During each decoding step, the predicted Mel-spectrogram frame from the previous step is first passed into a pre-net that has two 256-dimensional fully-connected layers with ReLU activations:

$$\mathbf{q}^t = \mathrm{PreNet}(\hat{\mathbf{y}}^{t-1}) \tag{4}$$

The pre-net acts as an information bottleneck, which is essential for learning attentions [69]. Next, the location-sensitive attention mechanism computes a 128-dimensional attention context vector $\mathbf{c}^t$ based on the pre-net output, the concatenated hidden representations, and the attention context from the previous step:

$$\mathbf{c}^t = \text{Attention}(\mathbf{q}^t, \mathbf{h}_{\mathbf{concat}}, \mathbf{c}^{t-1}) \tag{5}$$

Following this, the pre-net output is concatenated with the context vector and fed to two unidirectional LSTM layers with 256 cells. Then, the output of the second LSTM layer is concatenated again with the context vector $\mathbf{c}^t$ and passed through an 80-unit linear layer to make a prediction of the 80-dimensional L2 Mel-spectrogram frame:

$$\hat{\mathbf{y}}^t_{\mathbf{pre}} = \text{Linear}(\text{LSTM}(\mathbf{q}^t, \mathbf{c}^t), \mathbf{c}^t) \tag{6}$$

More importantly, the network also predicts if the generating process should stop at the current decoding step at the same time, i.e., a stop token $\hat{\mathbf{t}} \in \mathbb{R}^{T_o}$. Finally, to incorporate the spectral residual and improve synthesis quality, the predicted Mel-spectrogram is passed through a post-net consisting of five convolution layers to predict the residual. Each of these layers has 512 kernels with 5×1 shape and 1×1 stride, followed by *tanh* activation and batch normalization. The residual is added back to the original prediction to form the final prediction:

$$\hat{\mathbf{y}}^t = \hat{\mathbf{y}}^t_{\mathbf{pre}} + \text{PostNet}(\hat{\mathbf{y}}^t_{\mathbf{pre}}) \tag{7}$$

The model is optimized by minimizing the Euclidean distance between the target Mel-spectrogram and the prediction before/after the post-net. We also jointly minimize an extra cross-entropy loss to learn the stop token for model inference.

$$L = ||\hat{\mathbf{y}}^t_{\mathbf{pre}} - \mathbf{y}||^2_2 + ||\hat{\mathbf{y}}^t - \mathbf{y}||^2_2 + \lambda \text{CrossEntropy}(\hat{\mathbf{t}}, \mathbf{t}) \tag{8}$$

where $||\cdot||^2_2$ is the Euclidean distance; $\hat{\mathbf{t}}$ is the sequence of predicted stop tokens, and $\mathbf{t}$ is the sequence of target stop tokens; λ is the weight controlling the relative importance of the cross-entropy loss. Additionally, we use the teacher-forcing procedure during training by feeding in the correct output instead of the predicted output on the decoder side, which has been shown to improve the efficiency of the model training [78].

4. Experimental setup

4.1. Acoustic model

We trained the TDNN-F acoustic model using the Librispeech corpus [74], which consists of 960 hours of 16 kHz audiobook speech data produced by 2,484 native English speakers, the majority being American English. The training set consists of two "clean" subsets and a "noisy" subset[4]. We used both sets in training to ensure that the BNF was speaker-independent. In addition, we used a subset (200 hours) of the training set to train the i-vector extractor. We implemented the training following the official "tdnn_1d" recipe of the TDNN-F model in Kaldi[5]. The trained model achieves 3.76% word error rate (WER) on Librispeech's test-clean subset and 8.92% WER on the test-other subset.

4.2. Speaker encoder

We trained the speaker encoder using the VoxCeleb1 corpus [79], which contains 153,516 utterances of 16 kHz speech produced by 1,251 speakers. Specifically, we used the training set from the official identification split, which is comprised of 138,316 utterances (~300 hours) from these speakers. We extracted 257-dimensional magnitude spectrograms with a 25ms window and 10ms shift. We trained the model on a single NVIDIA Tesla V100 GPU with a batch size of 128. We used Adam Optimizer with an initial learning rate of 10^{-2}, which was annealed down to zero following a cosine schedule [80]. The trained model achieves 81.34% Top-1 accuracy and 94.49% Top-5 accuracy on the official VoxCeleb1 identification testing set.

[4] We use the term "noisy" subset to refer to the test-other test subset in the LibriSpeech corpus. LibriSpeech has two test subsets: test-clean and test-other. The recordings in test-other subset have significantly higher background noise level than the test-clean subset.

[5] `https://github.com/kaldi-asr/kaldi/blob/master/egs/librispeech/s5/local/chain/tuning/run_tdnn_1d.sh`

4.3. Accent encoder

We trained the accent encoder using the Speech Accent Archive dataset [81], which consists of recordings of the “Please call Stella” paragraph [81] produced by speakers in 386 native and non-native English accents. For most of the accents, however, the number of speakers is limited, which may degrade the performance of the accent encoder. To address this issue, we selected a subset of accents where each accent has at least 30 speakers. The resulting subset has 18 accents[6], with an average of 107 speakers in each accent. The total length of the selected subset is around 16 hours. We randomly selected 90% utterances from each accent as the training set and used the remaining 10% utterances as the testing set. The audio waveforms in the original dataset have 8 kHz sampling rate. To match it with other modules, we resample them to 16 kHz. Other configurations were the same as that for speaker recognition. Our trained model achieves 79.36% Top-1 accuracy and 95.42% Top-5 accuracy on the testing set.

4.4. Seq2seq foreign accent conversion model

To evaluate the proposed approach, we conducted experiments with the ARCTIC [82] and L2-ARCTIC corpora [39]. We used four native English speakers from ARCTIC (BDL, RMS, SLT, CLB) and all 24 non-native English speakers from L2-ARCTIC. For each speaker, we divided their utterances into three subsets: a training set of 1,032 utterances (∼1 hour of speech), a validation set of 50 utterances, and a testing set of 50 utterances. During training, we set BDL as the source speaker and paired it with all 28 speakers, including himself. During inference, we used both BDL (male) and CLB (female; used as an unseen L1 speaker for the zero-shot FAC setting) as the native reference speakers, and we performed FAC on four L2 speakers whose first languages were different:

[6] These 18 accents were Arabic, Cantonese, Dutch, English, Farsi, French, German, Hindi, Italian, Japanese, Korean, Mandarin, Polish, Portuguese, Russian, Spanish, Turkish, and Vietnamese.

NJS (Spanish, female), TXHC (Mandarin, male), YKWK (Korean, male), and ZHAA (Arabic, female).

The original L2-ARCTIC audio waveforms have a 44.1 kHz sampling rate, so we resampled them to 16 kHz to match the ARCTIC recordings. We extracted 80-dimensional Mel-spectrogram with a 25ms window and 10ms shift. Following the same frame shift, we extracted BNFs for each utterance using the acoustic model (Section 3.1). In addition, we extracted utterance-level speaker and accent d-vectors from the speaker encoder and accent encoder, respectively. We implemented the model using TensorFlow [83] and trained it on a single NVIDIA Tesla V100 GPU. The hyperparameter λ (eq. 8) in the loss function was set to 0.005 empirically. We set the batch size to 48, and we used an Adam Optimizer with an initial learning rate of 10^{-3}, which was then annealed down to 10^{-5} following exponential scheduling. The model converged after 200,000 steps, and the entire training time was around 100 hours.[7] During model inference, we used a separately trained speaker-independent WaveRNN vocoder to invert the Mel-spectrogram back to the time-domain waveform. We trained the WaveRNN model on the Librispeech dataset.

