The Sequence-to-Sequence Baseline for the Voice Conversion Challenge 2020: Cascading ASR and TTS

Wen-Chin Huang[1], Tomoki Hayashi[1], Shinji Watanabe[2], Tomoki Toda[1]

[1]Nagoya University, Japan
[2]Johns Hopkins University, USA

`wen.chinhuang@g.sp.m.is.nagoya-u.ac.jp`

Abstract

This paper presents the sequence-to-sequence (seq2seq) baseline system for the voice conversion challenge (VCC) 2020. We consider a naive approach to voice conversion (VC), which is to first transcribe the input speech with an automatic speech recognition (ASR) model, followed by the use of transcriptions to generate the voice of the target with a text-to-speech (TTS) model. We revisit this method under a sequence-to-sequence (seq2seq) framework by utilizing ESPnet, an open-source end-to-end speech processing toolkit, and the many well-configured pretrained models provided by the community. Official evaluation results show that our system comes out on top among the participating systems in terms of conversion similarity, demonstrating the promising ability of seq2seq models to convert speaker identity. The implementation is made open source at `https://github.com/espnet/espnet/tree/master/egs/vcc20`.

Index Terms: voice conversion, voice conversion challenge, espnet, automatic speech recognition, text-to-speech

1. Introduction

Voice conversion (VC) is a technique to transform the para-/non-linguistic characteristics included in a source speech waveform into a different one while preserving linguistic information [1,2]. VC has great potential in the development of various new applications such as speaking aid devices for vocal impairments, expressive speech synthesis, silent speech interfaces, or accent conversion for computer-assisted language learning.

The aim of the voice conversion challenge (VCC)[1] is to better understand different VC techniques built on a freely available common dataset to achieve a common goal and share views about unsolved problems and challenges faced by current VC techniques. The challenges focused on *speaker conversion*, for which VC models are built to automatically transform the voice identity. In the third version, VCC2020 [3], two new tasks are considered. The first task is *semiparallel* VC within the same language, in which only a small subset of the training set is parallel with the rest being nonparallel. The second task is *cross-lingual* VC, in which the training set of the source speaker is different from that uttered by the target speaker in language and content, thus nonparallel in nature. In conversion, the source speaker's voice in the source language is converted as if it was uttered by the target speaker while keeping linguistic contents unchanged.

It would be worth discussing two important factors when designing a VC system: data and model. First, from the data viewpoint, in either of the VCC2020 tasks, techniques for dealing with nonparallel data should be developed. In the liter-
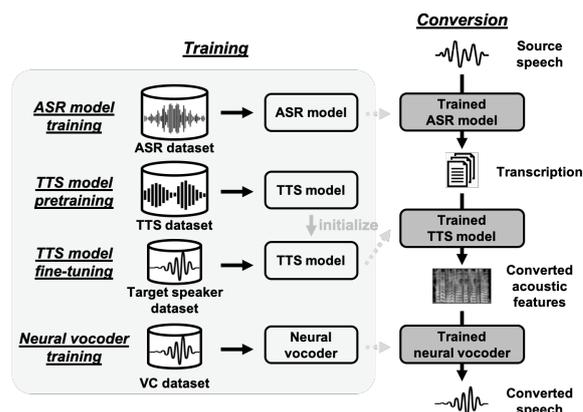

Figure 1: *Training and conversion processes of* ASR+TTS.

ature, a promising paradigm for nonparallel VC is through a recognition-synthesis framework. The idea is to first extract the linguistic contents from the source speech, followed by blending with the target speaker characteristics to generate the converted speech. Methods implementing this framework can be divided according to the type of linguistic representation. The first type encodes representations with an automatic speech recognition (ASR) model, for which a popular choice is the phonetic posteriorgram (PPG) [4,5]. A synthesis model is then trained to generate the voice of the target speaker. The second type usually employs an autoencoder-like model that estimates the recognizer and synthesizer simultaneously by implicitly factorizing the linguistic and speaker representations [6–10].

