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

Parrotron: An End-to-End Speech-to-Speech Conversion Model and its Applications to Hearing-Impaired Speech and Speech Separation

Fadi Biadsy, Ron J. Weiss, Pedro J. Moreno, Dimitri Kanvesky, Ye Jia

Google

{biadsy,ronw,pedro,dkanevsky,jiaye}@google.com

Abstract

We describe Parrotron, an end-to-end-trained speech-to-speech conversion model that maps an input spectrogram directly to another spectrogram, without utilizing any intermediate discrete representation. The network is composed of an encoder, spectrogram and phoneme decoders, followed by a vocoder to synthesize a time-domain waveform. We demonstrate that this model can be trained to normalize speech from any speaker regardless of accent, prosody, and background noise, into the voice of a *single* canonical target speaker with a fixed accent and consistent articulation and prosody. We further show that this normalization model can be adapted to normalize highly atypical speech from a deaf speaker, resulting in significant improvements in intelligibility and naturalness, measured via a speech recognizer and listening tests. Finally, demonstrating the utility of this model on other speech tasks, we show that the same model architecture can be trained to perform a speech separation task.

Index Terms: speech normalization, voice conversion, atypical speech, speech synthesis, sequence-to-sequence model

1. Introduction

Encoder-decoder models with attention have recently shown considerable success in modeling a variety of complex sequence-to-sequence problems. These models have been successfully adopted to tackle a diverse set of tasks in speech and natural language processing, such as machine translation [1], speech recognition [2], and even combined speech translation [3]. They have also achieved state-of-the-art results in end-to-end Text-To-Speech (TTS) synthesis [4] and Automatic Speech Recognition (ASR) [5], using a single neural network that directly generates the target sequences, given virtually raw inputs.

In this paper, we combine attention-based speech recognition and synthesis models to build a direct end-to-end speech-to-speech sequence transducer. This model generates a speech spectrogram as a function of a different input spectrogram, with no intermediate discrete representation.

We test whether such a unified model is powerful enough to normalize arbitrary speech from multiple accents, imperfections, potentially including background noise, and generate the same content in the voice of a *single* predefined target speaker. The task is to project away all non-linguistic information, including speaker characteristics, and to retain only what is being said, *not* who, where or how it is said. This amounts to a text-independent, many-to-one voice conversion task [6]. We evaluate the model on this voice normalization task using ASR and listening studies, verifying that it is able to preserve the underlying speech content and project away other information, as intended.

We demonstrated that the pretrained normalization model can be adapted to perform a more challenging task of converting highly atypical speech from a deaf speaker into fluent speech, significantly improving intelligibility and naturalness. Finally, we evaluate whether the same network is capable of performing a speech separation task. Readers are encouraged to listen to sound examples on the companion website. [1]

A variety of techniques have been proposed for voice conversion, including mapping code books [7], neural networks [8, 9], dynamic frequency warping [10], and Gaussian mixture models [11–13]. Recent work has also addressed accent conversion [14, 15] . In this paper we propose an end-to-end architecture that directly generates the target signal, synthesizing it from scratch. It is most similar to recent work on sequence-to-sequence voice conversion [16–18]. [16] uses a similar end-to-end model, conditioned on speaker identities, to transform word segments from multiple speakers into multiple target voices. Unlike [17], which trained separate models for each source-target speaker pair, we focus on many-to-one conversion. Our model is trained on source-target spectrogram pairs, without augmenting inputs with bottleneck features from a pretrained speech recognizer to more explicitly capture phonemic information in the source speech [17]. However, we do find it helpful to multitask train the model to predict source speech phonemes. Finally, in contrast to [18], we train the model without auxiliary alignment or auto-encoding losses.

Similar voice conversion techniques have also been applied to improving intelligibility for speakers with vocal disabilities [19, 20], and hearing-impaired speakers in particular [21]. We apply more modern machine learning techniques to this problem, and demonstrate that, given sufficient training data, an end-to-end trained one-to-one conversion model can dramatically improve intelligibility and naturalness of a deaf speaker.