5. Results

We have validated Accentron through a series of objective and subjective evaluations. For the objective evaluation, we used t-distributed stochastic neighbor embedding (t-SNE) [84] to analyze the distribution of speaker embeddings and accent embeddings for the original speech and the accent-converted speech. This allowed us to examine whether the two encoder networks can decouple speaker identity and accent. For the first of two sets of subjective evaluations, we tested the system when the test L1 and L2 speakers were seen during training (*standard* FAC setting) and compared it against two state-of-the-art FAC systems [15, 16]. We also tested whether our system could be used in the reverse

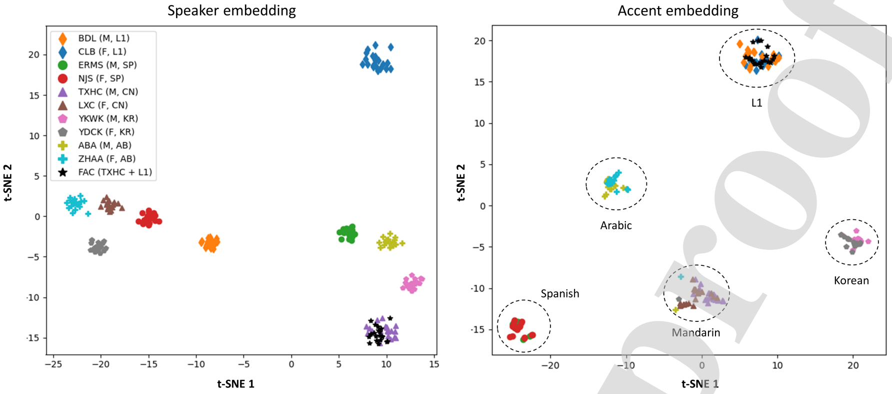
[7] Audio samples from this work can be found at `https://shaojinding.github.io/samples/accentron/`. We intend to open-source our code after this work has been peer-reviewed.

direction, i.e., to impart a non-native accent to a native speaker's voice. In the second set of subjective evaluation, we explored the effectiveness of Accentron when the test L1 and L2 speakers were unseen during training (*zero-shot* FAC setting). Further, we characterized the performance of Accentron as a function of the number of available L2 test utterances during inference (i.e., which are used to extract the L2 speaker's voice identity footprint), and we compared it against a system that uses these utterances to fine-tune a pre-trained FAC system to understand the tradeoff between using L2 test utterances to compute a speaker embedding (zero-shot learning) and using them to refine a pre-trained model (fine-tuning).

5.1. Objective evaluation: speaker and accent embedding spaces

We used t-SNE [84] to visualize the speaker and accent embedding spaces[8], which helped provide a qualitative and intuitive explanation of how our proposed system operates. First, we visualized the speaker and accent embeddings of 20 FAC utterances for TXHC, a male Mandarin speaker. We used the system in the zero-shot condition when both L1 and L2 speakers were unseen (see Section 5.3.1) to generate the syntheses, since it acts as a performance lower-bound for all our systems, and it can also provide insights for zero-shot FAC. We also plotted the embeddings of natural speech from ten L1 and L2 speakers as references (20 utterances for each speaker). Results are shown in Figure 4, where we use colors and shapes to represent speaker and accent, respectively (e.g., orange denotes speaker BDL, and diamond denotes L1 accent). As shown, the speaker embeddings of utterances from the same speaker form a cluster, and the boundary between different clusters are clear. Similarly, accent embeddings from speakers with the same accent form a cluster. These two results indicate that the speaker and accent encoders are operating as expected. In terms of

[8] The axes in a t-SNE plot have no actual meaning. t-SNE projects high dimensional vectors into a lower dimensional plane (e.g., 2-D), such that similar points in the projection are also similar in the original space, and vice versa.



(a) (b)

Figure 4: Speaker and accent embedding visualization of FAC syntheses for TXHC using t-SNE. (a): speaker embedding; (b): accent embedding. Colors and shapes represent speaker and accent, respectively. Speakers in the legend are annotated with gender and accent. L1: native accent; SP: Spanish accent; CN: Mandarin accent; KR: Korean accent; AB: Arabic accent.

the FAC utterances, their speaker embeddings are distributed in the cluster of TXHC, which indicate that Accentron can generate speech that matches the voice identity of the target speaker, and the accent embeddings lie in a cluster of L1 speakers (BDL and CLB), which also indicate the Accentron can generate the desired accent. Together, these visualizations indicate that Accentron can successfully generate speech with the same voice identity as TXHC but with a native accent.

We also conducted t-SNE visualizations on a “reverse” FAC task [12, 85], where the goal was to synthesize speech with the voice identity of a given L1 speaker but with an L2 accent. This is a straightforward process in Accentron, since we only need to change the inputs of the seq2seq model to use an L2 accent embedding and an L1 speaker embedding during inference. For these visualizations, we synthesized a voice that had the voice identity of CLB but with an L2 (Mandarin) accent. As shown in Figure 5, the speaker embeddings of the reverse FAC syntheses lie in the same cluster as CLB utterances, whereas the

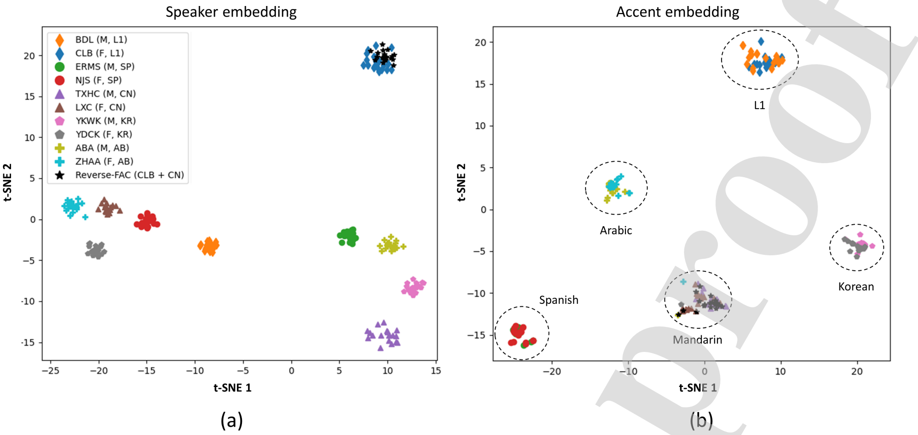


Figure 5: Speaker and accent embedding visualization of *reverse* FAC syntheses (CLB with a Mandarin accent) using t-SNE. (a): speaker embedding; (b): accent embedding. Colors and shapes represent speaker and accent, respectively. Speakers in the legend are annotated with gender and accent. L1: native accent; SP: Spanish accent; CN: Mandarin accent; KR: Korean accent; AB: Arabic accent.

accent embeddings lie in the same cluster as the two Mandarin speakers (TXHC and LXC), indicating that the reverse FAC syntheses have a voice identity of CLB and a Mandarin accent.

