Voice Conversion by Cascading Automatic Speech Recognition and Text-to-Speech Synthesis with Prosody Transfer

Jing-Xuan Zhang[1], Li-Juan Liu[2], Yan-Nian Chen[12], Ya-Jun Hu[2], Yuan Jiang[2], Zhen-Hua Ling[1], Li-Rong Dai[1]

[1]National Engineering Laboratory of Speech and Language Information Processing,
University of Science and Technology of China, P.R.China
[2]iFLYTEK Research, iFLYTEK Co., Ltd.

`nosisi@mail.ustc.edu.cn`, {`ljliu,yanchen5,yjhu,yuanjiang`}`@iflytek.com`,
{`zhling,lrdai`}`@ustc.edu.cn`

Abstract

With the development of automatic speech recognition (ASR) and text-to-speech synthesis (TTS) techniques, it's intuitive to construct a voice conversion system by cascading an ASR and TTS system. In this paper, we present an ASR-TTS method for voice conversion, which uses iFLYTEK ASR engine to transcribe the source speech into text and a Transformer TTS model with WaveNet vocoder to synthesize the converted speech from the decoded text. For the TTS model, a prosody code is used to describe the prosody information other than text and speaker information contained in speech. A prosody encoder is adopted to extract the prosody code. During conversion, the source prosody is transferred to converted speech by conditioning the Transformer TTS model with its code. Experiments were conducted to demonstrate the effectiveness of our proposed method. Our system also obtained the best naturalness and similarity in the mono-lingual task of Voice Conversion Challenge 2020.

Index Terms: voice conversion, automatic speech recognition, speech synthesize, prosody

1. Introduction

The objective of voice conversion (VC) is to modify certain aspects of the speech, such as accent or speaker identity, while preserving the linguistic content unchanged during this process [1, 2]. The conversion of speaker identity might be the most popular task of voice conversion, in which the utterance of a source speaker is processed to make it sounds like a target speaker. This work will also focus on the speaker conversion. Voice conversion has many practice applications, such as entertainment, personalized text-to-speech synthesize and voice anonymization.

Voice conversion has been well studied in recent decades. A conventional voice conversion system is based on the parallel training data, i.e., the same utterances are spoken by both the source and target speaker. Joint-density Gaussian mixture model (GMM) [3, 4], deep neural network (DNN) [5, 6] and recurrent neural nework (RNN) [7, 8] and sequence-to-sequence (seq2seq) model [9, 10] have been used for VC. Compared to the parallel VC, non-parallel VC is not constrained by the parallel training data. Many non-parallel VC methods have been proposed, such as varitional auto-encoder (VAE) based VC [11, 12], CycleGAN [13] and StarGAN [14] based VC.

Among those methods, recognition-synthesis based VC is one of the most popular [15, 16, 17, 18]. In this method, an automatic speech recognition (ASR) model is first trained to learn a mapping function from speech to text. Then, the linguistic descriptions are extracted by the ASR model. They are typically outputs from a certain layer of the ASR model, such as posteriorgrams (PPGs) [15] or content-related features [17]. Then, a synthesis model is trained to predict the target acoustic features using those linguistic descriptions as inputs. During the conversion stage, the source acoustic features are first encoded into the linguistic descriptions, which are then decoded to the target acoustic features. Our previous method using content-related features obtained the best performance in Voice Conversion Challenge 2018 [17]. Although its success, in those method, the source speaker information may not be completely separated from the PPGs or content-related features, which may harm the similarity of converted voice. Also, the model is constrained to perform the conversion frame by frame, since the PPGs or content-related features are frame level features. Therefore it's difficult to apply the seq2seq modeling, which has been proven the advantage in a variety of speech generation tasks [9, 19, 20, 21].

Recent years, automatic speech recognition and text-to-speech synthesize have become mature technique with research and development efforts from speech process communities. Under some circumstance, the automatic speech recognition has achieved accuracy levels comparable to human transcribers [22]. By using seq2seq acoustic modeling with neural vocoder [23], the synthesized speech is close to human quality [24, 21]. Given those facts, it's intuitive to construct a voice conversion system by cascading a ASR and TTS system directly, which is also a kind of recognition-synthesis based VC method. In this method, rather than PPGs or content-related features, text is used as the intermediate description to bridge different speakers, which contains only the sematic information. The iFLYTEK ASR engine and a Transformer TTS model are used in our experiments. In order to model the prosody information in addition to text and speaker identity, we further extract a global prosody code as input to the TTS model, which is similar to GST-Tacotron [25]. During conversion, the prosody code from the source utterance is transferred and used to condition the target Transformer TTS model [21] for generating more natural utterance. The overall scheme of our method is depicted in Figure 1.