2. Model Architecture

We use an end-to-end sequence-to-sequence model architecture that takes an input source speech and generates/synthesizes target speech as output. The only training requirement of such a model is a parallel corpus of paired input-output speech utterances. We refer to this speech-to-speech model as *Parrotron*.

As shown in Figure 1, the network is composed of an encoder and a decoder with attention, followed by a vocoder to synthesize a time-domain waveform. The encoder converts a sequence of acoustic frames into a hidden feature representation which the decoder consumes to predict a spectrogram. The core architecture is based on recent attention-based end-to-end ASR models [2, 22] and TTS models such as Tacotron [4, 23].

2.1. Spectrogram encoder

The base encoder configuration is similar to the encoder in [24], and some variations are evaluated in Section 3.1. From the input speech signal, sampled at 16 kHz, we extract 80-dimensional log-mel spectrogram features over a range of 125-7600 Hz, cal-

[1] `https://google.github.io/tacotron/publications/parrotron`

http://dx.doi.org/10.21437/Interspeech.2019-1789

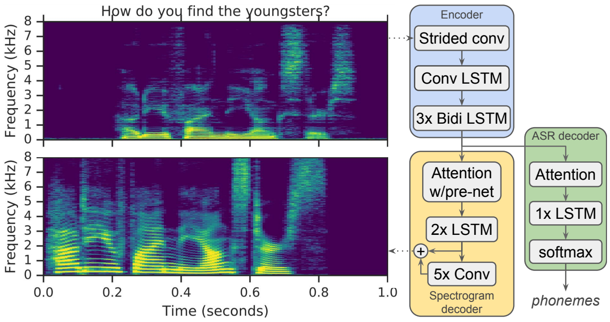

Figure 1: *Overview of the Parrotron network architecture. The output speech is from a different gender (having higher pitch and formants), and has a slightly slower speaking rate.*

culated using a Hann window, 50 ms frame length, 12.5 ms frame shift, and 1024-point Short-Time Fourier Transform (STFT).

The input features are passed into a stack of two convolutional layers with ReLU activations, each consisting of 32 kernels, shaped 3×3 in time $\times$ frequency, and strided by 2×2, downsampling the sequence in time by a total factor of 4, decreasing the computation in the following layers. Batch normalization [25] is applied after each layer.

This downsampled sequence is passed into a bidirectional convolutional LSTM (CLSTM) [26,27] layer using a 1×3 filter, i.e. convolving only across the frequency axis within each time step. Finally, this is passed into a stack of three bidirectional LSTM layers of size 256 in each direction, interleaved with a 512-dim linear projection, followed by batchnorm and ReLU activation, to compute the final 512-dim encoder representation.

2.2. Spectrogram decoder

The decoder targets are 1025-dim STFT magnitudes, computed with the same framing as the input features, and a 2048-point FFT. We use the decoder network described in [4], consisting of an autoregressive RNN to predict the output spectrogram from the encoded input sequence one frame at a time. The prediction from the previous decoder time step is first passed through a small pre-net containing 2 fully connected layers of 256 ReLU units, which was found to help to learn attention [4, 23]. The pre-net output and attention context vector are concatenated and passed through a stack of 2 unidirectional LSTM layers with 1024 units. The concatenation of the LSTM output and the attention context vector is then projected through a linear transform to produce a prediction of the target spectrogram frame. Finally, these predictions are passed through 5-layer convolutional post-net which predicts a residual to add to the initial prediction. Each post-net layer has 512 filters shaped 5×1 followed by batch normalization and tanh activation.

To synthesize an audio signal from the predicted spectrogram, we primarily use the Griffin-Lim algorithm [28] to estimate a phase consistent with the predicted magnitude, followed by an inverse STFT. However, when conducting human listening tests we instead use a WaveRNN [29] neural vocoder which has been shown to significantly improve synthesis fidelity [4, 30].

2.3. Multitask training with an ASR decoder

Since the goal of this work is to generate only speech sounds and not arbitrary audio, jointly training the encoder network to simultaneously learn a high level representation of the underlying language serves to bias the spectrogram decoder predictions toward a representation of the same underlying speech content. We accomplish this by adding an auxiliary ASR decoder to predict the (grapheme or phoneme) transcript of the output speech, conditioned on the encoder latent representation. Such a multitask trained encoder can be thought of as learning a latent representation of the input that maintains information about the underlying transcript, i.e. one that is closer to the latent representation learned within a TTS sequence-to-sequence network.

