Low-Resource Speech Synthesis with Speaker-Aware Embedding

Li-Jen Yang[⋆]*, I-Ping Yeh*[†]*, Jen-Tzung Chien*[⋆]

Institute of Electrical and Computer Engineering[⋆], Graduate Degree Program of Cybersecurity[†]
National Yang Ming Chiao Tung University, Taiwan
{lijen0918.ee10,ping629.cs10,jtchien}@nycu.edu.tw

Abstract

Speech synthesis has been successfully exploited for mapping from text sequence to speech waveform where high-resource languages have been well studied and learned from a large amount of text-speech paired data in public-domain corpora. However, developing speech synthesis under low-resource languages is challenging for speech communication in local regions since the collection of training data is expensive. In particular, the speaker-aware speech generation under low-resource settings is crucial in real world. Such a problem is increasingly difficult in case of very limited speaker-specific data. This paper presents a speaker-aware speech synthesis under low-resource settings based on an encoder-decoder framework by using transformer. Knowledge transfer is performed by incorporating a speaker-aware embedding through first learning a pretrained transformer from multi-speaker data of a low-populated spoken language and then fine-tuning the transformer to a target speaker with very limited speaker-specific embeddings. Experiments on low-resource Taiwanese speech synthesis are evaluated to show the merit of speaker-aware transformer in terms of Mel cepstral distortion and mean opinion score.

Index Terms: low-resource speech synthesis, speaker-aware embedding, encoder-decoder model, transformer

1. Introduction

Text-to-speech (TTS) [1, 2] is known as a research topic on sequence mapping which transforms a natural sentence in source domain into a speech utterance in target domain [3] where the technical data in two domains are presented in different styles. TTS has been deeply trained and successfully applied in various domain such as chatbots and intelligent assistants, e.g. Apple's Siri and Amazon's Alexa. The trained TTS is feasible to generate a human-like voice by using a sophisticated deep neural network in case that sufficient voice recordings from a speaker or multiple speakers are available [4]. Such a case is generally possible for high-resource languages since the text-speech paired data from multiple speakers have been abundant and accessible in public domain. Nevertheless, TTS for a target male or female speaker still requires a huge amount of paired data for supervised learning so that all possible acoustic contents are covered and learned. Basically, this is a difficult task when a budget limit for data collection is taken into account.

Recently, the trend of TTS system tends to build a model from multiple speakers [5, 6, 7]. There are several advantages of learning a TTS from multiple speakers instead of a single speaker. First, the learned TTS is able to represent various characteristics from different speakers based their speaker embeddings. This scheme provides the flexibility of synthesizing different speaker voices covering a wide range of acoustic events. Also, given by the well-trained multi-speaker model as an initialization, it is more likely to estimate TTS for an unseen speaker through model fine-tuning. Second, the requirement of voice recordings from an individual speaker can be relaxed. This requirement is significantly reduced when compared with the traditional TTS which usually requires training speech as long as 20 hours. Each speaker in multi-speaker TTS only needs several hours of speech recordings. Total length of multiple speakers is still much longer than that of single speaker. Third, when the transcriptions from different speakers are distinct, the coverage of sound events is accordingly enhanced in multi-speaker TTS. The issue of out-of-vocabulary or unseen acoustic events is mitigated. Following the advantages of multi-speaker TTS, a common way to synthesize the voice for a target speaker is to first construct a pretrained model and then adapt it to a customer voice using the limited data from the customer. A recent example was presented to develop a so-called LibriTTS [8] which was learned as a multi-speaker TTS in English by using the pretrained model learned from LibriSpeech speech corpus consisting of 2456 speakers. In addition, a similar idea was exploited for voice conversion which is another form of domain mapping for voices from a source speaker to a target speaker. Producing a target speaker's voice was implemented by cascading automatic speech recognition (ASR) and TTS [9]. ASR was applied to find text from source speaker's voice and then TTS was used to map this source text to a target speaker's voice [10].

