INTERSPEECH 2020
October 25–29, 2020, Shanghai, China

MultiSpeech: Multi-Speaker Text to Speech with Transformer

Mingjian Chen[1], *Xu Tan*[2], *Yi Ren*[3], *Jin Xu*[4], *Hao Sun*[1], *Sheng Zhao*[5], *Tao Qin*[2]

[1]School of Software and Microelectronics, Peking University
[2]Microsoft Research Asia,
[3]Zhejiang Univeristy,
[4]Tsinghua University,
[5]Microsoft Azure Speech

milk@pku.edu.cn, xuta@microsoft.com, rayeren@zju.edu.cn, j-xu18@mails.tsinghua.edu.cn, sigmeta@pku.edu.cn, Sheng.Zhao@microsoft.com, taoqin@microsoft.com

Abstract

Transformer-based text to speech (TTS) model (e.g., Transformer TTS [1], FastSpeech [2]) has shown the advantages of training and inference efficiency over RNN-based model (e.g., Tacotron [3]) due to its parallel computation in training and/or inference. However, the parallel computation increases the difficulty while learning the alignment between text and speech in Transformer, which is further magnified in the multi-speaker scenario with noisy data and diverse speakers, and hinders the applicability of Transformer for multi-speaker TTS. In this paper, we develop a robust and high-quality multi-speaker Transformer TTS system called MultiSpeech, with several specially designed components/techniques to improve text-to-speech alignment: 1) a diagonal constraint on the weight matrix of encoder-decoder attention in both training and inference; 2) layer normalization on phoneme embedding in encoder to better preserve position information; 3) a bottleneck in decoder pre-net to prevent copy between consecutive speech frames. Experiments on VCTK and LibriTTS multi-speaker datasets demonstrate the effectiveness of MultiSpeech: 1) it synthesizes more robust and better quality multi-speaker voice than naive Transformer based TTS; 2) with a MutiSpeech model as the teacher, we obtain a strong multi-speaker FastSpeech model with almost zero quality degradation while enjoying extremely fast inference speed.

Index Terms: text to speech, multi-speaker, Transformer, FastSpeech, attention alignment

1. Introduction

In recent years, neural text to speech (TTS) models such as Tacotron [4, 3], Transformer TTS [1] and FastSpeech [2, 5] have led to high-quality single-speaker TTS systems using large amount of clean training data. Thanks to the parallel computation in Transformer [6], Transformer based TTS enjoys much better training [1, 2] and inference [2] efficiency than RNN based TTS [4, 3].

To reduce deployment and serving cost in commercial applications, building a TTS system supporting multiple (hundreds or thousands) speakers has attracted much attention in both industry and academia [7, 8, 3, 9]. While it is affordable to record high-quality and clean voice in professional studios for a single speaker, it is costly to do so for hundreds or thousands of speakers to build a multi-speaker TTS system. Thus, multi-speaker TTS systems are usually built using multi-speaker data recorded for automatic speech recognition (ASR) [10, 11] or voice conversion [12], which is noisy and of low-quality due to the diversity and variances of prosodies, speaker accents, speeds and recording environments. Although Transformer based models have shown advantages over other neural models for single-speaker TTS, existing works on multi-speaker TTS mostly adopt RNN (e.g., Tacotron [4, 3]) or CNN (e.g., Deep Voice [7, 8]) as the model backbone, and few attempts have been made to build Transformer based multi-speaker TTS.

The main challenge of Transformer multi-speaker TTS comes from the difficulty of learning the text-to-speech alignment, while alignment plays an important role in TTS modeling [3, 8, 2]. While applying Transformer to multi-speaker TTS, the text-to-speech alignment between the encoder and decoder is more difficult than that of RNN models. When calculating the attention weights in RNN, location-sensitive attention [13] are leveraged to ensure the attention move forward consistently through the input, avoiding word skipping and repeating problems. Location-sensitive attention leverages attention results in previous decoder time steps, which, unfortunately, cannot be used in Transformer due to parallel computation during training. In single-speaker TTS, the text and speech data are usually of high-quality and the text-to-speech alignments are easy to learn. However, as aforementioned, the speech data for multi-speaker TTS is usually noisy, which makes the alignments much more difficult. Actually, CNN multi-speaker TTS also faces this challenge, and complex systems are designed based on the characteristics of CNN structure in [7, 8], which unfortunately cannot be easily applied on Transformer models.