The decoder input is created by concatenating a 64-dim embedding for the grapheme emitted at the previous step, and the 512-dim attention context. This is passed into a 256 unit LSTM layer. Finally the concatenation of the attention context and LSTM output is passed into a softmax to predict the probability of emitting each grapheme in the output vocabulary.

Table 1: *WER comparison of different architecture variations combined with different auxiliary ASR losses.*

ASR decoder target	#CLSTM	#LSTM	Attention	WER
None	1	3	Additive	27.1
Grapheme	1	3	Additive	19.9
Grapheme	1	3	Location	19.2
Phoneme	1	3	Location	18.5
Phoneme	0	3	Location	20.9
Phoneme	0	5	Location	18.3
Phoneme w/slow decay	0	5	Location	17.6

3. Applications

3.1. Voice normalization

We address the task of normalizing speech from an arbitrary speaker to the voice of a predefined canonical speaker. As discussed in Section 2, to make use of Parrotron, we require a parallel corpus of utterances spanning a variety of speakers and recording conditions, each mapped to speech from a canonical speaker. Since it is impractical to have single speaker record many hours of speech in clean acoustic environment, we use Google's Parallel WaveNet-based TTS [31] system to generate training targets from a large hand-transcribed speech corpus. Essentially this reduces the task to reproducing any input speech in the voice of a single-speaker TTS system. Using TTS to generate this parallel corpus ensures that: (1) the target is always spoken with a consistent predefined speaker and accent; (2) without any background noise or disfluencies. (3) Finally, we can synthesize as much data as necessary to scale to very large corpora.

3.1.1. Experiments

We train the model on a $\sim$30,000 hour training set consisting of about 24 million English utterances which are anonymized and manually transcribed, and are representative of Google's US English voice search traffic. Using this corpus, we run a TTS system to generate target utterances in a synthetic female voice.

To evaluate whether Parrotron preserves the linguistic content of the original input signal after normalization, we report word error rates (WERs) using a state-of-the-art ASR engine on the Parrotron output as a measure of speech intelligibility. Note that the ASR engine is not trained on Griffin-Lim synthesized speech, a domain mismatch leading to higher WER. Table 1 compares different architecture and loss configurations, evaluated on a hand-transcribed held-out test set of 10K anonymized utterances sampled from the same distribution as the train set.

Table 2: *Performance of Parrotron models on real speech.*

Model	MOS	WER
Real speech	4.04 ± 0.19	34.2
Parrotron (female)	3.81 ± 0.16	39.8
Parrotron (male)	3.77 ± 0.16	37.5

Table 3: *Subjective evaluation of Parrotron output quality.*

Survey question	Avg. score / agreement
How similar is the Parrotron voice to the TTS voice on the 5 point Likert scale?	4.6
Does the output speech	
use a standard American English accent?	94.4%
contain *any* background noise?	0.0%
contain *any* disfluencies?	0.0%
use consistent articulation, standard intonation and prosody?	83.3%

The WER on the original speech (matched condition) is 8.3%, which can be viewed as an upper bound. Synthesizing the *reference* transcripts with a high quality TTS model and transcribing them using our ASR engine obtains a WER of 7.4%.

The top row of Table 1 shows performance using the base model architecture described in Section 2, using a spectrogram decoder employing additive attention [1] without an auxiliary ASR loss. Adding a parallel decoder to predict graphemes leads to a significant improvement, reducing the WER from 27.1% to 19.9%. Extending the additive attention with a location sensitive term [32] further improves results. This improves outputs on long utterances where additive attention sometimes failed.

Since orthography in English does not uniquely predict pronunciation, we hypothesize that using phoneme targets for the ASR decoder (obtained from forced alignment to the reference transcript) may reduce noise propagated back to the encoder. Indeed we find that this also shows consistent improvements.