This paper presents a new speaker-adaptive multi-speaker TTS where the challenges of low-resource language and limited adaptation data are tackled. First, a pretrained TTS based on an encoder-decoder framework using transformer is learned from text-speech paired data of multiple speakers of a low-populated language. The styles, accents, ages and genders from different speakers are accommodated in the model. Second, the TTS transformer is fine-tuned for an unseen target speaker by incorporating utterance-based speaker features. The speaking style is then generalized from multi-speaker model to match that of a target speaker for various text transcriptions. This study is investigated by exploring a low-resource TTS in Taiwanese, a local spoken language in Taiwan, where a limited amount of training samples from multiple native speakers are collected. In particular, a speaker-aware voice conversion is implemented and evaluated for Taiwanese TTS where a customer with only a few hours of speech recordings is provided. A number of experiments are assessed to show the effectiveness of this method.

2. Background survey

2.1. Voice conversion

Voice conversion is recognized as a domain mapping for voices which aims to transform the voice of a source speaker into that of a target speaker. A naive way to handle this issue is to use the cascaded model based on ASR and TTS, a baseline system in the Voice Conversion Challenge 2020 (http://www.vc-challenge.org/). However, such a method

979-8-3503-9796-3/22/$31.00

was inefficient since it required two processing components and one fine-tuning component so as to convert the voices into a target speaker. In case of using encoder-decoder framework, there would be two types of encoder, namely speaker encoder and content encoder. These two encoders were used to disentangle the speaker features and content features. These features were then used as the inputs to a TTS decoder to reconstruct the voices of a target speaker. Therefore, if the voices or features of a target speaker could be used as the inputs to the speaker encoder and combined with the content features from source speaker, the converted voices to a new speaker with another speaking style could be generated.

2.2. Text-to-speech transformer

Transformer [11] is formed as an encoder-decoder architecture driven by self attention as well as cross attention which work successfully for sequential learning in a wide range of applications. Self attention is operated within individual input and output sequences in encoder and decoder, respectively, while cross attention is fulfilled to attend output sequence from the attended input sequence. Transformer has achieved state-of-the-art performance in many applications in natural language processing areas. For the application on TTS, the transformer-based TTS [1] was proposed as a variant of transformer where the network architecture for TTS was adjusted from the original transformer designed for machine translation and ASR [12]. The power of TTS transformer was inherited from that of original transformer. This transformer did improve state-of-the-art model based on Tacotron 2 [2]. In addition, the transformer-based TTS has become the mainstream method [13], such as the variants of Fastspeech [14, 15] and Adaspeech [7, 16]. This study presents a speaker-aware voice conversion based on a modern transformer so as to build TTS for a low-populated language.

3. Speaker-Aware Low-Resource TTS

Speech synthesis under low-resource settings is challenging but impacting because the data collection is difficult while the language heritage is crucial. Taiwanese (or specifically Taiwanese Hokkien) is a local spoken language with a variety of dialects where the percentage of home use in Taiwan is 81.9%. Different from Mandarin, English, and the other languages with large population, Taiwanese is viewed as a low-populated language where collection of a large speech corpus is expensive.

3.1. Spoken language processing in Taiwanese

This paper deals with speech synthesis in Taiwanese. The written language of Taiwanese is typically Chinese. The spoken sound in Taiwanese accent can be transcribed by a unique pronunciation based on Taiwanese Language Phonetic Alphabet (TLPA) developed by the Taiwan Language and Literature Society. TLPA is a phonetic symbol system, which mainly consists of Latin letters. Taiwanese is a tonal language where there are nine tones, each of which corresponds to a different Latin character code. However, language data should be represented when stored and operated. The American Standard Code for Information Interchange (ASCII) is a computer encoding system based on Latin alphabets. For computer processing in Taiwanese, this work adopts another encoding system called Taiwanese Language (TL) Pinyin, where tones are denoted by numbers. A comparison of phonetic transcription in Latin letter and TL Pinyin is shown in Table 1 where only seven tones are shown. There are three ways to deal with the other special tones. First, some dialects have a sixth tone, e.g. Latin representation of sixth tone of o is ŏ. Second, some dialects have ninth tones, e.g. Latin representation of ninth tone of o is ő. Third, the symbol "–" is marked between heavy and light sounds. For example, Latin representation of the word "passport" is "hōo tsiàuh", it can be converted to "hoo7-tsiau3" in TL Pinyin.