From the model viewpoint, we have observed how sequence-to-sequence (seq2seq) models [11] change the game in many research fields in only half a decade, and speech processing is no exception. Its application in VC is especially attractive because compared with conventional frame-based methods that perform conversion frame-by-frame, seq2seq models can implicitly learn the complex alignment and relationship between the source and target sequences to generate outputs of various lengths. It is therefore a natural choice to convert prosody including the speaking rate and F0 contour, which is closely related to speaker characteristics. As a result, seq2seq-based VC has been a promising approach in terms of conversion similarity [12–15].

In this paper, we describe the seq2seq baseline system for VCC2020. Our system is a cascade of seq2seq-based ASR and text-to-speech (TTS) models, which we will refer to as ASR+TTS. A suitable baseline system should meet the following requirements:

- The system should be a simple and easy-to-use starting ground for newcomers to base their work on.

---

[1]`http://www.vc-challenge.org/`

- The system should be an open-source project made publicly available to benefit potential future researchers.
- The system should serve as a competitive benchmark.

With these requirements in mind, we implemented the system using ESPnet, a well-developed open-source end-to-end (E2E) speech processing toolkit [16, 17], and made as much use of publicly available datasets as possible. Although it is generally believed that a simply cascading system used to perform a certain task is inferior to an end-to-end model, benefitting from recent advances in ASR and TTS models, as well as efforts such as implementation and hyperparameter tuning that are dedicated by the open-source community, we will show that our system is not only easy to use but also serves as a strong competing system in VCC2020.

2. System Overview

A naive approach to VC is a cascade of an ASR model and a TTS model. Although this method is not new, by revisiting this method using seq2seq models, we can model various aspects of prosody such as pitch, duration, and speaking rate, which are usually not well considered in the literature. Conceptually, the ASR model acts similarly to a speaker normalizer that first normalizes the input speech such that attributes of the source speaker are filtered out and only the linguistic content remains. Then, the TTS model functions to add speaker information to the recognition result so that the converted speech sounds similarly to the target speaker.

Our system, as depicted in Figure 1, consists of three modules: a speaker-independent ASR model, a separate speaker-dependent TTS model for each target speaker, and a neural vocoder that synthesizes the final speech waveform.

ASR model. ASR models are usually trained with a multi-speaker dataset, thus speaker-independent in nature. For both tasks 1 and 2, the source speech is always in English, so an English transcription is first obtained using an ASR model.

TTS model. In the TTS literature, training in a speaker-dependent manner, rather than training speaker-independently is a common practice since the former usually outperform the latter. However, the size of the training set of each target speaker in VCC2020 is too limited for seq2seq TTS learning. In light of this, we employ a pretraining and finetuning scheme that first pretrains on large TTS datasets followed by finetuning on the limited target speaker dataset [18]. This allows us to successfully train on even approximately five minutes of data.

Neural vocoder. In recent years, neural waveform generation modules (also known as vocoders) have brought significant improvement to VC. In this work, we used the Parallel WaveGAN (PWG) [19], since it enables high-quality, real-time waveform generation. We adopted an open-source implementation[2] and integrated it with ESPnet.

Our implementation was built on the E2E speech processing toolkit ESPnet [16, 17], which provides various useful utility functions and properly tuned pretrained models.

3. ASR Implementation

3.1. Data

Since the input is always English, we used the Librispeech dataset [20], which contains 960 hours of English speech data from over 2000 speakers.