Turning our attention to the encoder architecture, we found that reducing the number of parameters by removing the CLSTM significantly hurts performance. However, using 2 extra BLSTM layers instead of the CLSTM slightly improves results, while simultaneously simplifying the model. Hyperparameter tuning revealed that simply using slower learning rate decay (ending at 90k instead of 60K steps) on our best model yields 17.6% WER. See Figure 1 for an example model output.

Using the best-performing Parrotron model, we conducted listening studies on a more challenging test set, which contains heavily accented speech plus background noise. As shown in Table 2, we verify that under these conditions Parrotron still preserves the linguistic content, since its WER is comparable to that of real speech. The naturalness MOS score decreases slightly with Parrotron when compared to that of real speech. Recall that the objective in this work is to perform many-to-one speech normalization, not to improve ASR. Training an ASR engine on the output of Parrotron is likely to improve WER results. However, we leave evaluation of the impact of such normalization on ASR to future work.

Finally, we conduct another listening test to evaluate whether the model consistently generates normalized speech with the same TTS voice. We present a random sample of 20 utterances produced by Parrotron to 8 native English subjects and ask questions shown in Table 3 for each utterance. The results in the table verify that the model consistently normalizes speech.

3.1.2. Error analysis

We analyze the types of phoneme errors Parrotron makes after normalization. We first obtain the true phonemes by force aligning each manual transcript with the corresponding real speech signal. Using this alignment, we compute two confusion matrices on the test set: (A) one computed by aligning the true phonemes with the hypothesized phonemes from the original speech, i.e. the Parrotron input; (B) another computed by aligning the true phonemes to the hypothesized phonemes from the normalized speech. We subtract A from B and rank the phoneme confusions to identify confusions which occur more frequently in Parrotron output than in real speech. Since we have 40 phonemes (+ epsilon), we have 1681 phoneme confusion pairs. In the top 5% of confusions, we observe that 26% of them are plosives (/k/, /t/, /d/, /g/, and /b/) which are mostly dropped. The average rank of plosive confusions is 244/1681, suggesting that the model does not accurately model these short phonemes. We also observe another 12% correspond to vowel exchanges. This is not surprising since the model attempts to normalize multiple accents to that of the target TTS speaker.

Errors in plosive and other short phonemes are not surprising since the model uses an L2 reconstruction loss. Under this loss, a frame containing a vowel contributes the same amount as a frame containing /t/. Since there are significantly more vowel frames than plosives in the training data, this biases training to focus more on accurately generating phonemes of longer duration.

We observe that feeding Arabic and Spanish utterances into the US-English Parrotron model often results in output which echoes the original speech content with an American accent, in the target voice. Such behavior is qualitatively different from what one would obtain by simply running an ASR followed by a TTS for example. A careful listening study is needed to further validate these results.

3.2. Normalization of hearing-impaired speech

Addressing a more challenging accessibility application, we investigate whether the normalization model can be used to to convert atypical speech from a deaf speaker into fluent speech. This could be used to improve the vocal communication of people with such conditions or other speech disorders, or as a front-end to voice-enabled systems.

We focus on one case study of a profoundly deaf subject who was born in Russia to normal-hearing parents, and learned English as a teenager. The subject used Russian phonetic representation of English words and learned to speak them using Russian letters (e.g., cat $\rightarrow$ k a T). Using a live (human in the loop) transcription service and ASR systems for multiple years helped improve their articulation. See [33] for more details.

We experiment with adapting the best model from Section 3.1 using a dataset of 15.4 hours of speech, corresponding to read movie quotes. We use 90% of the data for adaptation (KADPT), and hold out the remainder: 5% (about 45 minutes) for dev and 5% for test (KTEST). This data was challenging; we learned that some prompts were difficult to pronounce by unimpaired but non-native English speakers. The WER using Google's ASR system on the TTS-synthesized *reference* transcripts is 14.8%. See the companion website for examples.