Another interesting question that we would like to investigate here is whether the speaker embedding also carries accent cues, as these two aspects are closely related in the recognition of speakers [25, 26, 69, 28]. If the speaker embedding is entangled with accent information, then there is the risk that Accentron would generate speech with an incorrect accent that is introduced by speaker embeddings. To examine this potential issue, we generated t-SNE visualizations of speaker embeddings for natural utterances from 16 speakers with 4 different accents in L2-ARCTIC corpus (See Figure 6 for details of the speakers and accents). Results in Figure 6 show no adjacency patterns among speakers sharing the same accent. Instead, t-SNE shows a distinct separation among speakers based on gender. These results suggest that speaker embedding mainly encodes information such as voice quality/timbre and pitch, and no evidence of potential

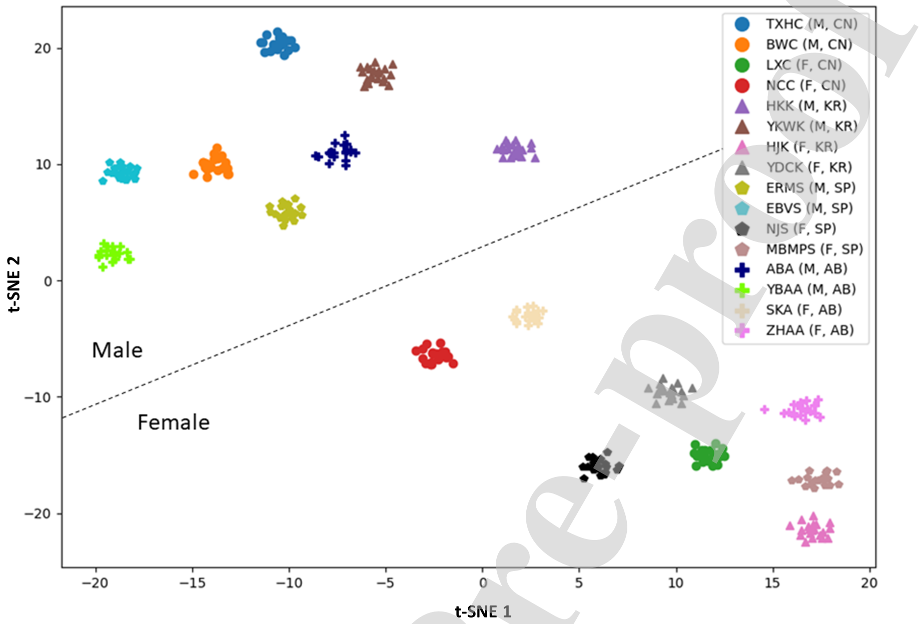


Figure 6: t-SNE visualization of the speaker embeddings from 16 speakers with 4 accents. Colors and shapes represent speaker and accent, respectively. . Speakers in the legend are annotated with gender and accent. L1: native accent; SP: Spanish accent; CN: Mandarin accent; KR: Korean accent; AB: Arabic accent.

entanglement between speaker and accent cues.

5.2. Subjective evaluations under standard FAC setting

We also evaluated Accentron through two sets of perceptual experiments. In the first set (this sub-section), we evaluated the system under the standard FAC setting, i.e., the test L1 and L2 speakers were seen during training. In the second set (Section 5.3), we evaluated the system under the zero-shot FAC setting, where the L1 speaker and/or the L2 speaker were unseen during training.

For the standard FAC setting, we used the union of the training sets of all 28 speakers to train the system. During inference, we used BDL as the L1 speaker, who then had been “seen” during training. First, we compared the proposed approach against two state-of-the-art FAC approaches:

- **Baseline1:** the FAC system proposed by Zhao *et al.* [15], a one-to-one FAC approach based on seq2seq model. This baseline system trains a seq2seq PPG-to-speech synthesizer for each L2 speaker, and drives the synthesizer with PPGs extracted from an L1 speaker. As such, the baseline system requires training separate models for each L2 speaker.

- **Baseline2:** the system proposed by Liu *et al.* [16], a reference-free many-to-many FAC approach based on a novel recognizer-synthesizer architecture. The system is trained on 105 speakers from CSTR VCTK dataset [86]. Audio samples were produced by feeding the test utterances through their system, which is provided as a courtesy by Liu *et al.* Due to the implementation differences between the systems, we conducted two post-processing steps to ensure a fair comparison. First, as the accent conversion model of Liu *et al.* was trained on VCTK speakers, the stop-token predictions on L2-ARCTIC test utterances are not robust, occasionally resulting in a few seconds of white noise at the end of speech in accent conversion syntheses. To solve this issue, we manually removed the trailing white noises in these test utterances. Second, we resampled the syntheses of Liu system from 22.05 kHz to 16 kHz to make the sampling rate be consistent with other systems.

We conducted listening tests through Amazon Mechanical Turk[9] to rate three perceptual attributes of the synthesized speech:

- **Accentedness:** : The test asked participants to rate the degree of foreign accentedness of each utterance in a 9-point scale (1-no foreign accent; 9-very strong foreign accent), which is commonly used in the pronunciation literature [87]. Participants were told that the native accent in this task was General American.

- **Acoustic quality:** The test asked participants to rate the acoustic qual-

[9] `https://www.mturk.com`

ity of each utterance through a standard 5-point Mean Opinion Score (MOS; 1-bad, 5-excellent).

- **Voice identity:** The test asked participants to rate the voice similarity between the FAC syntheses and the original L2 speech through a 14-point Voice Similarity Score (VSS) [88]. For each FAC-L2 utterances pair, participants were required to decide whether the two utterances were from the same speaker and then rate their confidence in the decision on a 7-point scale (1: not confident at all; 3: somewhat confident; 5: quite a bit confident; 7: extremely confident). The VSS was computed by collapsing the above two fields into a 14-point scale: -7 (definitely different speakers) to +7 (definitely the same speaker). To minimize the influence of accent, the two utterances had different linguistic content and were played in *reverse*, following [1].

Instructions were given in each test to help participants focus on the target speech attribute. For example, in the accentedness test, participants were asked to *"Try to ignore the audio quality (noise, distortions). Please focus only on the speaker's accent, for example, their pronunciation, rhythm, and fluency"*. Test utterances were randomly selected from our test set, and the presentation order was counter-balanced. Additionally, in each listening test, we included five calibration utterances to detect if participants were cheating. We excluded ratings of the calibration utterances from the data analysis [89]. We recruited 18 participants for each listening test. All participants resided in the United States, and they passed a qualification test that asked them to identify different regional accents in the United States.

5.2.1. Comparison to the baseline system

Accentedness. Participants rated 20 utterances per system (5 utterances for each test L2 speaker). These utterances shared the same linguistic content across all the systems to ensure a fair comparison. Additionally, participants also rated the same set of sentences from the original L1 and L2 speakers as a

Table 2: Accentedness (1-no foreign accent, 9-very strong foreign accent) results and acoustic quality (1-bad, 5-excellent) results under standard FAC setting. All the results are shown as average ± 95% confidence intervals.