Table 1: *TTS training datasets in task 2. "phn" and "char" stand for phoneme and character, respectively.*

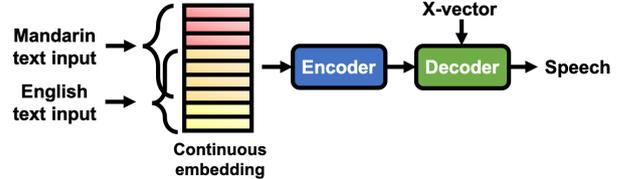| Lang.	Dataset	Spkrs	Hours	Input
Eng.	M-AILABS [27]	2	32	phn or char
Ger.	M-AILABS [27]	5	190	char
Fin.	CSS10 [28]	1	10	char
Man.	CSMSC [29]	1	12	pinyin

Figure 2: *Illustration of bilingual TTS used in task 2.*

3.2. Model

The backbone of the ASR model was the Transformer [21–23]. The model was trained in an end-to-end manner using a hybrid CTC/attention loss [24], and a recurrent-neural-network-based language model (RNNLM) was used for decoding. We directly used a pretrained model (including the RNNLM) provided by ESPnet.

4. TTS Implementation

We are faced with a more difficult challenge in implementing the TTS model. In task 2, the input language is different from the languages of the training data. That is, the TTS model should learn the voice of an unseen language. This is sometimes referred to as cross-lingual voice cloning [25]. As there has not been a standard, promising protocol especially when only five minutes of training data is available, we adopt a simple method that constructs x-vector-based [26] bilingual TTS models by pretraining with corpora of English and the target language and finetuning with the target language.

4.1. Data

The target language for task 1 is English. Therefore, for pretraining, we used the multispeaker LibriTTS [30] dataset, which contains around 250 hours of English data from over 2000 speakers. In task 2, the target languages are German, Finnish, and Mandarin. Considering the open-source implementation, we wanted to avoid using commercial or private datasets. Unfortunately, under such constraint, there is not much choice, and the available datasets at the time we developed the system were large but contained only data from a single speaker or a few speakers, as shown in Table 1. Although it has been shown that combining imbalanced multispeaker datasets improves performance [31], this effect remains unknown in the cross-lingual setting. To this end, for the English data, we decided to use not the LibriTTS dataset, which has many speakers yet a small amount of data per speaker, but the M-AILABS dataset [27], which has a large amount of data from only a few speakers. Finally, since the task 2 datasets were of different sampling rates, we downsampled all task 2 data to 16 kHz. As for the x-vector extractor, the Kaldi toolkit was used and the model was pretrained on VoxCeleb [32].
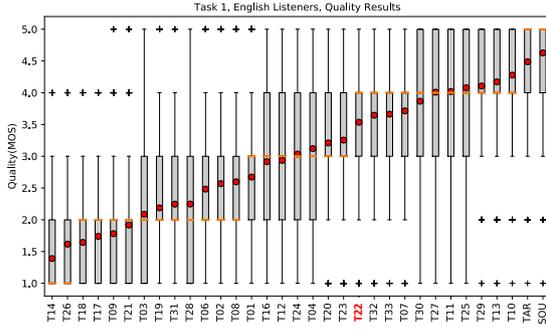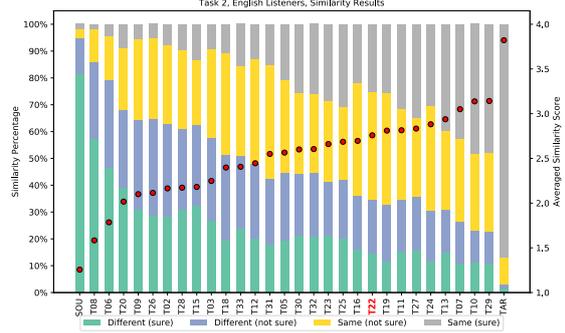
(a) *Naturalness results for task 1.*
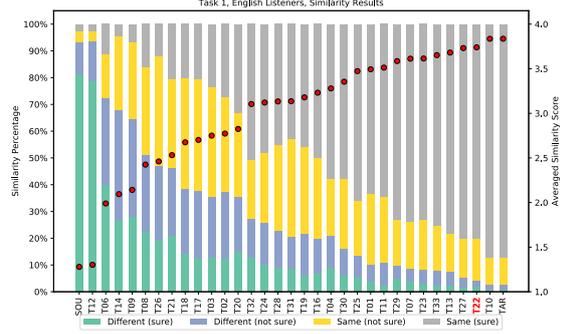
(b) *Similarity results for task 1.*
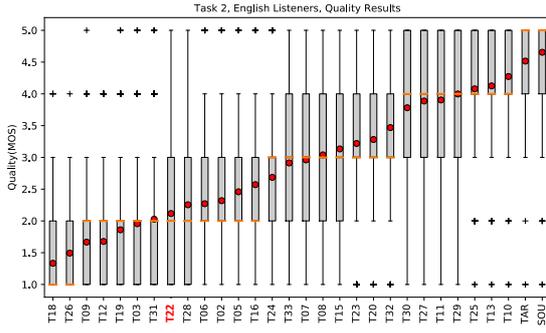
(c) *Naturalness results for task 2.*