Table 1: *Comparison of Latin tokens and TL Pinyin tokens.*

Phonetic							
Latin	tong	tóng	tòng	tok	tông	tōng	tǒk
TL	tong1	tong2	tong3	tok4	tong5	tong7	tok8

In implementation of TTS conversion, a text string is basically translated into a phoneme sequence by a text-to-phoneme converter so that the sound synthesis with a single pronunciation can be obtained. For Taiwanese speech synthesis, the first step is to convert the sequence of Chinese characters into a phone sequence in TL Pinyin. The resulting sentence is then converted to TL tokens, e.g. TL sentence "i1 iau2-be7 lai5" (He hasn't arrived yet) can be pronounced by a character string "i 1 ⟨space⟩ i a u 2 ⟨space⟩ b e 7 ⟨space⟩ l a i 5". It is because there doesn't exist a formal rule or mapping table between TL Pinyin and phoneme symbol. Taiwanese TTS can be implemented through 26 English characters to carry out the pronunciation for Taiwanese speech. Such a model is only learned with a pronunciation model over 26 English letters and 9 tones.

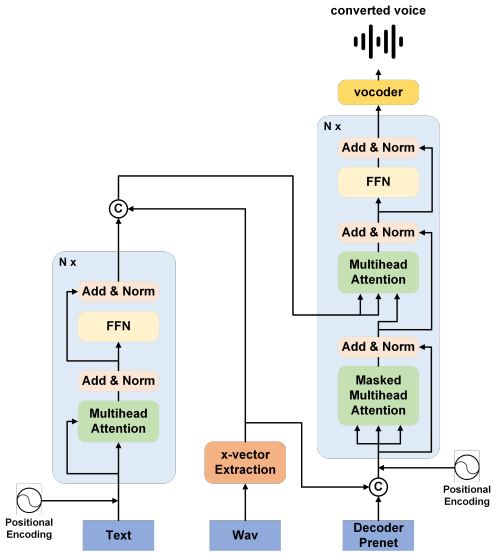

Figure 1: *Speaker-aware low-resource speech synthesis using transformer. Outputs of encoder from text data of multiple speakers are cascaded with x-vectors of a target speaker to synthesize the converted speech for a target speaker.*

3.2. Speaker embedding and awareness

This paper proposes a speaker-aware low-resource speech synthesis where the system overview is illustrated in Figure 1. This system is pretrained and fine-tuned as a speaker adaptive Taiwanese TTS based on a new transformer where the architecture is combined with a speaker module. A multi-speaker training is performed to assure the quality of synthesized speech in presence of low-resource language given with a limited amount of

training utterances from multiple speakers as well as a target speaker. Speaker awareness is implemented by incorporating a speaker embedding into a multi-speaker TTS. This embedding contains different styles of voices. In the implementation, the x-vector [17, 18], which is popular as a single vector to represent speaker characteristics within an utterance, is introduced and cascaded with the output vector of transformer encoder as well as the input vector of transformer encoder. The first cascaded vector is used to perform cross attention between input text and output speech while the second cascaded vector is adopted as the input vector to run the masked multihead attention. The speaker module is jointly trained with TTS module under a specialized transformer. In the preprocessing of x-vectors, an entire speech signal is first sliced and then the spectral features of individual slices are calculated by a deep neural network (DNN) with data augmentation as detailed in [17]. Empirically, the speaker embeddings based on i-vector [19, 20] and d-vector [21] can be also taken into account to carry out a speaker-aware low-resource TTS in Taiwanese.

3.3. Learning for low-resource speech synthesis

There are two stages in the training procedure as illustrated in Algorithm 1 including pretraining stage and fine-tuning stage where both stages need to calculate the spectral feature for input and output strings $\{\mathbf{x}, \mathbf{y}\}$. The Mel-scaled spectrogram is calculated find filter bank features for these two strings. A kind of transfer learning is implemented to build an adaptive multi-speaker transformer. There are two sets of paired sequences of word tokens and speech frames where the total length of the utterances of each speaker in both data sets is limited. Transfer learning is performed to transfer the pretrained transformer, learned from paired data $\{\mathbf{x}_s, \mathbf{y}_s\}$ in presence of multiple speakers in source domain Ω_s, to the fine-tuned transformer, learned from paired data $\{\mathbf{x}_t, \mathbf{y}_t\}$ of a target speaker in target domain Ω_t. A mapping function $f : \Omega_s \rightarrow \Omega_t$ is learned for voice conversion in a speaker-aware low-resource TTS.