System	**Accentedness**	**Acoustic quality**
Original L2	7.11 ± 0.21	3.67 ± 0.28
Original L1	1.06 ± 0.12	4.90 ± 0.10
Baseline1	4.63 ± 0.10	3.47 ± 0.14
Baseline2	6.25 ± 0.39	3.12 ± 0.13
Proposed	**3.39 ± 0.14**	**3.51 ± 0.15**

reference. Results are shown in the first column of Table 2. Accentron received significantly lower ratings of foreign accentedness (3.39) than the original L2 utterance (7.11), though not as low as those of the original L1 utterance (1.06). These results suggest that our proposed seq2seq FAC model can effectively reduce foreign accentedness from the L2 speech. Accentron also outperformed the two baseline systems (Baseline1: 4.63, 27% relative improvement, $p \ll 0.001$; Baseline2: 6.25, 46% relative improvement, $p \ll 0.001$).

Acoustic quality. As shown in the second column of Table 2, the proposed method achieved an MOS of 3.51, which is comparable to Baseline1 (3.47, $p > 0.5$) but significantly higher than Baseline2 (3.12, 13% relative improvement, $p \ll 0.001$). The original L1 speech received the highest MOS (4.90), followed by the original L2 speech (3.67). Note that the MOS ratings of Accentron are closer to those of the original L2 speech than to the original L1 speech, possibly due to native listeners confounding acoustic quality with intelligibility [1], and therefore, they may be influenced by the intelligibility and provide lower ratings for non-native speech. Thus, the proposed system achieves similar acoustic quality as the baseline systems (or better), but unlike them does not require training a separate model for each new test L2 speaker.

Voice identity. Participants rated 20 pairs of utterances per system (5 pairs of utterances for each test L2 speaker). Each pair consisted of a FAC utterance and an utterance randomly selected from the L2 speaker. Voice identity

Table 3: Voice identity results under standard FAC setting. Voice Similarity Score ranges from -7 (definitely different speakers) to +7 (definitely the same speaker). All the results are shown as average ± 95% confidence intervals.

System	Voice Similarity
Baseline1	5.05 ± 0.28
Baseline2	3.81 ± 0.29
Proposed (All pairs)	$\mathbf{5.05 \pm 0.31}$
Proposed (Intra-gender)	5.29 ± 0.30
Proposed (Inter-gender)	4.80 ± 0.35

results are shown in Table 3. Accentron achieved a 5.05 VSS, indicating that participants were "quite confident" that the FAC syntheses and the L2 speech were produced by the same speaker. These ratings are comparable to those of Baseline1 (5.05 VSS, $p > 0.5$) and significantly higher than those of Baseline2 (3.81 VSS, 33% relative improvement, $p \ll 0.001$). It is worth noting that the L1 speaker in Baseline1 had the same gender as the L2 speaker, whereas Accentron used the same L1 speaker for all L2 speakers. As a result, syntheses from Accentron included both intra (same)-gender FAC pairs and inter (different)-gender FAC pairs, the latter being more challenging due to the differences in prosody and pitch range. Although the VSS on inter-gender pairs (4.80) was lower than that on intra-gender pairs (5.29) and Baseline1, the difference was not significant ($p = 0.14$). These results suggest that the proposed system can generate FAC syntheses that greatly resemble the voice identity of L2 speakers of any gender, using a canonical reference L1 speaker.

5.2.2. Performance on reverse FAC

To evaluate Accentron on the reverse FAC task, we synthesized testing utterances using the accent embeddings from NJS, TXHC, YKWK, and ZHAA, and the speaker embedding from BDL. Table 4 shows the accentedness, acoustic quality, and voice identity results of the reverse FAC evaluation. Accentron received a 5.58 accentedness rating, much closer to that of the original L2 speech

Table 4: Accentedness (1-no foreign accent, 9-very strong foreign accent) results, acoustic quality (1-bad, 5-excellent) results, and voice identity results (-7-definitely different speakers, +7-definitely the same speaker) of *reverse* foreign accent conversion under standard condition. All the results are shown as average ± 95% confidence intervals.

System	**Accentedness**	**Acoustic quality**	**Voice Similarity (All pairs)**	**Voice Similarity (Intra-gender)**	**Voice Similarity (Inter-gender)**
Proposed	5.58 ± 0.35	3.24 ± 0.17	4.91 ± 0.34	5.11 ± 0.35	4.71 ± 0.32

(7.11) than to the original L1 speech (1.06), indicating that our approach was able to impart an L2 accent to utterances from an L1 speaker. Accentron also received a 3.24 MOS, significantly lower ($p = 0.02$) than the MOS of the "direct" FAC syntheses (3.51), a result that is likely due to the correlation between acoustic quality and intelligibility –see Section 5.2.1. Finally, Accentron received a 4.91 VSS, indicating that raters were "quite confident" that the reverse FAC syntheses and the L1 speech were produced by the same speaker; we found no significant differences between the voice identity ratings of reverse and direct FAC syntheses. Thus, we can conclude that Accentron can also operate in the reverse direction, generating non-native utterances with the voice identity of a native speaker.

5.3. Subjective evaluations under zero-shot FAC setting

In the second set of subjective evaluations, we evaluated the proposed system under the zero-shot FAC setting, where the L1 speaker and/or the L2 speaker were unseen during training. The zero-shot FAC setting is appealing for real-world applications since it requires minimal data from the target speaker. First, we compared the performance of Accentron when using seen/unseen L1 or L2 speakers during inference. Then, we characterize its performance as a function of the number of available L2 utterances.

5.3.1. Comparing different conditions in zero-shot foreign accent conversion

We considered four different conditions in this experiment, as summarized in Table 5. In condition SS, the L1 speaker and the L2 speaker were both

Table 5: The four conditions in zero-shot FAC experiment.

		L1 speaker	
		Seen	**Unseen**
L2 speaker	**Seen**	Condition SS	Condition US
	Unseen	Condition SU	Condition UU

seen during training. Note this condition is the same as the system evaluated in Section 5.2.1, so it serves as a best-case scenario. In condition US, the L1 speaker was unseen during training, and the L2 speaker was seen during training. In condition SU, the L1 speaker was seen during training, and the L2 speaker was unseen during training. Finally, in condition UU, the L1 speaker and the L2 speaker were both unseen during training. Thus, condition UU was the most challenging of the four.

To ensure that the test speakers were unseen during training, we trained four models using different training sets. In Condition SS, we used the same model as in the standard FAC condition. In Condition US, we excluded CLB from the training set and used it as the test L1 speaker. In Condition SU, we excluded the four test L2 speakers from the training set. In Condition UU, we excluded the four test L2 speakers and CLB from the training set, and we also used CLB as the test L1 speaker. For unseen L1/L2 speakers, we used the 50 utterances from the test set to generate the accent/speaker embedding. As before, we conducted three types of listening tests through Amazon Mechanical Turk to rate the accentedness, acoustic quality, and voice similarity of the synthesized speech. In addition, we kept the participants the same as those in the first experiment, so that the results are comparable between different experiments (e.g., participants in the accentedness test for the two experiments were the same).

Results from the accentedness, acoustic quality, and voice identity tests are shown in Table 6. We found no statistically significant differences between condition SS (best-case scenario) and the three more challenging conditions

Table 6: Accentedness (1-no foreign accent, 9-very strong foreign accent) results, acoustic quality (1-bad, 5-excellent) results, and voice identity results (-7-definitely different speakers, +7-definitely the same speaker) under zero-shot FAC condition. All the results are shown as average ± 95% confidence intervals.