In our experiments section, we compared our proposed method with the content-related features based baseline. Our method achieved the best performance on the mono-lingual task
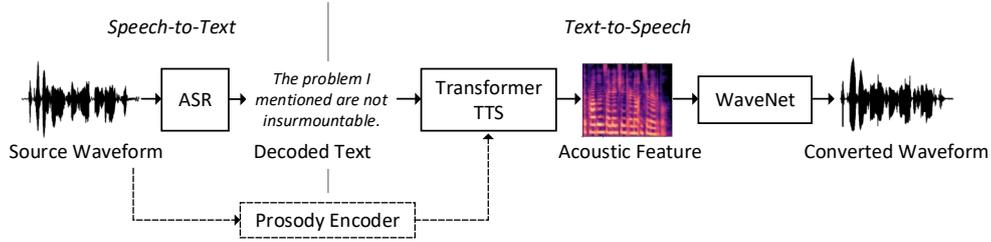
Figure 1: *Overview of our proposed method for voice conversion. The prosody encoder part is optional.*
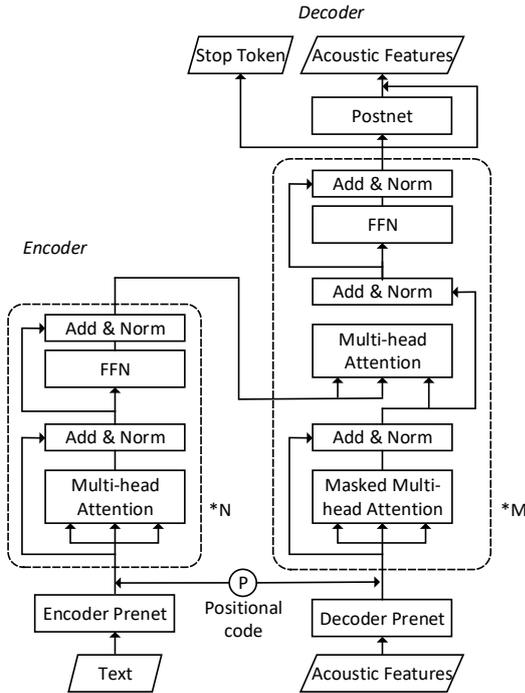
Figure 2: *Structure overview of Transformer TTS.*

Figure 3: *Multi-speaker Transformer TTS with prosody code conditioning. GRL denotes the gradient reversal layer.*

of Voice Conversion Challenge 2020, in terms of both naturalness and similarity. It's also noticeable our system has no statistical significant difference with the natural speech in terms of similarity.

2. Background

Transformer TTS is a seq2seq model that has been proposed for text-to-speech synthesis [21]. Transformer TTS mainly relies on the multi-head attention mechanism [26] instead of convolutions in CNN or recurrences in RNN for modeling sequential data. It is composed of a text encoder and a decoder module as illustrated in Figure 2. The text is first processed with a encoder Prenet then added with triangle positional embeddings. Then they are passed through stacks of encoder blocks. Each block contains two sub-layers, including a multi-head self-attention mechanism and a position-wise fully connected network. Also, residual connection followed by layer normalization is employed after each sub-layer. The decoder predicts the next frame auto-regressively, consuming the previously gener-
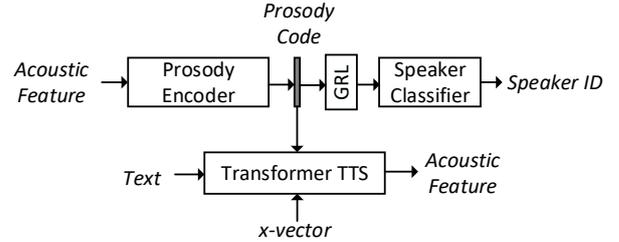
ated acoustic frame and the encoder outputs as the inputs. The acoustic frame is first passed through a decoder Prenet then fused with the positional code by adding. Then it's passed through a stacks of decoder blocks. Compared to the encoder block, the decoder block inserts an extra sub-layer that attends to the encoder outputs. Also, the self-attention in the decoder is masked to ensure the causality, i.e. the decoder should only make use of the history information each step. At last, in order to refine the acosutic features' reconstruction, a CNN based Postnet is employed to produce a residual, which is added with initial outputs. Also, a stop token is predicted to determine the end of generation process. For the detail of Transformer TTS, it suggested that the reader refers to the original paper [21].