3.2.1. Experiments

Our first experiment is to test the performance of Google's state-of-the-art ASR system on KTEST. As shown in Table 4, we find that the ASR system performs very poorly on this speech,

Table 4: *Performance on speech from a deaf speaker.*

Model	MOS	WER
Real speech	2.08 ± 0.22	89.2
Parrotron (male)	2.58 ± 0.20	109.3
Parrotron (male) finetuned	$\mathbf{3.52 \pm 0.14}$	**32.7**

Table 5: *Parrotron speech separation performance.*

Data	WER	del	ins	sub
Original (Clean)	8.8	1.6	1.5	5.8
Noisy	33.2	3.6	19.1	10.5
Denoised using Parrotron	**17.3**	6.7	2.2	8.4

obtaining 89.2% WER on the test set. The MOS score on KTEST is 2.08, rated by subjects unfamiliar with the subject's speech.

We then test whether our best out-of-the-box Parrotron trained for the normalization task, shown in Section 3.1, can successfully normalize this type of speech. The only difference here is that Parrotron is trained on a male TTS speech, obtained form our production WaveNet-based TTS. Testing on KTEST, we find that the output of this model was rated as natural as the original speech, but our ASR engine performs even more poorly on the converted speech than the original speech. In other words, Parrotron normalization system trained on standard speech fails completely to normalize this type of speech. We have also manually inspected the output of this Parrotron and found that the model produces speech-like sounds but nonsense words.

Now, we test whether utilizing KADPT would have any impact on Parrotron performance. We first take the fully converged male Parrotron normalization model and conduct multiple finetuning experiments using KADPT. With a constant learning rate of 0.1, we (1) adapt all parameters on the fully converged model; (2) adapt all parameters except freezing the spectrogram decoder parameters; (3) freeze both spectrogram decoder and phoneme decoder parameters while finetuning only the encoder.

We find that all finetuning strategies lead to intelligible and significantly more natural speech. The best finetuning strategy was adapting *all* parameters, which increased the MOS naturalness score by over 1.4 points compared to the original speech, and dramatically reduced the WER from 89.2% to 32.7%. Finetuning strategy (2) obtains 34.1% WER and adapting only encoder parameters (strategy (3)), obtains 38.6% WER.

Note that one advantage of directly converting speech to speech over cascading a finetuned ASR engine with TTS is as follows. Synthesizing the output of an ASR engine may generate speech far from intended, due to unavoidable ASR errors. A speech-to-speech model, however, is likely to produce sounds closer to the original speech. We have seen significant evidence to support this hypothesis, but leave it to future work to quantify.

3.3. Speech separation

Finally, to illustrate that the Parrotron architecture can be used in a variety of speech applications, we evaluate it on a speech separation task of *reconstructing* the signal from the loudest speaker within a mixture of overlapping speech. We focus on instantaneous mixtures of up to 8 different speakers.

It is important to stress that our intent in this section is not to propose a state of the art separation system, but rather to demonstrate that the proposed architecture may apply to different speech applications. More importantly, in contrast to previous applications which made use of synthetic training targets, we evaluate whether Parrotron is able to generate speech from an open set of speakers, generalizing beyond the training set. Furthermore, unlike state-of-the-art speech separation techniques [34, 35], Parrotron generates the signal from scratch as opposed to using a masking-based filtering approach and is able to rely on an implicit phoneme language model.

We use the same voice-search data described in Section 3.1 to artificially construct instantaneous mixtures of speech signals. For each target utterance in the training data, we randomly select a set of 1 to 7 utterances to mix together as the background noise. The number of background utterances is also randomly selected. Before mixing, we normalize all utterances to have similar gains.

We mix target utterances with the background noise by simply averaging the two signals with a randomly sampled weight $w \in [0.1, 0.5)$ for the background and $1 - w$ for the target utterance. This results in an average SNR across all artificially constructed utterances of 12.15 dB, with a standard deviation of 4.7. 188K utterances from this corpus are held out for testing. While we do not explicitly incorporate reverberation or non-speech noise, the underlying utterances come from a variety of recording environments with their own background noise.

To evaluate whether Parrotron can perform this separation task, we train a model to the best performing architecture as in Section 3.1. We feed as inputs our mixed utterances and train the model to generate corresponding original *clean* utterances.