(d) *Similarity results for task 2.*

Figure 3: *Official evaluation results of VCC2020. Our system T22 is highlighted in red.*

4.2. Model

We used an x-vector-based [26] multispeaker TTS model [33] with a Transformer backbone [34]. The input was a linguistic representation sequence and the output was the mel filterbank sequence extracted from the (optionally downsampled) waveform. In task 1, since the input is always English, the model simply takes English characters an input.

However, in task 2, it is nontrivial to decide the input representation since it is often language-dependent. For example, there is no overlap in the text representation between Mandarin and English [25]. When we finetune a pretrained model for a Mandarin speaker, since the Mandarin corpus does not contain English words, the model has no clue how the target speaker pronounces English words. This mismatch may cause quality degradation. Below, we describe how we alleviate this issue.

We used a shared input embedding space when training the bilingual TTS model. In neural TTS, the input embedding lookup table is a projection from discrete input symbols to continuous representation and is trained with the rest of the model by backpropagation. It is useful in that the model can implicitly learn how to pronounce each input token, such that different tokens with similar pronunciations can have similar embeddings. The assumption here is that there is an overlap between the input representations of the two languages. For example, if we train a Mandarin/English TTS model, the "ah" phoneme in English and "a" pinyin representation may have similar embeddings. As a result, even if only "a" is seen during training, by learning how the target speaker pronounces such a vowel, the model may still know how to pronounce "ah".

For the Mandarin/English TTS, we used phonemes and pinyin as the input, whereas for the Finnish/English and Ger-

man/English TTS, we used characters as the input. In the finetuning stage, the parameters are updated using the training utterances of the target speaker, except that the embedding lookup table in Figure 2 is fixed.

5. Neural Vocoder Implementation

The PWG had a non-autoregressive (non-AR) WaveNet-like architecture and was trained by jointly optimizing a multiresolution spectrogram loss and a waveform adversarial loss [19]. The input was a mel filterbank and the output was a raw waveform. For each task, we trained a separate PWG using the training data from all available speakers. That is, data of 8 and 10 speakers were used to train PWGs for tasks 1 and 2, respectively. Notably, in task 2, although mel filterbanks were extracted from a 16kHz waveform as mentioned in Sections 4.1 and 4.2, we still map them to a 24kHz waveform during training, as the quality degradation from such a mismatch has shown to be acceptable [30].

6. Challenge Results

6.1. VCC2020 Dataset

The VCC2020 database had two male and two female English speakers as the source speakers. For task 1, two male and two female English speakers were chosen as target speakers, and one male and one female for each of Finnish, German, and Mandarin in task 2. Each of the source and target speakers has a training set of 70 sentences, which is around 5 minutes of speech data. Note that in task 1, the target and source speakers have 20 parallel sentences, where the remaining 50 sen-

Table 2: *Character/word error rates (CER/WER) (%) calculated using a pretrained ASR model. The scores are averaged over all target speakers.*

	Input		Task 1		Task 2	
Source	CER	WER	CER	WER	CER	WER
SEF1	2.9	6.5	12.1	22.1	19.9	34.3
SEF2	1.4	3.7	12.6	22.7	21.4	36.2
SEM1	0.2	0.9	14.2	20.1	20.3	36.8
SEM2	2.9	7.5	18.5	30.9	22.7	38.0

tences are different. Twenty-five test sentences for evaluation are shared for tasks 1 and 2.