Compared to the RNN based seq2seq TTS model, such as Tacotron, there're two advantages of Transformer TTS. First, without using any recurrent units, the computation of Transformer TTS is highly parallel during the training time, leading to higher training efficiency. Second, by using multi-head attention mechanism, two frames with any distance in a sequence can be associated in one step, thus the long-range dependency can be easily captured.

3. Proposed Method

3.1. ASR model

The ASR model is used to decode the text from source utterances, which determines the content of the converted utterances. By the definition of voice conversion task, linguistic content should not be changed during this process. Therefore, it's crucial that the ASR model to recognize the text information as much accurately as possible. For the ASR model, we compared two methods, including the commercial iFLYTEK ASR engine and an open-source ASR model based on ESPnet [27]. And results were reported in our experiments section.

Table 1: *Character Error Rate (CER) and Word Error Rate (WER) of iFLYTEK ASR engine and ESPnet based ASR model.*

	CER (%)	WER (%)
iFLYTEK	0.958	2.954
ESPnet	2.415	7.0l57

Table 2: *Mean opinion score (MOS) and 95% confidence interval of our proposed method and the baseline. Natural denotes the natural target speech.*

	Naturalness	Similarity
VCC2018+	3.781±0.070	3.610±0.076
proposed	3.802±0.067	3.739±0.074
Natural	3.815±0.077	3.703±0.08

3.2. TTS model

Transformer TTS is extended to the multi-speaker condition by using speaker x-vectors [28] as additional inputs to its decoder. The x-vectors are projected by linear transformation then added to each frame of encoder outputs. Another difference with the original Transformer TTS paper is that we don't use the encoder Prenet, since no explicit improvement was found in our preliminary experiments. The amount of target training data is usually small in voice conversion task. Thus the model often suffers from over-fitting effect if no external dataset is exploited. Therefore, in order to improving the generalization capacity of the model, it is first pre-trained on a multi-speaker dataset, and then adapted to the target speaker by finetuning.

3.3. Prosody transfer

We assume that the speech signal contains not only the text and speaker identity information, but also the prosody information. Therefore, a prosody code is further provided to the synthesis model to describe the prosodic ingredients of the speech. In our method, a prosody encoder is employed to extract a global prosody code for each utterance as shown in Figure 3. The Transformer TTS model is further conditioned on the prosody code. Specifically, the code is projected by linear transformation then added to each frame of encoder outputs. During conversion, the source prosody code is transferred and used to synthesize the converted utterance to make it more natural sounded. However, because that the TTS model is pre-trained on the multi-speaker dataset, the desired prosody information can be easily entangled with the speaker identity information. Therefore, we further adopted a speaker adversarial loss to regularize the extracted code to be speaker-independent. To this end, a speaker classifier is adopted to predict the speaker identity from the prosody code. While the prosody encoder is optimized with the opposite objective to lower the classifier's accuracy of prediction.

In our implementation, a gradient reversal layer is inserted between the prosody code and the classifier. The speaker classifier is updated four times for each update of the prosody encoder. It is critical to keep the classifier close to the optimal so that it can backpropagate useful gradients rather than noises. Our prosody encoder is only updated during the multi-speaker pre-training stage while it's frozen during fine-tuning on the target speaker.