We evaluate the impact of this separation model using Google's ASR system. We compare WERs on three sets of 188k held-out utterances: (1) the original *clean* speech before adding background speech; (2) the *noisy set* after mixing background speech; (3) the cleaned output generated by running Parrotron on the noisy set. As shown in Table 5, we observe significant WER reduction after running Parrotron on the noisy set, demonstrating that the model can preserve speech from the target speaker and separate them from other speakers. Parrotron significantly reduces insertions, which correspond to words spoken by background speakers, but suffers from increased deletions, which is likely due to early end-of-utterance prediction.

4. Conclusion

We described Parrotron, an end-to-end speech-to-speech model that converts an input spectrogram directly to another spectrogram, without intermediate symbolic representation. We find that the model can be trained to normalize speech from different speakers into speech of a single target speaker's voice while preserving the linguistic content and projecting away non-linguistic content. We then showed that this model can successfully be adapted to improve WER and naturalness of speech from a deaf speaker. We finally demonstrate that the same model can be trained to successfully identify, separate and reconstruct the loudest speaker in a mixture of overlapping speech, improving ASR performance. The Parrotron system has other potential applications, e.g. improving intelligibility by converting heavily accented or otherwise atypical speech into standard speech. In the future, we plan to test it on other speech disorders, and adopt techniques from [16, 30] to preserve the speaker identity.

5. Acknowledgments

We thank Françoise Beaufays, Michael Brenner, Diamantino Caseiro, Zhifeng Chen, Mohamed Elfeky, Patrick Nguyen, Bhuvana Ramabhadran, Andrew Rosenberg, Jason Pelecanos, Johan Schalkwyk, Yonghui Wu, and Zelin Wu for useful feedback.

6. References

[1] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” *Proc. International Conference on Learning Representations*, 2015.

[2] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, “Listen, attend and spell: A neural network for large vocabulary conversational speech recognition,” in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2016, pp. 4960–4964.

[3] B. Alexandre, P. Olivier, S. Christophe, and B. Laurent, “Listen and translate: A proof of concept for end-to-end speech-to-text translation,” in *NIPS Workshop on End-to-end Learning for Speech and Audio Processing*, 2016.

[4] J. Shen, R. Pang, R. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. J. Skerry-Ryan, R. A. Saurous, Y. Agiomyrgiannakis, and Y. Wu, “Natural TTS synthesis by conditioning WaveNet on mel spectrogram predictions,” in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2018.

[5] C.-C.Chiu, T. N. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen, A. Kannan, R. J. Weiss, K. Rao, K. Gonina, N. Jaitly, B. Li, J. Chorowski, and M. Bacchiani, “State-of-the-art speech recognition with sequence-to-sequence models,” in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2018.

[6] T. Toda, Y. Ohtani, and K. Shikano, “One-to-many and many-to-one voice conversion based on eigenvoices,” in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2007.

[7] M. Abe, S. Nakamura, K. Shikano, , and H. Kuwabara, “Voice conversion through vector quantization,” *Journal of the Acoustical Society of Japan (E)*, vol. 11, no. 2, pp. 71–76, 1990.

[8] T. Watanabe, T. Murakami, M. Namba, and T. H. Y. Ishida, “Transformation of spectral envelope for voice conversion based on radial basis function networks,” in *International Conference on Spoken Language Processing*, 2002.

[9] M. Narendranath, H. A. Murthy, S. Rajendran, and B. Yegnanarayana, “Transformation of formants for voice conversion using artificial neural networks,” *Speech Communication*, 1995.

[10] H. Valbret, E. Moulines, and J. P. Tubach, “Voice transformation using PSOLA technique,” in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1992.

[11] A. Kain and M. W. Macon, “Design and evaluation of a voice conversion algorithm based on spectral envelop mapping and residual prediction,” in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2001.

[12] A. R. Toth and A. W. Black, “Using articulatory position data in voice transformation,” in *Workshop on Speech Synthesis*, 2007.

[13] T. Toda, A. W. Black, and K. Tokuda, “Acoustic-to-articulatory inversion mapping with gaussian mixture model,” in *ICSLP*, 2004.

[14] A. Bearman, K. Josund, and G. Fiore, “Accent conversion using artificial neural networks,” Stanford University, Tech. Rep., 2017.