In the learning procedure, the embeddings of word or character tokens $\mathbf{x} = \{\mathbf{x}_n\}_{n=1}^{T_x}$ are first transformed to find an intermediate feature representation via an encoder with N layers where each layer consists of a multihead *self attention* network [22] and a feedforward network (FFN). The speaker embedding based on x-vector is added as a speaker aware representation before sending it to the decoder, and the speaker embedding is fixed during the training procedure. Finally, the decoder performs a multihead *cross attention* given with the inputs from speaker aware features as well as the target speech features which are calculated from the spectra of a speech utterance $\mathbf{y} = \{\mathbf{y}_n\}_{n=1}^{T_y}$ via multihead self attention with causal mask. Accordingly, the outputs from N-layer decoder with masked self attention, cross attention and FFN in each layer are used to find the converted voice $\widehat{\mathbf{y}} = \{\widehat{\mathbf{y}}_n\}_{n=1}^{T_y}$ where the vocoder based on the pretrained parallel WaveGAN [23] (a simple and effective parallel waveform generation method based on a generative adversarial network) is employed to produce the speech waveform. A multi-speaker encoder-decoder attention network combined with a target speaker module is trained. Typically, cross attention is adopted to precisely align between the sequences of word tokens $\mathbf{x}$ and speech spectra $\mathbf{y}$. With a reliable alignment, it is likely to assure producing the speech waveform with correct duration. In the fine-tuning stage, the model parameters are first initialized with the pretrained model with weights θ. The number of learning epochs τ_f for fine-tuning a model to a target speaker is reduced when compared with that τ_p in pretraining stage. The training objective here is the same as that in TTS using Tacotron 2 [2]. The objective consists of the mean squared error (MSE) loss (or equivalently ℓ_2 loss) $\ell_{\text{mse}} = \frac{1}{T_y}\sum_{n=1}^{T_y}(\mathbf{y}_n - \widehat{\mathbf{y}}_n)^2$ and the ℓ_1 loss $\ell_1 = \frac{1}{T_y}\sum_{n=1}^{T_y}|\mathbf{y}_n - \widehat{\mathbf{y}}_n|$ where $\widehat{\mathbf{y}}_n$ denotes the predicted spectrogram and $\mathbf{y}_n$ denotes the ground-truth spectrogram. Furthermore, the decoder needs to predict the token of stopping in a token sequence. If the stop token is true, the model plans to stop decoding. There is a binary cross entropy (BCE) loss is measured by $\ell_{\text{bce}} = \text{y}_{\text{stop}} \cdot \log \text{y}_{\text{pred}} + (1 - \text{y}_{\text{stop}}) \cdot \log(1 - \text{y}_{\text{pred}})$ where y_{stop} denotes the label of stop token, and y_{pred} denotes the prediction of stop token. The training objective is the combination of ℓ_1 loss, MSE loss and BCE loss as $\mathcal{L} \triangleq \ell_{\text{mse}} + \ell_1 + \ell_{\text{bce}}$. Speaker-aware transformer parameter ϕ is first pretrained and then fine-tuned.

Algorithm 1 Pretraining and fine-tuning for low-resource speech synthesis

Input: source data $\{\mathbf{x}_s, \mathbf{y}_s\}$, target data $\{\mathbf{x}_t, \mathbf{y}_t\}$, hyperparameters τ_p, τ_f
Output: Fine-tuned model parameter ϕ
Pretraining stage $(\mathbf{x}_s, \mathbf{y}_s, \tau_p)$
initialize model parameter θ
while $epoch < \tau_p$ **do**
  model generates $\widehat{\mathbf{y}}_s$ from $\mathbf{x}_s$
  compute loss $\mathcal{L}$ with $\widehat{\mathbf{y}}_s$ and $\mathbf{y}_s$
  update the model parameter θ
Fine-tuning stage $(\mathbf{x}_t, \mathbf{y}_t, \theta, \tau_f)$
initialize model parameter: $\phi \leftarrow \theta$
while $epoch < \tau_f$ **do**
  model generates $\widehat{\mathbf{y}}_t$ from $\mathbf{x}_t$
  compute loss $\mathcal{L}$ with $\widehat{\mathbf{y}}_t$ and $\mathbf{y}_t$
  update the model parameter ϕ
return ϕ