4. Experiments

4.1. Experiment conditions

Dataset from the mono-lingual task of Voice Conversion Challenge 2020 was adopted, which contained 4 source English speakers (SEF1, SEF2, SEM1 and SEM2) and 4 target English speakers (TEF1, TEF2, TEM1 and TEM2). Each speaker contained 70 utterances for training, 20 of which were parallel for source and target speakers and the remaining were non-parallel. Our model only relies on the target utterances for model adapta-

tion. Therefore, the non-parallel part of source utterances were used as our internal evaluation set. For evaluation in Voice Conversion Challenge 2020, 25 test utterances were further provided for each source speaker. And they were converted to each target speaker. LibriTTS corpus[1] was adopted for pretraining our model. We used about 460 hours training data from 1150 speakers.

For ASR model, we compared the iFLYTEK ASR engine and an open-source ASR model[2] based on ESPnet [27] and the results on our internal evaluation set were presented in Table 1. We observed that the iFLYTEK ASR engine achieved better performance than the ESPnet based one. Therefore, the former was adopted in our experiments.

The Transformer TTS model contained 6 blocks for both encoder and decoder. The number of attention head was 4 and the layer width was 1536. The architecture of the prosody encoder followed the reference encoder of GST-Tacotron [25], producing a 128-dimensional prosody code for each utterance. The speaker classifier was a 3×512 DNN. 512-dimensional x-vectors were extracted by Kaldi toolkit[3]. For acoustic features, 80-dimensional Mel-spectrograms were extracted with Mel filters spanning from 80 Hz to 7600 Hz. We used Adam optimizer and Noam scheduler [26] for adjusting the learning rate. The batch size was 120, which was later set as 16 during finetuning. At last, WaveNet vocoder was trained to model 24kHz-16bit waveforms for each target speaker.

4.2. Comparison with the baseline

We first compared our proposed ASR-TTS method without using prosody code to a baseline method, which was an improved version of N10 system in Voice Conversion Challenge 2018, denoted by **VCC2018+**. Compared to N10 system, this baseline adopted a new auto-regressive synthesis model. Also, a higher quality WaveNet vocoder was adopted. For the details of this model, the reader can refer to another our paper for the cross-lingual task of Voice Conversion Challenge 2020 [29].

We conducted the subjective listening test in terms of both naturalness and similarity. 40 utterances for each method were randomly selected from our internal evaluation set. We used Amazon Mechanical Turk[4] platform and at least 30 native listeners participated in each of our experiments. They were required to used headphones during the test and the stimuli were presented in random order. Table 2 summarized the results. We observed from the table that our proposed method achieved similar naturalness ($p = 0.564$ in t-test) and higher similarity ($p = 1.19 \times 10^{-3}$ in t-test) than the VCC2018+ baseline. And it even obtained similar results to the natural target speech

---

[1] http://www.openslr.org/60/
[2] https://drive.google.com/file/d/
1BtQvAnsFvVi-dp_qsaFP7n4A_5cwnlR6/view?usp=
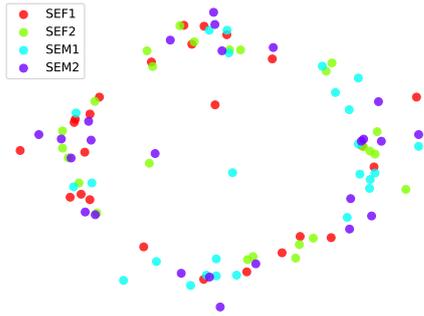drive_open
[3] https://kaldi-asr.org/
[4] https://www.mturk.com/

Figure 4: *Visualization of prosody code from four source speakers. Each point denotes a prosody code from an utterance.*

Table 3: *Mean opinion score (MOS) and 95% confidence interval of the proposed method with (w pc) and without (w/o pc) using prosody code.*

Target speaker	Method	Naturalness	Similarity
TEF1	*w/o pc*	3.673±0.188	3.593±0.193
	w pc	3.713±0.183	3.553±0.189
TEF2	*w/o pc*	3.705±0.163	3.771±0.165
	w pc	3.757±0.160	3.871±0.162
TEM1	*w/o pc*	3.729±0.162	3.843±0.163
	w pc	3.781±0.160	3.724±0.163
TEM2	*w/o pc*	3.711±0.162	3.632±0.178
	w pc	3.668±0.166	3.616±0.181
Average	*w/o pc*	3.707±0.083	3.721±0.087
	w pc	3.733±0.082	3.704±0.086

($p = 0.789$ and $p = 0.461$ for naturalness and similarity, respectively).