In order to bring the advantages of Transformer into multi-speaker TTS modeling, in this paper, we develop a robust and high-quality multi-speaker TTS system called MultiSpeech. Specifically, we introduce several techniques to improve the alignments based on empirical observations and insights. First, considering the attention alignments between the text encoder and speech decoder are usually monotonic and diagonal, we introduce a diagonal constraint on the weight matrix of the encoder-decoder attention during training and inference. Second, position embeddings are important in Transformer and can help text-to-speech alignment [8]. However, the scale of phoneme embeddings can vary a lot, which causes magnitude mismatch[1] while added together and consequently increases the difficulty of model training. Therefore, we add layer normalization on phoneme embeddings to make them more comparable. Third, text-to-speech alignments should be learnt by

[1]The embeddings of some phonemes are large and will dominate position embeddings, and some phonemes are of small embeddings and will be dominated by position embeddings, both of which will harm the alignment learning.

http://dx.doi.org/10.21437/Interspeech.2020-3139

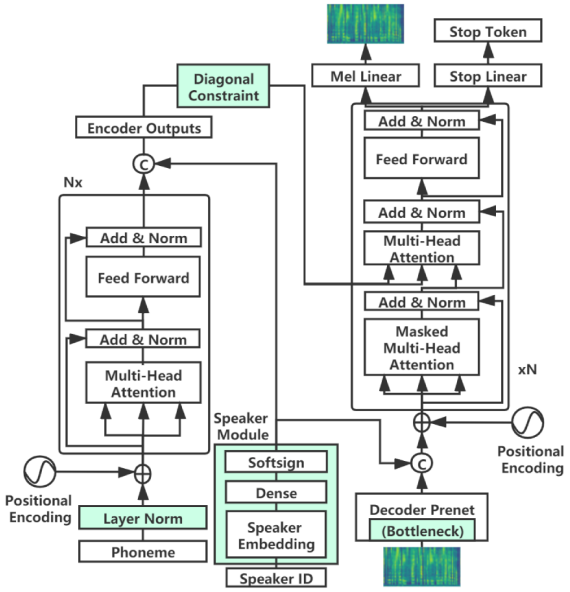

Figure 1: *The model structure of our proposed MultiSpeech. The green blocks are the newly added modules for multi-speaker TTS based on Transformer.*

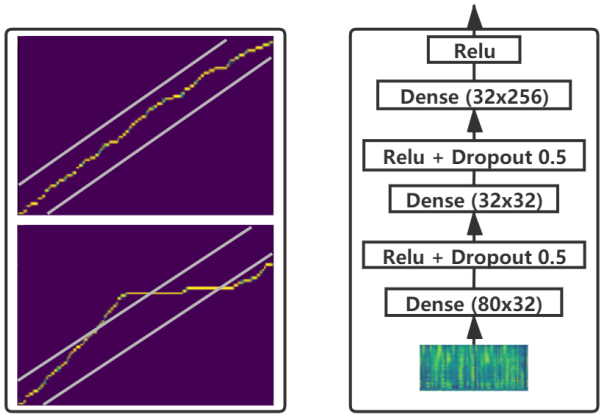

Figure 2: *(a) The illustration of diagonal constraint in attention, where the above figure has a small diagonal constraint loss and the below figure has a large diagonal constraint loss. (b) The model structure of the pre-net bottleneck in decoder.*

attending to source phonemes while generating target speech frames. However, two adjacent speech frames are usually similar and standard Transformer decoder tends to directly copy previous frame to generate the current frame. Consequently, no alignments between text and speech can be learned. To prevent direct copy between consecutive speech frames, we employ a bottleneck structure in the decoder pre-net which encourages the decoder to generalize on the representation of speech frame instead of memorization, and forces the decoder to attend to text/phoneme inputs.

Experiments on VCTK and LibriTTS multi-speaker datasets show that 1) MultiSpeech achieves great improvements (1.01 MOS gain on VCTK and 1.46 MOS gain on LibriTTS) over naive Transformer based TTS and synthesizes robust and high-quality multi-speaker voice. 2) The three proposed techniques can indeed improve text-to-speech alignments, measured by the attention diagonal rate. 3) MultiSpeech model can be used as a teacher for FastSpeech and obtain a strong multi-speaker FastSpeech model without quality degradation but enjoying extremely fast inference.