4. Experiments

4.1. Experimental setups

This study conducted the experiments on the Taiwanese across Taiwan (TAT) corpus [24] and the Suisiann dataset (`https://suisiann-dataset.ithuan.tw/`). TAT provided 100 hours of speech utterances from 100 speakers which contained different genders and accents. Suisiann (means the beautiful sound) is a public dataset which included roughly 4 hours of Taiwanese speech with only one female speaker. The sampling rate of speech waveform was adjusted to 16KHz for both datasets which was fitted to the setting of Kaldi [25] for feature extraction when using the ESPnet [26]. Furthermore, Kaldi library was used to trim the silence segments at the beginning and end of an utterance. This scheme would refrain the model to be affected when aligning between text sentence and speech waveform. For text processing, the individual TL tokens with numbers were seen as the characters which were encoded as the inputs to the learned model.

In the implementation, the transformer was trained via ESPnet with six transformer blocks in both encoder and decoder. Number of attention heads was set as 4 and the size of hidden features was 384. The speaker embedding using x-vector with a dimension of 512 was calculated. A personal computer, equipped by Quadro RTX 6000 GPU, was used to carry out the proposed method where 20 hours of training utterances were

utilized. The hyperparameters for the number of epochs in running the pretrained and fine-tuned models were set as 200 and 30, respectively. The length of training time was roughly 30 minutes. The Adam optimizer was used with parameters $\beta_1 = 0.9$, $\beta_2 = 0.98$, $\epsilon = 10^{-9}$ which were defined in [27]. In the inference time, the pretrained parallel Wave-GAN [23] was used as the vocoder to synthesize speech waveform.

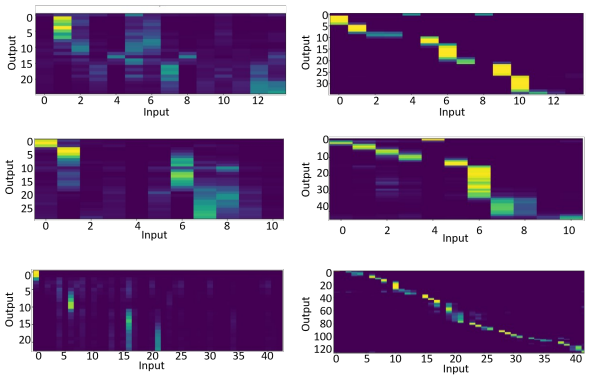

Figure 2: *Comparison of cross-attention maps for ablation study. Left-hand-side sub-figures are the results of three Taiwanese utterances from the direct-trained TTS without speaker awareness, and the right-hand-side sub-figures are the corresponding results from speaker-aware TTS.*

4.2. Experimental results

4.2.1. Ablation study on speaker-aware TTS

First, the ablation study on low-resource TTS with and without speaker-aware knowledge transfer is conducted. The comparison is made by showing the cross-attention maps between encoder input and decoder output of a transformer. Figure 2 shows the attention maps of direct-trained TTS without speaker awareness (left-hand-side) and speaker-aware TTS (right-hand-side) in presence of three different Taiwanese utterances. The alignment result between input and output sequences based on the attention weights reveals that the speaker-aware embedding in transformer does improve the prediction of duration information of the synthesized voices in sequence mapping.