4.3. Using prosody transfer strategy

In order to analyze the latent space of prosody code, they were extracted from four source speakers, i.e., SEF1, SEF2, SEM1 and SEM2, then projected to 2 dimension by t-SNE for visualization. The 25 test utterances for each speaker were used and the results were presented in Figure 4. We observed from this figure the prosody code from each speaker equally distributed in the same latent space, which demonstrated the speaker-indepentent property of the prosody code.

Based on the preliminary experiments, we heuristically scaled the prosody code with a factor of 1.5 during conversion. And the proposed method with and without prosody code were compared by subjective listening test. 10 sentences from internal evaluation set were randomly selected for each target speaker, resulting a total of 40 sentences for evaluation. The results were grouped by the target speaker as presented in Table 3. As we can see from the table, the proposed method with and without prosody code obtained close naturalness and similarity score (all have $p > 0.1$). Averagely, *w pc* method obtained slightly better naturalness but slightly lower similarity than *w/o pc* method. For the target speaker TEF2, *w pc* method achieved slightly better results in both naturalness and similarity. It should be notice that our *w/o pc* method already achieved comparable results with the natural target speech, as shown in previous section. Therefore, there's no clear advantage of using prosody transfer strategy in our experiments. To explore this strategy with dataset containing expressive utterances will be an interesting research direction in the future.

4.4. Voice Conversion Chanllenge 2020

Different conversion methods were selected according to their performance on the conversion pairs in Voice Conversion Challenge 2020[5]. Based on the experimental results in previous section, we adopted the proposed method with prosody transfer strategy when the target speaker was TEF2. For SEM1-to-TEM1 and SEM1-to-TEM2, VCC2018+ method was used since the proposed ASR-TTS achieved similar results to it on those conversion pairs. For remaining conversion pairs, we used the proposed method without prosody transfer strategy.

The subjective listening results, which were collected from 68 native English speakers, were shown in Figure 5. As we can
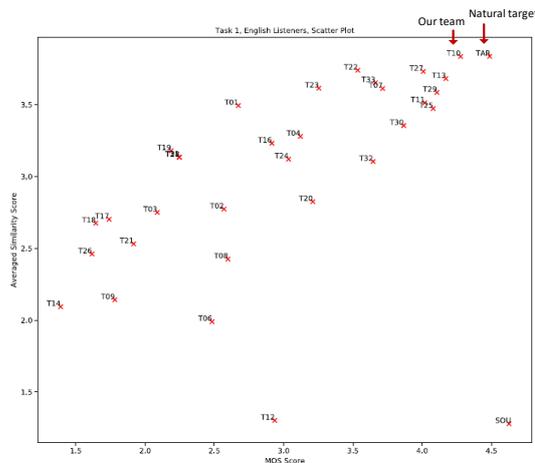

Figure 5: *Average similarity score and naturalness MOS for different teams in Voice Conversion Challenge 2020.*

observed from this figure, our method achieved the best performance among the participating teams. Also, it's noticeable our method achieved similar similarity score with the natural target speech. According to further results provided by the organizer, converted utterances by our method had no statistical significant difference with the natural target speech in terms of similarity.

5. Conclusion

We present a voice conversion method by cascading a ASR and TTS module in this paper. iFLYTEK ASR engine is used for decoding text from the source utterance and a Transformer TTS model is trained for synthesizing the target speech. The Transformer TTS is pretrained on a multi-speaker dataset then fine-tuned on the target speaker in order to boosting its generalization capacity. A prosody transfer technique is further applied, in which a prosody code is extracted by a prosody encoder from the source then used to condition the target TTS model. Our experimental results showed the effectiveness of proposed method for voice conversion. However, the ASR model is still imperfect. The speech recognition errors will lead to the change of the linguistic content of converted speech. Investigating on how to minimize those errors by using a mix of recognized text and the content-related features will be investigated in our future work.

---

[5] http://www.vc-challenge.org/

6. References

[1] D. G. Childers, K. Wu, D. M. Hicks, and B. Yegnanarayana, "Voice conversion," *Speech Communication*, vol. 8, no. 2, pp. 147–158, 1989.

[2] D. G. Childers, B. Yegnanarayana, and K. Wu, "Voice conversion: Factors responsible for quality," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1985, pp. 748–751.