6.2. Evaluation Protocol

The VCC2020 organizing committee conducted a large-scale subjective test on all submitted systems for both tasks 1 and 2. The evaluations included naturalness and similarity tests. In the naturalness test, a five-point mean opinion score (MOS) test was adopted, where listeners were asked to rate the naturalness of each speech clip from 1 to 5. In the similarity test, listeners were presented with a converted utterance and a ground truth target utterance, and they were asked to decide whether or not the two utterances were spoken by the same person on a four-point scale. Figure 3 shows the overall results[3].

6.3. Task 1 Results

Figures 3(a) and 3(b) show the overall results for task 1. For naturalness, our system received a MOS of about 3.5, which ranks 11 out of all the 31 submitted systems in task 1. This shows that, as many systems are specifically designed for VC, simply combining state-of-the-art ASR and TTS systems can already achieve competitive results, owing to the well-developed technologies in the two research fields. The performance gap between our system and the superior teams may originate from the difficulty in finetuning the TTS model with only 70 utterances. As for similarity, our system had a similarity score of approximately 90%, indicating that about 90% of the converted utterances were considered spoken by the same target speaker by the participants. This made our system rank second among all teams, which is strong evidence of the superiority of seq2seq models when it comes to converting the speaker identity.

6.4. Task 2 Results

Figures 3(c) and 3(d) show the overall results for task 2. For naturalness, our system had a MOS of about 2.0, ranking 21 out of all the 28 submitted systems in task 2, which is much worse than the performance in task 1. On the other hand, our system ranked 9 among the 28 teams in the similarity test. It can be inferred from the two results that our system can still well capture the speaker characteristics owing to the power of seq2seq modeling; however, it suffers from severe quality degradation. This is possibly due to the limited training data and the lack of pretraining data, as well as the difficulty in handling the cross-lingual data using the overly simple TTS model we implemented.

---

[3]Although the official report contained results from Japanese and English listeners, here we only report results of English listeners since the two listener groups share a similar tendency.

7. Analysis on Linguistic Contents

A potential threat of the cascading paradigm is that error in early stages might propagate to downstream models. In our proposed method, the recognition failure in the first ASR stage might harm the linguistic consistency in VC. We examine this phenomenon by measuring the intelligibility with an Transformer-ASR model trained on LibriSpeech, provided in ESPnet.

Table 2 shows the results. First, the error rates of the input source speech were not severe because they are similar to that of the test set of LibriSpeech. However, the scores of the converted speech are much worse, indicating that the imperfect TTS modeling is the main cause of intelligibility degradation. We also observe that the error rates of task 2 are much higher than those of task 1, which is consistent with the results in Section 6.4.

8. Conclusion and Discussion

In this paper, we described the seq2seq baseline system of the VCC2020, including the intuition, system design, training datasets, and results. Built on the E2E, seq2seq framework, our ASR+TTS baseline served as a simple starting point and a benchmark for participants. Subjective evaluation results released by the organizing committee showed that our system is a strong baseline in terms of conversion similarity, confirming the effectiveness of seq2seq modeling. The results also demonstrate the naive yet promising power of combining state-of-the-art ASR and TTS models. Yet, there is still much room for improvement, and below we discuss several possible directions that might be addressed in an advanced version.

Enhance the pretraining data. As stated in Section 4.1, there was not much choice for pretraining data in task 2 under the open-source constraint. Using a multispeaker pretraining dataset as in task 1 might improve the performance. Moreover, using datasets with a higher sampling rate might also improve the quality of the vocoder.