Table 2: *Comparison of MCD and RMSLE of synthesized speech under various data amounts and speaker embeddings.*

Fine-tuning data amount	MCD	RMSLE
100 utters with i-vector	9.52 ± 0.91	0.54 ± 0.27
100 utters with x-vector	9.65 ± 1.02	0.51 ± 0.28
300 utters with i-vector	8.68 ± 1.36	0.33 ± 0.22
300 utters with x-vector	**8.59 ± 0.72**	0.33 ± 0.16
500 utters with i-vector	8.76 ± 1.16	**0.28 ± 0.20**
500 utters with x-vector	8.76 ± 1.26	0.3 ± 0.15

4.2.2. Evaluation on data amount and speaker embedding

In objective evaluation, the Mel cepstral distortion (MCD) [28] and the root mean squared logarithmic error (RMSLE) (`https://www.kaggle.com/code/marknagelberg/rmsle-function/script`) of synthesized speech are measured. For both measures, the lower the better. The effects of data amount and speaker embedding are evaluated for low-resource Taiwanese TTS. The number of fine-tuning utterances from a target speaker is varied from 100 to 300 and 500 utterances where the length of total utterances is 3.67, 12.06 and 22.56 minutes, respectively. On the other hand, the speaker-aware embedding for low-resource TTS using i-vector [19] and x-vector [17] is investigated by comparing MCD and RMSLE of the corresponding synthesized speech. As shown in Table 2, the lowest MCD is obtained by using 300 adaptation utterances where x-vectors are applied while the lowest RMSLE is achieved by using 500 utterances with i-vectors. The results with one standard deviation are shown. Basically, MCD is a popular measure in speech synthesis. In terms of MCD, the length of adaptation utterances with 12.06 minutes is sufficient. The results of i-vector and x-vector are comparable in this comparison.

Table 3: *Comparison of MCD and MOS of synthesized speech without and with speaker awareness for fine-tuned model.*

Method	MCD	MOS
ground-truth speech	N/A	4.90 ± 0.09
TTS w/o speaker awareness	12.62 ± 1.63	1.62 ± 0.34
speaker-aware TTS	8.59 ± 0.72	3.32 ± 0.45

4.2.3. Evaluation on human judgement

Table 3 further shows the ablation study on comparing MCD of Taiwanese TTS without and with speaker awareness fine fine-tuned model. This comparison reveals significant improvement by fine-tuning the model with speaker-aware embedding. The improvement is still clear even with adaptation utterances as limited as 3.67 minutes. In addition, the subjective evaluation of synthesized speech in terms of mean opinion score (MOS) with 95% confidence with score between 1 and 5 is displayed in the comparison. There are five persons evolved in human evaluation. Ten test utterances are evaluated. MOS of ground-truth speech is included. As shown in Table 3, the result of MOS is substantially improved by introducing the proposed speaker-aware TTS in transformer. This paper also provided the synthesized speech data for listening test [1].

5. Conclusions

This paper presented a speaker-aware approach to low-resource Taiwanese text-to-speech. A new encoder-decoder framework based on transformer was developed. A voice conversion scheme in an adaptive TTS was incorporated by combining the encoder outputs from multi-speaker voices and the speaker-specific embeddings from low-resource enrollment utterances. This combination passed through a decoder to generate the converted voice with a vocoder. The experimental results showed that the speaker-aware knowledge was merged to substantially enhance the voice quality. Speaker-adaptive speech synthesis was implemented to achieve the desirable performance with as limited as 12.06 minutes. Future studies could be extended to further fine-tune the results with the gender information and additionally control the prosody of synthesizer with the features of pitch, energy and duration [29].

[1] Synthesized speech samples are provided at `https://nycu-mllab.github.io/Low-Resource_Speech_Synthesis_with_Speaker-Aware_Embedding/`

6. References

[1] N. Li, S. Liu, Y. Liu, S. Zhao, and M. Liu, "Neural speech synthesis with transformer network," in *Proc. of AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 6706–6713.

[2] 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 *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing*, 2018, pp. 4779–4783.

[3] L. Li, M.-W. Mak, and J.-T. Chien, "Contrastive adversarial domain adaptation networks for speaker recognition," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 5, pp. 2236–2245, 2022.

[4] D. Yu, G. Hinton, N. Morgan, J.-T. Chien, and S. Sagayama, "Introduction to the special section on deep learning for speech and language processing," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 4–6, 2011.

[5] M. Chen, X. Tan, Y. Ren, J. Xu, H. Sun, S. Zhao, T. Qin, and T.-Y. Liu, "Multispeech: Multi-speaker text to speech with transformer," *arXiv preprint arXiv:2006.04664*, 2020.