2. Background

Transformer TTS. Transformer based TTS (e.g. [1]) adopts the basic model structure of Transformer [6], as shown in Figure 1 (remove the green blocks). Each transformer block consists of a multi-head self-attention and a feed-forward network. Additionally, a decoder pre-net is leveraged to pre-process the mel-spectrogram frame, a mel linear layer is used to predict the mel-spectrogram frame and a stop linear layer to predict if should stop in each time step. Transformer can ensure parallel computation during training, which, as a side effect, harms the attention alignments between text and speech. As a result, it is challenging to build multi-speaker TTS on Transformer considering the complicated acoustic conditions in multi-speaker speech. In this paper, we analyze each component in Transformer TTS to figure out why it fails to learn alignments, and propose the corresponding modifications to improve the alignments.

Multi-Speaker TTS. Several works have built multi-speaker text to speech systems based on RNN [14, 15] and CNN [7, 8]. RNN-based multi-speaker model enjoys the benefits of recurrent attention computation as in Tacotron 2 [3], which can leverage the attention information in previous steps to help the attention calculation in current step. CNN-based multi-speaker model [8] develops many sophisticated mechanisms in the speaker embedding and attention block to ensure the synthesized quality. VAE-based method [9] is further leveraged to handle noisy multi-speaker speech data [11]. Considering the advantages of Transformer including parallel training over RNN and effective sequence modeling over CNN, in this paper, we build multi-speaker TTS on Transformer model.

Text-to-Speech Alignment. Since text and speech correspond to each other in TTS, the alignments are generally monotonic and diagonal in the encoder-decoder attention weights. Previous works have tried different techniques to ensure the alignments in encoder-decoder model. Location sensitive attention [13] is proposed to align the source and target better by leveraging previous attention information. [16, 17, 18, 19] improve text-to-speech alignments by designing sophisticated techniques on attention. [8] design position encoding with its angular frequency determined dynamically by each speaker embedding to ensure the text-speech alignment. [4] uses large dropout in decoder pre-net and finds it is helpful for attention alignment. In this paper, we introduce several techniques to improve the alignments specifically in Transformer model.

3. Improving Text-to-Speech Alignment

In this section, we introduce several techniques to improve the text-to-speech alignments in MultiSpeech, from the attention, encoder and decoder part respectively, as shown in Figure 1.

3.1. Diagonal Constraint in Attention

Monotonic and diagonal alignments in the attention weights between text and speech are critical to ensure the quality of synthesized speech [13, 16, 17, 8, 18, 19]. In multi-speaker scenario, the speech is usually noisy and different speakers have different speeds and acoustic conditions, making the alignments difficult.

Therefore, we propose to add diagonal constraint on the attention weights to force the model to learn correct alignments.

We first formulate the diagonal attention rate r as

$$r = \frac{\sum_{t=1}^{T} \sum_{s=kt-b}^{kt+b} A_{t,s}}{S}, \quad (1)$$

where S is the length of speech mel-spectrogram and T is the length of text (phoneme or character). $k = \frac{S}{T}$ is the slop for each training sample and b is a hyperparameter for bandwidth, both of which determine the shape the diagonal area. $A_{t,s}$ is the t-th row and s-th column of the attention weight matrix A. The numerator represents how much weight lie in the diagonal area while the denominator represents the total attention weight which equals to speech length S. The diagonal constraint loss L_{DC} encourages larger attention weights in the diagonal area as shown in Figure 2, which is defined as $L_{DC} = -r$, where r is defined in Equation 1. L_{DC} is added on the original TTS loss with a weight λ to adjust the strength of the constraint.

In order to ensure the correct alignment during inference, we also add attention constrain in the autoregressvie generation process. We introduce an attention sliding window in the text side and compute the attention weights only within this window. The range of the window is [-1, 4], where 0 in the window represents the window center and is initialized as position 0 in the beginning. The window allows the predicted frame to attend on both previous 1 phoneme and future 4 phonemes of the center. We design a sliding window moving strategy: we define the attention centroid of s-th predicted frame as $C_s = \lfloor \sum_{t=0}^{T} (A_{t,s} * t) \rfloor$. If C_s deviates the window center beyond 3 consecutive frames, we move the sliding window center one step forward.

Compared with the attention constraint strategy proposed in [8], our method has the following advantages: 1) our sliding window allows to attend to the previous position, and 2) we use attention centroid rather than simply the position of the highest attention weight within the current window as new sliding window center. These improvements can prevent the sliding window from moving forward too early, which usually results in skipping phonemes and fast speaking speed.