System	Accentedness	Acoustic quality	Voice Similarity
Condition SS	3.39 ± 0.14	3.51 ± 0.15	5.05 ± 0.28
Condition US	3.33 ± 0.26	3.47 ± 0.13	4.99 ± 0.30
Condition SU	3.35 ± 0.25	3.50 ± 0.12	4.92 ± 0.28
Condition UU	3.30 ± 0.26	3.43 ± 0.12	4.59 ± 0.34

(US, SU, UU); $p > 0.5$ in all cases. This result suggests that Accentron has no trouble generalizing to unseen L1 or (and) L2 speakers during inference without any degradation in accentedness, acoustic quality, and voice identity.

5.3.2. Influence of the number of available L2 utterances

For practical FAC applications, it is important to understand the minimum amount of data needed from a target speaker. Requiring L2 learners to record a large amount of speech before they can hear their "golden speaker" voice can be tedious and demotivating. On the other hand, training the system with insufficient speech data might significantly degrade synthesis quality. To characterize the data requirements of Accentron under zero-shot FAC condition, we measured its performance of the UU codition (unseen L1 speaker and unseen L2 speaker) as a function of the number of available L2 utterances. We used the UU condition since it is the most flexible for real-world applications, and also the most challenging, which provides a lower bound of performance. For these experiments, we used 50 test L2 utterances to produce the speaker embedding during inference, and reduced the number from 50 to 1 (N = 50, 20, 10, 5, 1) and re-evaluated system performance. Results are shown in Table 7. Reducing the number of utterances from 50 to 1 has no impact on any of the three perceptual measures ($p > 0.5$ in all cases). These results indicate that as little as a single utterance (∼3 seconds of speech) is sufficient to generate accent conversions for a new unseen L2 speaker, with no impact on performance.

Table 7: Accentedness (1-no foreign accent, 9-very strong foreign accent) results, acoustic quality (1-bad, 5-excellent) results, and voice identity results (-7-definitely different speakers, +7-definitely the same speaker) with different numbers of available L2 (non-native) utterances during inference. All the results are shown as average ± 95% confidence intervals.

#L2 utterances	**Accentedness**		**Acoustic quality**		**Voice Similarity**	
	Proposed	**Fine-tuned**	**Proposed**	**Fine-tuned**	**Proposed**	**Fine-tuned**
50	3.30 ± 0.26	3.03 ± 0.24	3.43 ± 0.12	3.54 ± 0.11	4.59 ± 0.34	4.97 ± 0.27
20	3.30 ± 0.22	3.47 ± 0.18	3.45 ± 0.11	3.48 ± 0.11	4.68 ± 0.30	4.65 ± 0.23
10	3.34 ± 0.26	3.84 ± 0.18	3.44 ± 0.12	3.46 ± 0.11	4.59 ± 0.29	4.06 ±0.26
5	3.32 ± 0.23	4.58 ± 0.10	3.43 ± 0.11	3.38 ± 0.10	4.42 ± 0.33	3.49 ± 0.34
1	3.31 ± 0.25	4.72 ± 0.08	3.43 ± 0.12	3.24 ± 0.13	4.57 ± 0.29	3.73 ± 0.35

To some extent, the above result is to be expected since test utterances are only used to compute the speaker embedding. Thus, we also examined whether test utterances could instead be more beneficial if they were used to fine-tune a pre-trained FAC system. Starting with a pre-trained UU model, we fine-tuned the model on each unseen L1-L2 speaker pair (i.e., CLB-NJS, CLB-TXHC, CLB-YKWK, CLB-ZHAA) with N = 50, 20, 10, 5, 1 test utterances, resulting in 20 fine-tuned models (4 speakers; 5 models with different number of training utterances for each speaker) for the unseen L2 speakers. Results are also shown in Table 7. We observe performance degradations in all three measurements when reducing the number of utterances from 50 to 1. When there are 50 test utterances, the fine-tuned system shows a marginal improvement compared to the zero-shot model (i.e., without fine-tuning), though the differences are not statistically significant (Accentedness: 3.03 vs. 3.30, $p = 0.03$; Acoustic quality: 3.54 vs. 3.43, $p = 0.18$; Voice identity: 4.97 vs. 4.59, $p = 0.25$). When decreasing the number from 50 to 20, the fine-tuned system achieves comparable performance as the zero-shot system ($p > 0.5$). Surprisingly, however, fine-tuning the systems with fewer than 20 utterances degrades performance compared to the zero-shot model. In the extreme case (with only 1 utterance), the zero-shot model significantly outperforms the fine-tuned model in all three measurements (Accentedness: 4.72 vs. 3.31, $p \ll 0.001$; Acoustic quality: 3.24 vs. 3.43, $p = 0.01$; Voice identity: 3.73 vs. 4.57, $p \ll 0.001$). These results further speak of the robustness of Accentron in the zero-shot condition, and they also illustrate the tradeoff between zero-shot learning models and fine-tuning models in FAC.

6. Discussion

We have proposed Accentron, a zero-shot many-to-many speech synthesizer that can convert utterances from a source speaker to appear as if someone else, and with a different accent, had produced it. We thoroughly evaluated the system through a series of objective and perceptual listening experiments. Visualizations through t-SNE show that Accentron captures the target voice identity and accent, and that the speaker and accent embeddings are independent of each other and effectively summarize the speaker and accent characteristic of an utterance.

We evaluated Accentron in a standard FAC setting and compared it against two state-of-the-art baseline FAC systems. Compared to baseline 1 [15], Accentron achieves significantly better (i.e., lower) ratings of foreign accentedness, and similar acoustic quality and voice identity ratings. This is an important finding since baseline 1 builds a dedicated model for each pair of L1-L2 speakers, which one would expect would help capture voice identity more faithfully than Accentron's many-to-many mapping. Although baseline 1 and Accentron use the same backbone architecture for the seq2seq model, Accentron achieves significantly better (lower) ratings of accentedness. A possible explanation for this result is that baseline 1 is trained exclusively on L2 utterances. Thus, if the L2 speaker has systematic substitution or deletion errors (e.g., Mandarin speakers from certain areas systematically substitute /SH/ with /S/), the correct pronunciations will be missing in their utterances. Thus, when the baseline 1 model is driven by L1 BNFs during inference, it has to interpolate these missing pronunciations, which leads to noticeable segmental errors. In contrast, Accentron avoids this potential issue since it is trained using both L1 and L2 speech. Compared to baseline 2 [16], Accentron achieves better ratings in all three measurements (accentedness, acoustic quality, and voice identity). Though Accentron and baseline 2 both use a seq2seq architecture, Accentron has two additional components: an acoustic model and an accent encoder to extract the linguistic content and accent embeddings, respectively, from an L1 reference during inference. These two embeddings capture the L1 segmental and prosodic patterns, respectively, which are shown to be essential to achieve advanced FAC performance.