[6] J. Kim, J. Kong, and J. Son, "Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech," in *Proc. of International Conference on Machine Learning*, 2021, pp. 5530–5540.

[7] M. Chen, X. Tan, B. Li, Y. Liu, T. Qin, S. Zhao, and T.-Y. Liu, "AdaSpeech: Adaptive text to speech for custom voice," in *Proc. of International Conference on Learning Representations*, 2020.

[8] 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," *Proc. Annual Conference of International Speech Communication Association*, pp. 1526–1530, 2019.

[9] Y. Liao, W. Hsu, C. Pan, W. Wang, M. Pleva, and D. Hladek, "Personalized Taiwanese speech synthesis using cascaded ASR and TTS framework," in *Proc. of International Conference Radioelektronika*, 2022, pp. 01–05.

[10] M.-W. Mak and J.-T. Chien, *Machine Learning for Speaker Recognition*. Cambridge University Press, 2020.

[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[12] C.-H. Leong, Y.-H. Huang, and J.-T. Chien, "Online compressive transformer for end-to-end speech recognition." in *Proc. of Annual Conference of International Speech Communication Association*, 2021, pp. 2082–2086.

[13] H. Lio, S.-E. Li, and J.-T. Chien, "Adversarial mask transformer for sequential learning," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing*, 2022, pp. 4178–4182.

[14] Y. Ren, Y. Ruan, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, "FastSpeech: Fast, robust and controllable text to speech," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[15] Y. Ren, C. Hu, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, "FastSpeech 2: Fast and high-quality end-to-end text to speech," in *Proc. of International Conference on Learning Representations*, 2020.

[16] Y. Yan, X. Tan, B. Li, T. Qin, S. Zhao, Y. Shen, and T.-Y. Liu, "Adaspeech 2: Adaptive text to speech with untranscribed data," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing*, 2021, pp. 6613–6617.

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

[18] J.-T. Chien and S. Luo, "Self-supervised learning for online speaker diarization," in *Proc. of Asia-Pacific Signal and Information Processing Association Annual Summit and Conference*, 2021, pp. 2036–2042.

[19] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2010.

[20] W.-W. Lin, M.-W. Mak, and J.-T. Chien, "Multisource i-vectors domain adaptation using maximum mean discrepancy based autoencoders," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 12, pp. 2412–2422, 2018.

[21] E. Variani, X. Lei, E. McDermott, I. L. Moreno, and J. Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing*, 2014, pp. 4052–4056.

[22] J.-T. Chien and C.-W. Wang, "Hierarchical and self-attended sequence autoencoder," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 9, pp. 4975–4986, 2022.

[23] 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 *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing*, 2020, pp. 6199–6203.

[24] Y.-F. Liao, C.-Y. Chang, H.-K. Tiun, H.-L. Su, H.-L. Khoo, J. S. Tsay, L.-K. Tan, P. Kang, T.-g. Thiann, U.-G. Iunn *et al.*, "Formosa speech recognition challenge 2020 and Taiwanese across Taiwan corpus," in *Prof. of Conference of International Committee for the Co-ordination and Standardisation of Speech Databases and Assessment Techniques*, 2020, pp. 65–70.

[25] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The Kaldi speech recognition toolkit," in *Proc. of IEEE Workshop on Automatic Speech Recognition and Understanding*, 2011.

[26] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N.-E. Y. Soplin, J. Heymann, M. Wiesner, N. Chen *et al.*, "ESPnet: End-to-end speech processing toolkit," in *Proc. of Annual Conference of International Speech Communication Association*, 2018, pp. 2207–2211.

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

[28] R. Kubichek, "Mel-cepstral distance measure for objective speech quality assessment," in *Proc. of IEEE Pacific Rim Conference on Communications Computers and Signal Processing*, vol. 1, 1993, pp. 125–128.

[29] H.-S. Choi, J. Lee, W. Kim, J. Lee, H. Heo, and K. Lee, "Neural analysis and synthesis: Reconstructing speech from self-supervised representations," *Advances in Neural Information Processing Systems*, vol. 34, pp. 16 251–16 265, 2021.