3.2. Position Information in Encoder

The encoder of Transformer based TTS model usually takes $x + p$ as input, where x is the embedding of phoneme/character token and p is positional embedding to give the Transformer model a sense of token order. p is usually formulated as triangle positional embeddings [6] and the scale of its value is fixed into $[-1, 1]$. However, the embedding x is learned end-to-end, and the scale of the its value can be very large or small. As a result, the position information p in $x + p$ is relatively small or large, which will affect the alignment learning between the source (text) and target (speech) sequence.

To preserve the position information properly in $x + p$, we first add layer normalization [20] on x and then add with p, i.e., $LN(x) + p$, as shown in Figure 1. $LN(x)$ is defined as

$$LN(x) = \gamma \frac{x - \mu}{\sigma} + \beta, \quad (2)$$

where μ and σ are the mean and variance of vector x, γ and β are the scale and bias parameters. In this case, the scale of phoneme embedding x can be restricted to a limited range by learning the scale and bias parameters in layer normalization.

In Transformer TTS [1], a scalar trainable weight α is leveraged to adjust p before adding on x, i.e., $x + \alpha p$. However, it cannot necessarily ensure enough position information in $x + \alpha p$, since a single scalar α cannot balance the scales between position information p and embedding x, considering different phonemes/characters have different scales[2]. We also verify the advantage of our layer normalization over simple scalar trainable weight in the experiment part[3].

3.3. Pre-Net Bottleneck in Decoder

The adjacent frames of mel-spectrogram are usually very similar since the hop size is usually much smaller than the window size[4], which means two adjacent frames have large information overlap. As a consequence, when predicting next frame given current frame as input in autoregressive training, the model is prone to directly copy some information from the input frame instead of extracting information from text side for meaningful prediction. The decoder pre-net in [4] leverages a structure like 80-256-128 where each number represents the hidden size of each layer in the pre-net, while the decoder pre-net in [3, 1] leverages a structure like 80-256-256-512, both with dropout rate of 0.5. The authors [4, 3] claim this structure can act like a bottleneck to prevent from copy (the hidden size is halved in the bottleneck, e.g., 128 vs.256 or 256 vs. 512). However, the mel-spectrogram with a dimension of 80 is first converted into 512 or 256 hidden and is then halved to 256 or 128, which is still larger than 80 and cannot necessarily prevent copy and learn alignments in multi-speaker scenario, according to our experiments. As shown in Figure 2, we further reduce the bottleneck hidden size to as small as $1/8$ of the original hidden size (e.g., 32 vs. the original hidden size 256) plus with 0.5 dropout ratio, and the structure becomes 80-32-32-256. We found this small bottleneck size is essential to learn meaningful alignments and avoid direct copying input frame.

4. Experiments and Results

In this section, we conduct experiments to verify the advantages of MultiSpeech and the effectiveness of the proposed techniques to improve text-to-speech alignments.

4.1. Experimental Setup

Datasets. We conducted experiments on the VCTK [12] and LibriTTS [11] multi-speaker datasets. The VCTK dataset contains 44 hours speech with 108 speakers, while the LibriTTS dataset contains 586 hours speech with 2456 speakers. We convert the speech sampling rate of both corpus to 16KHz, and use 12.5ms hop size, 50ms window size to extract mel-spectrogram. We convert text into phoneme using grapheme-to-phoneme conversion [21] and take phoneme as the encoder input.

Model Configuration. The model structure of MultiSpeech is shown in Figure 1. Both the encoder and decoder use 4-layer transformer blocks. The hidden size, attention head, feed-forward filter size and kernel size are 256, 2, 1024 and 9 respectively. In addition, the decoder pre-net bottleneck, as shown in Figure 2, is 32, which is 1/8 of the hidden size. For the speaker module as shown in Figure 1, we follow the structure in [8].

Training and Inference. We use 4 P100 GPUs, each with

[2]We do not normalize the input in decoder, since mel-specotrgram is not learnable and usually normalized into a fixed range. This point is also confirmed in [1], where the scalar trainable weight in decoder is much more stable and closer to 1 than that in encoder.

[3]Our layer normalization is also better than learnable position embeddings since it still learns a global embedding for each position.