Utilize linguistic knowledge. One principle of E2E learning is to use as less domain-specific knowledge as possible. That is, the system performance is expected to be improved when such knowledge is utilized. For example, as reported in [25], using phoneme inputs can greatly improve multilingual TTS systems, but we were unable do so in task 2 owing to the unfamiliarity with target languages such as Finnish and German.

Select an advanced multispeaker TTS model. The multispeaker TTS model [33] we adopted was rather naive, and a more state-of-the-art model similar to that in [35] might improve the performance.

Improve the neural vocoder. We adopted a non-AR neural vocoder for fast generation, but it is generally believed that AR neural vocoders are still superior. Because this is a popular research field, it is expected that real-time neural vocoders maintaining the output quality will soon be developed. Also, finetuning the vocoders can further improve the performance, as stated in Section 5.

9. Acknowledgement

10. References

[1] Y. Stylianou, O. Cappe, and E. Moulines, "Continuous probabilistic transform for voice conversion," *IEEE Transactions on Speech and Audio Processing*, vol. 6, no. 2, pp. 131–142, 1998.

[2] 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.

[3] Y. Zhao, W.-C. Huang, X. Tian, J. Yamagishi, R. K. Das, T. Toda, T. Kinnunen, and Z. Ling, "Voice conversion challenge 2020 — intra-lingual semiparallel and cross-lingual voice conversion —," in *ISCA Joint Workshop for the Blizzard Challenge and Voice Conversion Challenge 2020*, 2020, pp. XXX–XXX.

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

[5] L.-J. Liu, Z.-H. Ling, Y. Jiang, M. Zhou, and L.-R. Dai, "WaveNet Vocoder with Limited Training Data for Voice Conversion," in *Proc. Interspeech*, 2018, pp. 1983–1987.

[6] 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 *Proc. APISPA ASC*, 2016, pp. 1–6.

[7] ——, "Voice conversion from unaligned corpora using variational autoencoding wasserstein generative adversarial networks," in *Proc. Interspeech*, 2017, pp. 3364–3368.

[8] J.-C. Chou, C.-C. Yeh, H.-Y. Lee, and L.-S. Lee, "Multi-target Voice Conversion without Parallel Data by Adversarially Learning Disentangled Audio Representations," in *Proc. Interspeech*, 2018, pp. 501–505.

[9] K. Qian, Y. Zhang, S. Chang, X. Yang, and M. Hasegawa-Johnson, "AutoVC: Zero-Shot Voice Style Transfer with Only Autoencoder Loss," in *Proceedings of the 36th International Conference on Machine Learning*, 2019, pp. 5210–5219.

[10] W.-C. Huang, H. Luo, H.-T. Hwang, C.-C. Lo, Y.-H. Peng, Y. Tsao, and H.-M. Wang, "Unsupervised Representation Disentanglement Using Cross Domain Features and Adversarial Learning in Variational Autoencoder Based Voice Conversion," *IEEE Transactions on Emerging Topics in Computational Intelligence*, pp. 1–12, 2020.

[11] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to Sequence Learning with Neural Networks," in *Advances in Neural Information Processing Systems*, 2014, pp. 3104–3112.

[12] 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)*, May 2019, pp. 6805–6809.

[13] J. Zhang, Z. Ling, L. Liu, Y. Jiang, and L. 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.

[14] J. Zhang, Z. Ling, and L. Dai, "Non-parallel sequence-to-sequence voice conversion with disentangled linguistic and speaker representations," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 540–552, 2020.

[15] W.-C. Huang, T. Hayashi, Y.-C. Wu, H. Kameoka, and T. Toda, "Voice transformer network: Sequence-to-sequence voice conversion using transformer with text-to-speech pretraining," *arXiv preprint arXiv:1912.06813*, 2019, to appear in Interspeech 2020.