[3] A. Kain, "Spectral voice conversion for text-to-speech synthesis," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 1, 1998, pp. 285–288.

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

[5] S. Desai, E. V. Raghavendra, B. Yegnanarayana, A. W. Black, and K. Prahallad, "Voice conversion using artificial neural networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2009, pp. 3893–3896.

[6] S. Desai, A. W. Black, B. Yegnanarayana, and K. Prahallad, "Spectral mapping using artificial neural networks for voice conversion," *IEEE Transactions on Audio Speech and Language Processing*, vol. 18, no. 5, pp. 954–964, 2010.

[7] L. Sun, S. Kang, K. Li, and H. Meng, "Voice conversion using deep bidirectional long short-term memory based recurrent neural networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 4869–4873.

[8] T. Nakashika, T. Takiguchi, and Y. Ariki, "Voice conversion using RNN pre-trained by recurrent temporal restricted Boltzmann machines," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 3, pp. 580–587, 2015.

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

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

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

[12] C.-C. Hsu, H.-T. Hwang, Y.-C. Wu, Y. Tsao, and H.-M. Wang, "Voice conversion from unaligned corpora using variational autoencoding Wasserstein generative adversarial networks," in *Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2017, pp. 3364–3368.

[13] T. Kaneko and H. Kameoka, "CycleGAN-VC: Non-parallel voice conversion using cycle-consistent adversarial networks," in *European Signal Processing Conference (EUSIPCO)*, 2018, pp. 2114–2117.

[14] H. Kameoka, T. Kaneko, K. Tanaka, and N. Hojo, "StarGAN-VC: Non-parallel many-to-many voice conversion with star generative adversarial networks," *arXiv preprint arXiv:1806.02169*, 2018.

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

[16] H. Miyoshi, Y. Saito, S. Takamichi, and H. Saruwatari, "Voice conversion using sequence-to-sequence learning of context posterior probabilities," in *Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2017, pp. 1268–1272.

[17] L.-J. Liu, Z.-H. Ling, Y. Jiang, M. Zhou, and L.-R. Dai, "WaveNet vocoder with limited training data for voice conversion," in *Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2018, pp. 1983–1987.

[18] S. Liu, J. Zhong, L. Sun, X. Wu, X. Liu, and H. Meng, "Voice conversion across arbitrary speakers based on a single target-speaker utterance," in *Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2018, pp. 496–500.

[19] R. Liu, X. Chen, and X. Wen, "Voice conversion with transformer network," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7759–7759.

[20] Y. Wang, R. J. 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 *Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2017, pp. 4006–4010.

[21] N. Li, S. Liu, Y. Liu, S. Zhao, and M. Liu, "Neural speech synthesis with transformer network," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 6706–6713.

[22] A. Stolcke and J. Droppo, "Comparing human and machine errors in conversational speech transcription," in *INTERSPEECH*, 08 2017, pp. 137–141.

[23] A. V. Den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior, and K. Kavukcuoglu, "WaveNet: A generative model for raw audio," in *9th ISCA Speech Synthesis Workshop (SSW9)*, 2016, pp. 125–125.

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

[25] Y. Wang, D. Stanton, Y. Zhang, R. Skerry-Ryan, E. Battenberg, J. Shor, Y. Xiao, F. Ren, Y. Jia, and R. A. Saurous, "Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis," *arXiv preprint arXiv:1803.09017*, 2018.

[26] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, u. Kaiser, and I. Polosukhin, "Attention is all you need," in *International Conference on Neural Information Processing Systems*, 2017, pp. 6000–6010.

[27] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. Enrique, Y. Soplin, J. Heymann, M. Wiesner, N. Chen, A. Renduchintala, and T. Ochiai, "ESPnet: End-to-end speech processing toolkit," in *INTERSPEECH*, 2018, pp. 2207–2211.

[28] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5329–5333.

[29] L.-J. Liu, Y.-N. Chen, J.-X. Zhang, Y. Jiang, Y.-J. Hu, Z.-H. Ling, and L.-R. Dai, "Non-parallel voice conversion with autoregressive conversion model andduration adjustment," in *ISCA Joint Workshop for the Blizzard Challenge and Voice Conversion Challenge*. ISCA, 2020.