[4]The typical parameters of window size and hop size in TTS is 50ms and 12.5ms.

batch size of about 20,000 speech frames. We use Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.98$, $\epsilon = 10^{-9}$ and follow the learning rate schedule in [6]. The bandwidth b in the attention constraint is set to 50, and the weight λ of L_{DC} is set to 0.01 according to the valid performance. During inference, we use attention constraint as described in Section 3.1 to ensure the text-to-speech alignments. WaveNet [22] is used as vocoder.

Evaluation. We use MOS (mean opinion score) to measure the voice quality. Each sentence is judged by 20 native speakers. We also use the diagonal attention rate r to measure the quality of text-to-speech alignments. A higher MOS means better voice quality while a higher r means better alignments. For both VCTK and LibriTTS, we select 6 speakers (3 men and 3 women, each with 5 sentences) for evaluation respectively.

4.2. The Quality of MultiSpeech

The MOS results are shown in Table 1. We compare our proposed *MultiSpeech* with 1) *GT*, the ground-truth recording, 2) *GT mel + Vocoder*, convert the recording into mel-spectrogram and then convert the mel-spectrogram to audio with Vocoder, and 3) *Transformer based TTS*, only add the speaker embedding module on naive Transformer based TTS model to support multiple speakers, without using any of our techniques. It can be seen that MultiSpeech achieves large MOS score improvements over Transformer based TTS. The MOS score of MultiSpeech on VCTK is close to *GT mel + Vocoder*. These results demonstrate the advantages of MultiSpeech for multi-speaker TTS. We show some demo audios and case analyses in this link[5].

Table 1: *The MOS scores with* 95% *confidence intervals on VCTK and LibriTTS.*

Setting	VCTK	LibriTTS
GT	4.04 ± 0.14	4.14 ± 0.16
GT mel + Vocoder	3.89 ± 0.20	3.90 ± 0.08
Transformer based TTS	2.64 ± 0.35	1.49 ± 0.09
MultiSpeech	3.65 ± 0.14	2.95 ± 0.14

4.3. Method Analysis

Ablation Study. We first conduct ablation study on VCTK dataset to verify the effectiveness of each technique: diagonal constraint (DC) in attention, layer normalization (LN) in encoder, pre-net bottleneck (PB) in decoder. The results are shown in Table 2. After removing diagonal constraint (DC), layer normalization (LN) and pre-net bottleneck (PB) respectively, both MOS score and diagonal rate r drop. After further removing all the three techniques (-DC-LN-PB), both MOS and r drop largely. These ablation studies verify the effectiveness of the three techniques.

Comparison between layer normalization and learnable weight. We calculate the similarity between p and three settings: 1) $LN(x)+p$, our proposed layer normalization (*LN*); 2) $x+\alpha p$, the learnable weight (*LW*) used in [1]; 3) $x+p$, the naive Transformer baseline (*Baseline*). As shown in Table 3, the similarity of *LN* is in between *LW* and *Baseline*, which shows the position information in *LN* is neither too weak (as in *Baseline*) nor too strong (as in *LW*[6]) and is helpful for attention alignment. This is also verified by the diagonal attention rate r in

[5]https://speechresearch.github.io/multispeech/

[6]We check the final learnable weight $\alpha = 2.62$, which is much bigger than the single-speaker setting in [1] (α is about 0.5). We guess that when added with different scales of phoneme embedding, *LW* simply learns a global large α to highlight position embedding.

Table 2: *The MOS with* 95% *confidence intervals and diagonal attention rate* r *of the ablation study on VCTK.* $-$*DC means not using diagonal constraint during training and inference.* $-$*LN means using* $x + p$ *as encoder input but not* $LN(x) + p$. $-$*PB means using pre-net structure like 80-256-256-256 instead of our proposed 80-32-32-256.*

Setting	MOS	r
MultiSpeech	3.65 ± 0.14	0.694
$-$*DC*	3.59 ± 0.25	0.502
$-$*LN*	3.08 ± 0.05	0.637
$-$*PB*	3.36 ± 0.27	0.658
$-$*DC*$-$*LN*$-$*PB*	2.64 ± 0.35	0.366

Table 3. Our proposed *LN* achieves the highest r while *LW* the lowest, which demonstrates that too strong position dominates phoneme embedding and harms the attention alignment.