[16] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. E. Y. Soplin, J. Heymann, M. Wiesner, N. Chen, A. Renduchintala, and T. Ochiai, "ESPnet: End-to-End Speech Processing Toolkit," in *Proc. Interspeech*, 2018, pp. 2207–2211.

[17] T. Hayashi, R. Yamamoto, K. Inoue, T. Yoshimura, S. Watanabe, T. Toda, K. Takeda, Y. Zhang, and X. Tan, "ESPnet-TTS: Unified, Reproducible, and Integratable Open Source End-to-End Text-to-Speech Toolkit," *arXiv preprint arXiv:1910.10909*, 2019.

[18] K. Inoue, S. Hara, M. Abe, T. Hayashi, R. Yamamoto, and S. Watanabe, "Semi-supervised speaker adaptation for end-to-end speech synthesis with pretrained models," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 7634–7638.

[19] R. Yamamoto, E. Song, and J.-M. Kim, "Parallel WaveGAN: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 6199–6203.

[20] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "LibriSpeech: An ASR corpus based on public domain audio books," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 5206–5210.

[21] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is All you Need," in *Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008.

[22] L. Dong, S. Xu, and B. Xu, "Speech-Transformer: A No-Recurrence Sequence-to-Sequence Model for Speech Recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 5884–5888.

[23] S. Karita, N. E. Y. Soplin, S. Watanabe, M. Delcroix, A. Ogawa, and T. Nakatani, "Improving Transformer-Based End-to-End Speech Recognition with Connectionist Temporal Classification and Language Model Integration," in *Proc. Interspeech*, 2019, pp. 1408–1412.

[24] S. Watanabe, T. Hori, S. Kim, J. R. Hershey, and T. Hayashi, "Hybrid CTC/Attention Architecture for End-to-End Speech Recognition," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 8, pp. 1240–1253, 2017.

[25] Y. Zhang, R. J. Weiss, H. Zen, Y. Wu, Z. Chen, R. Skerry-Ryan, Y. Jia, A. Rosenberg, and B. Ramabhadran, "Learning to Speak Fluently in a Foreign Language: Multilingual Speech Synthesis and Cross-Language Voice Cloning," in *Proc. Interspeech*, 2019, pp. 2080–2084.

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

[27] Munich Artificial Intelligence Laboratories GmbH, "The M-AILABS speech dataset," 2019, accessed 30 November 2019. [Online]. Available: https://www.caito.de/2019/01/the-m-ailabs-speech-dataset/

[28] K. Park and T. Mulc, "CSS10: A Collection of Single Speaker Speech Datasets for 10 Languages," in *Proc. Interspeech*, 2019, pp. 1566–1570.

[29] Data Baker China, "Chinese standard mandarin speech corpus," accessed 05 May 2020. [Online]. Available: www.data-baker.com/open_source.html

[30] H. Zen, V. Dang, R. Clark, Y. Zhang, R. J. Weiss, Y. Jia, Z. Chen, and Y. Wu, "LibriTTS: A Corpus Derived from LibriSpeech for Text-to-Speech," in *Proc. Interspeech*, 2019, pp. 1526–1530.

[31] H.-T. Luong, X. Wang, J. Yamagishi, and N. Nishizawa, "Training multi-speaker neural text-to-speech systems using speaker-imbalanced speech corpora," in *Proc. Interspeech*, 2019, pp. 1303–1307.

[32] A. Nagrani, J. S. Chung, W. Xie, and A. Zisserman, "Voxceleb: Large-scale speaker verification in the wild," *Computer Science and Language*, 2019.

[33] 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.

[34] 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.

[35] W.-N. Hsu, Y. Zhang, R. Weiss, H. Zen, Y. Wu, Y. Cao, and Y. Wang, "Hierarchical generative modeling for controllable speech synthesis," in *International Conference on Learning Representations*, 2019.