We also evaluated Accentron on a "reverse FAC" task, where the goal was to impart an L2 accent to a native utterance. Results on this task corroborates the t-SNE visualizations, and suggest that Accentron can also preserve an L2 accent and implant it into an L1 speaker's utterance. Combined, results from the direct and reverse FAC tasks indicate that the proposed system can disentangle linguistic content, voice identity, and accent in speech signals, instead of merely memorizing mappings between different speaker pairs.

Finally, we also evaluated Accentron in a zero-shot FAC setting. First, we compared all four combinations in which the L1 speaker and L2 speakers could have been seen/unseen during training (i.e., SS, SU, US, UU). And we found no significant differences among the four conditions in terms of accentedness, acoustic quality, or voice identity. Thus, Accentron performs equally well under a standard FAC setting (condition SS) and a zero-shot FAC setting (condition UU), which indicates that it generalizes to unseen L1 and L2 speakers without the need to re-train or fine-tune the model.

Can Accentron generalize to unseen accents? In principle, the Accentron architecture can naturally generalize to unseen accents –in the same way that it is able to model unseen speakers. However, this requires access to sufficient training data on a large number of accents. The Speech Accent Archive has hundreds of accents, but unfortunately each speaker only produces less than one minute of speech, which is insufficient for training the FAC model (empirically, at least one hour per speaker is needed). Its counterpart, the L2-ARCTIC corpus, has substantially more speech data per subject (around one hour), but only has six accents, which is insufficient to achieve reasonable transferability to an unseen accent. Thus, while we believe that Accentron can be used to generate unseen accents, this ability can only be truly assessed when more accented speech corpora become available.

Our results under both standard and zero-shot FAC setting indicate that Accentron can generate FAC synthesis with high-quality, and can achieve it with limited data from new L1 and L2 speakers (one utterance, or around three seconds of speech). This capability can dramatically simplify the deployment of pronunciation-training tools (our envisioned target application). When deploying conventional FAC models, one needs to design and implement the model training pipeline, as it requires training a dedicated model for each target L2 speaker. Running the training pipeline is usually resource- and time-consuming, and therefore, one has to include specific modules to manage the server's computational resources (e.g., an asynchronous queue was used in [3]). Such systems are hard to scale to an increasing number of users, due to the heavy resource demands. Instead, Accentron only requires running the model inference pipeline when producing FAC synthesis, and the inference can be accomplished in real time, thus essentially reducing the resource demands and simplifying the application design and deployment. In addition, in previous FAC studies (e.g., [11, 12, 15]), inter-gender conversion usually achieved inferior performance than intra-gender conversion, due to the mismatch in pitch ranges. As a result, when deploying these methods into practical applications, they have to use a reference L1 speaker that has the same gender as the target L2 speaker. In contrast, Accentron achieves similar voice identity ratings for intra-gender pairs and inter-gender pairs, which gives more flexibility when choosing the reference L1 speakers in the pronunciation-training tool. Finally, Accentron significantly reduces the data required for each new L2 speaker from hours to seconds. As a result, the L2 learners only need to go through a simple speaker enrollment process (recording several seconds of their speech), before practicing with their "golden speaker" voices, which makes Accentron an ideal system for performing pronunciation-training studies at scale.

7. Conclusion and future work

In this paper, we have proposed Accentron, a zero-shot learning system that can generate accent conversion for any L2 speaker (seen or unseen). Our proposed approach is in contrast to most of existing FAC approaches, which require building a separate model for each L2 speaker. The proposed approach first trains separate models to extract L1 bottleneck features, L1 accent embeddings, and L2 speaker embeddings. Then it uses a seq2seq model to transform L1 bottleneck features to accent-converted Mel-spectrogram, conditioned on an L1 accent embedding and L2 speaker embedding. Our results suggest that the system can successfully transform L1 speech to match the voice identity of an L2 speaker while using a small amount of data from the L2 speaker.

One possible future direction of this work is to improve its robustness in generating long utterances. Currently, our system uses a location-sensitive attention mechanism [35] in the seq2seq model, which can fail when the utterances are too long [90] (e.g., longer than 10 seconds). To solve this problem, an alternative attention mechanism could be used, such as Gaussian mixture attention mechanism [91], which has been shown to be more robust in generating long utterances [90]. An additional potential improvement would be to add an auxiliary decoder to perform phoneme recognition (during training) [8, 70]. Such auxiliary decoder would guide the hidden representation produced by the encoder to preserve phonetic information, enforcing the synthesized speech to be phonetically reasonable and improving synthesis quality [8, 70].

References

[1] D. Felps, H. Bortfeld, R. Gutierrez-Osuna, Foreign accent conversion in computer assisted pronunciation training, Speech communication 51 (10) (2009) 920–932.

[2] K. Probst, Y. Ke, M. Eskenazi, Enhancing foreign language tutors–in search of the golden speaker, Speech Communication 37 (3-4) (2002) 161–173.

[3] S. Ding, C. Liberatore, S. Sonsaat, I. Lučić, A. Silpachai, G. Zhao, E. Chukharev-Hudilainen, J. Levis, R. Gutierrez-Osuna, Golden speaker builder–an interactive tool for pronunciation training, Speech Communication 115 (2019) 51–66.

[4] R. Wang, J. Lu, Investigation of golden speakers for second language learners from imitation preference perspective by voice modification, Speech Communication 53 (2) (2011) 175–184.

[5] O. Turk, L. M. Arslan, Subband based voice conversion, in: International Conference on Spoken Language Processing, 2002.

[6] L. Sun, H. Wang, S. Kang, K. Li, H. M. Meng, Personalized, cross-lingual TTS using phonetic posteriorgrams., in: INTERSPEECH, 2016, pp. 322–326.

[7] Y. Oshima, S. Takamichi, T. Toda, G. Neubig, S. Sakti, S. Nakamura, Non-native speech synthesis preserving speaker individuality based on partial correction of prosodic and phonetic characteristics, in: INTERSPEECH, 2015.

[8] F. Biadsy, R. J. Weiss, P. J. Moreno, D. Kanvesky, Y. Jia, Parrotron: An end-to-end speech-to-speech conversion model and its applications to hearing-impaired speech and speech separation, in: INTERSPEECH, 2019, pp. 4115–4119.

[9] S. Aryal, D. Felps, R. Gutierrez-Osuna, Foreign accent conversion through voice morphing., in: INTERSPEECH, 2013, pp. 3077–3081.

[10] M. Huckvale, K. Yanagisawa, Spoken language conversion with accent morphing, in: ISCA Workshop on Speech Synthesis, 2007, pp. 64–70.

[11] S. Aryal, R. Gutierrez-Osuna, Can voice conversion be used to reduce non-native accents?, in: ICASSP, IEEE, 2014, pp. 7879–7883.

[12] G. Zhao, R. Gutierrez-Osuna, Using phonetic posteriorgram based frame pairing for segmental accent conversion, IEEE/ACM Transactions on Audio, Speech, and Language Processing 27 (10) (2019) 1649–1660.

[13] S. Aryal, R. Gutierrez-Osuna, Articulatory-based conversion of foreign accents with deep neural networks, in: INTERSPEECH, 2015.

[14] S. Aryal, R. Gutierrez-Osuna, Reduction of non-native accents through statistical parametric articulatory synthesis, The Journal of the Acoustical Society of America 137 (1) (2015) 433–446.