Table 3: *The comparison of similarity and diagonal attention rate* r *between LN, LW and Baseline settings.*

Setting	*LN*	*LW*	*Baseline*
Similarity	0.126	0.184	0.089
r	0.694	0.506	0.637

4.4. Extension on FastSpeech

We further use MultiSpeech as a teacher to teach a multi-speaker FastSpeech [2] on VCTK dataset, following the setting in [2]. We select 6 speakers (3 men and 3 women, each with 10 sentences) for MOS evaluation. As shown in Table 4, we can obtain a strong FastSpeech model with nearly the same MOS score with MultiSpeech teacher.

Table 4: *The MOS score of multi-speaker FastSpeech on VCTK with* 95% *confidence intervals.*

Setting	*GT*	*MultiSpeech*	*FastSpeech*
MOS	4.02 ± 0.09	3.53 ± 0.22	3.45 ± 0.13

5. Conclusions

In this paper, we developed MultiSpeech, a multi-speaker Transformer TTS system that leverages diagonal constraint, layer normalization and pre-net bottleneck to improve the text-to-speech alignments in multi-speaker scenario. Experiments on VCTK and LibriTTS multi-speaker datasets demonstrate effectiveness of MutiSpeech: 1) it generates much higher-quality and more stable voice compared with Transformer TTS baseline; 2) using MultiSpeech as a teacher, we obtain a strong multi-speaker FastSpeech model to enjoy extremely fast inference speed. In the future, we will continue to improve the voice quality of MultiSpeech and multi-speaker FastSpeech model.

6. References

[1] 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.

[2] Y. Ren, Y. Ruan, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, "Fastspeech: Fast, robust and controllable text to speech," in *Advances in Neural Information Processing Systems*, 2019, pp. 3165–3174.

[3] 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 *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4779–4783.

[4] Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio *et al.*, "Tacotron: Towards end-to-end speech synthesis," *arXiv preprint arXiv:1703.10135*, 2017.

[5] 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," *arXiv*, pp. arXiv–2006, 2020.

[6] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[7] A. Gibiansky, S. Arik, G. Diamos, J. Miller, K. Peng, W. Ping, J. Raiman, and Y. Zhou, "Deep voice 2: Multi-speaker neural text-to-speech," in *Advances in neural information processing systems*, 2017, pp. 2962–2970.

[8] W. Ping, K. Peng, A. Gibiansky, S. O. Arik, A. Kannan, S. Narang, J. Raiman, and J. Miller, "Deep voice 3: Scaling text-to-speech with convolutional sequence learning," *arXiv preprint arXiv:1710.07654*, 2017.

[9] W.-N. Hsu, Y. Zhang, R. J. Weiss, H. Zen, Y. Wu, Y. Wang, Y. Cao, Y. Jia, Z. Chen, J. Shen *et al.*, "Hierarchical generative modeling for controllable speech synthesis," *arXiv preprint arXiv:1810.07217*, 2018.

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

[11] 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," *arXiv preprint arXiv:1904.02882*, 2019.

[12] C. Veaux, J. Yamagishi, K. MacDonald *et al.*, "Superseded-cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit," 2016.

[13] 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.

[14] 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.

[15] R. Skerry-Ryan, E. Battenberg, Y. Xiao, Y. Wang, D. Stanton, J. Shor, R. J. Weiss, R. Clark, and R. A. Saurous, "Towards end-to-end prosody transfer for expressive speech synthesis with tacotron," *arXiv preprint arXiv:1803.09047*, 2018.

[16] M. He, Y. Deng, and L. He, "Robust sequence-to-sequence acoustic modeling with stepwise monotonic attention for neural tts," *arXiv preprint arXiv:1906.00672*, 2019.

[17] J.-X. Zhang, Z.-H. Ling, and L.-R. Dai, "Forward attention in sequence-to-sequence acoustic modeling for speech synthesis," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4789–4793.

[18] H. Tachibana, K. Uenoyama, and S. Aihara, "Efficiently trainable text-to-speech system based on deep convolutional networks with guided attention," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4784–4788.

[19] Y. Yasuda, X. Wang, and J. Yamagishi, "Initial investigation of an encoder-decoder end-to-end tts framework using marginalization of monotonic hard latent alignments," *arXiv preprint arXiv:1908.11535*, 2019.

[20] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[21] H. Sun, X. Tan, J.-W. Gan, H. Liu, S. Zhao, T. Qin, and T.-Y. Liu, "Token-level ensemble distillation for grapheme-to-phoneme conversion," *arXiv preprint arXiv:1904.03446*, 2019.

[22] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.