[15] D. Felps, H. Bortfeld, and R. Gutierrez-Osuna, “Foreign accent conversion in computer assisted pronunciation training,” in *Speech Communication*, 2009.

[16] A. Haque, M. Guo, and P. Verma, “Conditional end-to-end audio transforms,” in *Proc. Interspeech*, 2018.

[17] J. Zhang, Z. Ling, L.-J. Liu, Y. Jiang, and L.-R. Dai, “Sequence-to-sequence acoustic modeling for voice conversion,” *IEEE Transactions on Audio, Speech, and Language Processing*, 2019.

[18] K. Tanaka, H. Kameoka, T. Kaneko, and N. Hojo, “AttS2S-VC: Sequence-to-sequence voice conversion with attention and context preservation mechanisms,” *arXiv:1811.04076*, 2018.

[19] J. Yamagishi, C. Veaux, S. King, and S. Renals, “Speech synthesis technologies for individuals with vocal disabilities: Voice banking and reconstruction,” *Acoustical Science and Technology*, 2012.

[20] A. B. Kain, J.-P. Hosom, X. Niu, J. P. van Santen, M. Fried-Oken, and J. Staehely, “Improving the intelligibility of dysarthric speech,” *Speech Communication*, vol. 49, no. 9, pp. 743–759, 2007.

[21] C.-L. Lee, W.-W. Chang, and Y.-C. Chiang, “Spectral and prosodic transformations of hearing-impaired mandarin speech,” *Speech Communication*, vol. 48, no. 2, pp. 207–219, 2006.

[22] Y. Zhang, W. Chan, and N. Jaitly, “Very deep convolutional networks for end-to-end speech recognition,” in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2017.

[23] Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio, Q. Le, Y. Agiomyrgiannakis, R. Clark, and R. A. Saurous, “Tacotron: Towards end-to-end speech synthesis,” in *Proc. Interspeech*, 2017, pp. 4006–4010.

[24] R. J. Weiss, J. Chorowski, N. Jaitly, Y. Wu, and Z. Chen, “Sequence-to-sequence models can directly translate foreign speech,” in *Proc. Interspeech*, Aug. 2017.

[25] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in *Proc. International Conference on Machine Learning*, 2015.

[26] S. Xingjian, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo, “Convolutional LSTM network: A machine learning approach for precipitation nowcasting,” in *Advances in Neural Information Processing Systems*, 2015, pp. 802–810.

[27] M. Schuster and K. K. Paliwal, “Bidirectional recurrent neural networks,” *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, Nov. 1997.

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

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

[30] Y. Jia, Y. Zhang, R. J. Weiss, Q. Wang, J. Shen, F. Ren, Z. Chen, P. Nguyen, R. Pang, I. L. Moreno *et al.*, “Transfer learning from speaker verification to multispeaker text-to-speech synthesis,” in *Advances in Neural Information Processing Systems*, 2018.

[31] A. van den Oord, Y. Li, I. Babuschkin, K. Simonyan, O. Vinyals, K. Kavukcuoglu, G. van den Driessche, E. Lockhart, L. Cobo, F. Stimberg, N. Casagrande, D. Grewe, S. Noury, S. Dieleman, E. Elsen, N. Kalchbrenner, H. Zen, A. Graves, H. King, T. Walters, D. Belov, and D. Hassabis, “Parallel WaveNet: Fast high-fidelity speech synthesis,” in *Proc. International Conference on Machine Learning*, 2018.

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

[33] R. Cholewiak and C. Sherrick, “Tracking skill of a deaf person with long-term tactile aid experience: a case study.” *J Rehabil Res Dev*, vol. 23, no. 2, pp. 20–26, 1986.

[34] J. R. Hershey, Z. Chen, J. Le Roux, and S. Watanabe, “Deep clustering: Discriminative embeddings for segmentation and separation,” in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2016.

[35] K. Wilson, M. Chinen, J. Thorpe, B. Patton, J. Hershey, R. A. Saurous, J. Skoglund, and R. F. Lyon, “Exploring tradeoffs in models for low-latency speech enhancement,” in *Proc. International Workshop on Acoustic Signal Enhancement*, 2018.