[15] G. Zhao, S. Ding, R. Gutierrez-Osuna, Foreign accent conversion by synthesizing speech from phonetic posteriorgrams., in: INTERSPEECH, 2019, pp. 2843–2847.

[16] 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: ICASSP, IEEE, 2020, pp. 6289–6293.

[17] C. H. Lampert, H. Nickisch, S. Harmeling, Learning to detect unseen object classes by between-class attribute transfer, in: IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2009, pp. 951–958.

[18] S. Liu, J. Zhong, L. Sun, X. Wu, X. Liu, H. Meng, Voice conversion across arbitrary speakers based on a single target-speaker utterance., in: INTERSPEECH, 2018, pp. 496–500.

[19] C.-C. Hsu, H.-T. Hwang, Y.-C. Wu, Y. Tsao, H.-M. Wang, Voice conversion from non-parallel corpora using variational auto-encoder, in: Asia-Pacific Signal and Information Processing Association Annual Summit and Conference, IEEE, 2016, pp. 1–6.

[20] H. Kameoka, T. Kaneko, K. Tanaka, N. Hojo, ACVAE-VC: Non-parallel many-to-many voice conversion with auxiliary classifier variational autoencoder, arXiv preprint arXiv:1808.05092.

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

[22] 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.

[23] S. Ö. Arık, J. Chen, K. Peng, W. Ping, Y. Zhou, Neural voice cloning with a few samples, in: International Conference on Neural Information Processing Systems, 2018, pp. 10040–10050.

[24] A. Das, G. Zhao, J. Levis, E. Chukharev-Hudilainen, R. Gutierrez-Osuna, Understanding the effect of voice quality and accent on talker similarity, in: INTERSPEECH, 2020, pp. 1763–1767.

[25] M. Leikin, R. Ibrahim, Z. Eviatar, S. Sapir, Listening with an accent: Speech perception in a second language by late bilinguals, Journal of psycholinguistic research 38 (5) (2009) 447.

[26] R. C. Major, S. F. Fitzmaurice, F. Bunta, C. Balasubramanian, The effects of nonnative accents on listening comprehension: Implications for esl assessment, TESOL quarterly 36 (2) (2002) 173–190.

[27] M. Van Heugten, C. Bergmann, A. Cristia, The effects of talker voice and accent on young children's speech perception, in: Individual differences in speech production and perception, Peter Lang, 2015, pp. 57–88.

[28] A. Cristia, A. Seidl, C. Vaughn, R. Schmale, A. Bradlow, C. Floccia, Linguistic processing of accented speech across the lifespan, Frontiers in psychology 3 (2012) 479.

[29] S. Ding, R. Gutierrez-Osuna, Group latent embedding for vector quantized variational autoencoder in non-parallel voice conversion., in: INTERSPEECH, 2019, pp. 724–728.

[30] Y. Saito, Y. Ijima, K. Nishida, S. Takamichi, Non-parallel voice conversion using variational autoencoders conditioned by phonetic posteriorgrams and d-vectors, in: ICASSP, IEEE, 2018, pp. 5274–5278.

[31] W.-N. Hsu, Y. Zhang, J. Glass, Unsupervised learning of disentangled and interpretable representations from sequential data, in: Advances in Neural Information Processing Systems, 2017, pp. 1878–1889.

[32] S. H. Mohammadi, T. Kim, One-shot voice conversion with disentangled representations by leveraging phonetic posteriorgrams., in: INTERSPEECH, 2019, pp. 704–708.

[33] H. Lu, Z. Wu, D. Dai, R. Li, S. Kang, J. Jia, H. Meng, One-shot voice conversion with global speaker embeddings., in: INTERSPEECH, 2019, pp. 669–673.

[34] S. Ding, G. Zhao, R. Gutierrez-Osuna, Improving the speaker identity of non-parallel many-to-many voice conversion with adversarial speaker recognition, in: INTERSPEECH, 2020, pp. 776–780.

[35] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, Y. Bengio, Attention-based models for speech recognition, in: Advances in Neural Information Processing Systems, 2015, pp. 577–585.

[36] D. Griffin, J. Lim, Signal estimation from modified short-time fourier transform, IEEE Transactions on Acoustics, Speech, and Signal Processing 32 (2) (1984) 236–243.

[37] A. Tamamori, T. Hayashi, K. Kobayashi, K. Takeda, T. Toda, Speaker-dependent wavenet vocoder., in: INTERSPEECH, Vol. 2017, 2017, pp. 1118–1122.

[38] N. Kalchbrenner, E. Elsen, K. Simonyan, S. Noury, N. Casagrande, E. Lockhart, F. Stimberg, A. van den Oord, S. Dieleman, K. Kavukcuoglu, Efficient neural audio synthesis, in: International Conference on Machine Learning, 2018.

[39] G. Zhao, S. Sonsaat, A. Silpachai, I. Lucic, E. Chukharev-Hudilainen, J. Levis, R. Gutierrez-Osuna, L2-ARCTIC: A non-native English speech corpus, in: INTERSPEECH, 2018, pp. 2783–2787.

[40] D. Felps, C. Geng, R. Gutierrez-Osuna, Foreign accent conversion through concatenative synthesis in the articulatory domain, IEEE Transactions on Audio, Speech, and Language Processing 20 (8) (2012) 2301–2312.

[41] S. Aryal, R. Gutierrez-Osuna, Data driven articulatory synthesis with deep neural networks, Computer Speech & Language 36 (2016) 260–273.

[42] M. Brand, Voice puppetry, in: Annual conference on Computer graphics and interactive techniques, 1999, pp. 21–28.

[43] B. Denby, M. Stone, Speech synthesis from real time ultrasound images of the tongue, in: ICASSP, Vol. 1, IEEE, 2004, pp. I–685.

[44] R. Mumtaz, S. Preuß, C. Neuschaefer-Rube, C. Hey, R. Sader, P. Birkholz, Tongue contour reconstruction from optical and electrical palatography, IEEE Signal Processing Letters 21 (6) (2014) 658–662.

[45] A. Toutios, T. Sorensen, K. Somandepalli, R. Alexander, S. S. Narayanan, Articulatory synthesis based on real-time magnetic resonance imaging data., in: INTERSPEECH, 2016, pp. 1492–1496.

[46] S. H. Mohammadi, A. Kain, An overview of voice conversion systems, Speech Communication 88 (2017) 65–82.

[47] B. Sisman, J. Yamagishi, S. King, H. Li, An overview of voice conversion and its challenges: From statistical modeling to deep learning, arXiv preprint arXiv:2008.03648.

[48] T. Toda, A. W. Black, K. Tokuda, Voice conversion based on maximum-likelihood estimation of spectral parameter trajectory, IEEE Transactions on Audio, Speech, and Language Processing 15 (8) (2007) 2222–2235.

[49] D. Erro, A. Moreno, A. Bonafonte, INCA algorithm for training voice conversion systems from nonparallel corpora, IEEE Transactions on Audio, Speech, and Language Processing 18 (5) (2009) 944–953.

[50] R. Takashima, T. Takiguchi, Y. Ariki, Exemplar-based voice conversion in noisy environment, in: 2012 IEEE Spoken Language Technology Workshop (SLT), IEEE, 2012, pp. 313–317.

[51] S. Ding, G. Zhao, C. Liberatore, R. Gutierrez-Osuna, Learning structured sparse representations for voice conversion, IEEE/ACM Transactions on Audio, Speech, and Language Processing 28 (2019) 343–354.

[52] L.-H. Chen, Z.-H. Ling, L.-J. Liu, L.-R. Dai, Voice conversion using deep neural networks with layer-wise generative training, IEEE/ACM Transactions on Audio, Speech, and Language Processing 22 (12) (2014) 1859–1872.

[53] S. Desai, A. W. Black, B. Yegnanarayana, K. Prahallad, Spectral mapping using artificial neural networks for voice conversion, IEEE Transactions on Audio, Speech, and Language Processing 18 (5) (2010) 954–964.

[54] T. Nakashika, T. Takiguchi, Y. Minami, Non-parallel training in voice conversion using an adaptive restricted boltzmann machine, IEEE/ACM Transactions on Audio, Speech, and Language Processing 24 (11) (2016) 2032–2045.

[55] F.-L. Xie, F. K. Soong, H. Li, A KL divergence and DNN-based approach to voice conversion without parallel training sentences., in: INTERSPEECH, 2016, pp. 287–291.

[56] H. Miyoshi, Y. Saito, S. Takamichi, H. Saruwatari, Voice conversion using sequence-to-sequence learning of context posterior probabilities, in: INTERSPEECH, 2017, pp. 1268–1272.

[57] T. Kaneko, H. Kameoka, K. Tanaka, N. Hojo, CycleGAN-VC2: Improved CycleGAN-based non-parallel voice conversion, in: ICASSP, IEEE, 2019, pp. 6820–6824.

[58] Y. Zhou, X. Tian, H. Xu, R. K. Das, H. Li, Cross-lingual voice conversion with bilingual phonetic posteriorgram and average modeling, in: ICASSP, IEEE, 2019, pp. 6790–6794.

[59] W.-C. Huang, Y.-C. Wu, H.-T. Hwang, P. L. Tobing, T. Hayashi, K. Kobayashi, T. Toda, Y. Tsao, H.-M. Wang, Refined WaveNet vocoder for variational autoencoder based voice conversion, in: European Signal Processing Conference, IEEE, 2019, pp. 1–5.

[60] T. Kaneko, H. Kameoka, K. Hiramatsu, K. Kashino, Sequence-to-sequence voice conversion with similarity metric learned using generative adversarial networks., in: INTERSPEECH, Vol. 2017, 2017, pp. 1283–1287.

[61] J.-X. Zhang, Z.-H. Ling, 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 28 (2019) 540–552.

[62] C.-C. Hsu, H.-T. Hwang, Y.-C. Wu, Y. Tsao, H.-M. Wang, Voice conversion from unaligned corpora using variational autoencoding wasserstein generative adversarial networks, in: INTERSPEECH, Vol. 2017, 2017.

[63] A. Van Den Oord, O. Vinyals, et al., Neural discrete representation learning, in: Advances in Neural Information Processing Systems, 2017, pp. 6306–6315.

[64] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, P. Ouellet, Front-end factor analysis for speaker verification, IEEE Transactions on Audio, Speech, and Language Processing 19 (4) (2010) 788–798.

[65] E. Variani, X. Lei, E. McDermott, I. L. Moreno, J. Gonzalez-Dominguez, Deep neural networks for small footprint text-dependent speaker verification, in: ICASSP, IEEE, 2014, pp. 4052–4056.

[66] I. Sutskever, O. Vinyals, Q. V. Le, Sequence to sequence learning with neural networks, in: Advances in Neural Information Processing Systems, 2014, pp. 3104–3112.

[67] D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, in: International Conference on Learning Representations, 2015.

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

[69] 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: ICASSP, IEEE, 2018, pp. 4779–4783.

[70] Y. Jia, R. J. Weiss, F. Biadsy, W. Macherey, M. Johnson, Z. Chen, Y. Wu, Direct speech-to-speech translation with a sequence-to-sequence model, in: INTERSPEECH, 2019.

[71] D. Povey, G. Cheng, Y. Wang, K. Li, H. Xu, M. Yarmohammadi, S. Khudanpur, Semi-orthogonal low-rank matrix factorization for deep neural networks., in: INTERSPEECH, 2018, pp. 3743–3747.

[72] V. Peddinti, D. Povey, S. Khudanpur, A time delay neural network architecture for efficient modeling of long temporal contexts, in: INTERSPEECH, 2015.

[73] S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, in: International Conference on Machine Learning, 2015, pp. 448–456.

[74] V. Panayotov, G. Chen, D. Povey, S. Khudanpur, Librispeech: An ASR corpus based on public domain audio books, in: ICASSP, IEEE, 2015, pp. 5206–5210.

[75] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

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

[77] W. Chan, N. Jaitly, Q. Le, O. Vinyals, Listen, attend and spell: A neural network for large vocabulary conversational speech recognition, in: ICASSP, IEEE, 2016, pp. 4960–4964.

[78] R. J. Williams, D. Zipser, A learning algorithm for continually running fully recurrent neural networks, Neural Computation 1 (2) (1989) 270–280.

[79] A. Nagrani, J. S. Chung, A. Zisserman, VoxCeleb: A large-scale speaker identification dataset, in: INTERSPEECH, 2017, pp. 2616–2620.

[80] I. Loshchilov, F. Hutter, SGDR: Stochastic gradient descent with warm restarts, in: International Conference on Learning Representations, 2017.

[81] S. Weinberger, Speech accent archive, George Mason University.

[82] J. Kominek, A. W. Black, The CMU ARCTIC speech databases, in: Fifth ISCA workshop on speech synthesis, 2004.

[83] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, et al., Tensorflow: Large-scale machine learning on heterogeneous distributed systems, arXiv preprint arXiv:1603.04467.

[84] L. Van der Maaten, G. Hinton, Visualizing data using t-sne., Journal of machine learning research 9 (11).

[85] G. E. Henter, J. Lorenzo-Trueba, X. Wang, M. Kondo, J. Yamagishi, Cyborg speech: Deep multilingual speech synthesis for generating segmental foreign accent with natural prosody, in: ICASSP, IEEE, 2018, pp. 4799–4803.

[86] C. Veaux, J. Yamagishi, K. MacDonald, et al., Superseded-cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit.

[87] M. J. Munro, T. M. Derwing, Foreign accent, comprehensibility, and intelligibility in the speech of second language learners, Language Learning 49 (s1) (2002) 285–310.

[88] D. Felps, R. Gutierrez-Osuna, Developing objective measures of foreign-accent conversion, IEEE Transactions on Audio, Speech, and Language Processing 18 (5) (2010) 1030–1040.

[89] S. Buchholz, J. Latorre, Crowdsourcing preference tests, and how to detect cheating., in: INTERSPEECH, 2011, pp. 3053–3056.

[90] R. Skerry-Ryan, E. Battenberg, Y. Xiao, Y. Wang, D. Stanton, J. Shor, R. J. Weiss, R. Clark, R. A. Saurous, Towards end-to-end prosody transfer for expressive speech synthesis with Tacotron, arXiv preprint arXiv:1803.09047.

[91] A. Graves, Generating sequences with recurrent neural networks, arXiv preprint arXiv:1308.0850.

Declaration of interests

☒ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: