TTS-BY-TTS: TTS-DRIVEN DATA AUGMENTATION FOR FAST AND HIGH-QUALITY SPEECH SYNTHESIS

Min-Jae Hwang[1]*, Ryuichi Yamamoto*[2]*, Eunwoo Song*[3] *and Jae-Min Kim*[3]

[1]Search Solutions Inc., Seongnam, Korea,
[2]LINE Corp., Tokyo, Japan, [3]NAVER Corp., Seongnam, Korea

ABSTRACT

In this paper, we propose a text-to-speech (TTS)-driven data augmentation method for improving the quality of a non-autoregressive (AR) TTS system. Recently proposed non-AR models, such as FastSpeech 2, have successfully achieved fast speech synthesis system. However, their quality is not satisfactory, especially when the amount of training data is insufficient. To address this problem, we propose an effective data augmentation method using a well-designed AR TTS system. In this method, large-scale synthetic corpora including text-waveform pairs with phoneme duration are generated by the AR TTS system, and then used to train the target non-AR model. Perceptual listening test results showed that the proposed method significantly improved the quality of the non-AR TTS system. In particular, we augmented five hours of a training database to 179 hours of a synthetic one. Using these databases, our TTS system consisting of a FastSpeech 2 acoustic model with a Parallel WaveGAN vocoder achieved a mean opinion score of 3.74, which is 40% higher than that achieved by the conventional method.

***Index Terms*—** Speech synthesis, text-to-speech, TTS-driven data augmentation, FastSpeech, Parallel WaveGAN

1. INTRODUCTION

Recently proposed end-to-end text-to-speech (TTS) systems, which generate a speech signal directly from an input text, have provided high-quality synthetic speech [1–5]. Popular end-to-end TTS systems consist of two subsystems: a sequence-to-sequence acoustic model, which generates the acoustic features of the speech signal from the input text, and a neural vocoder, which generates the speech waveform from the acoustic features.

Two approaches have focused on the acoustic model: *autoregressive* (AR) and *non-AR* approaches. In AR approach-based models, including Tacotron, the acoustic features are sequentially generated by conditioning previously generated ones [1,2]. As the models efficiently learn the temporal variation of acoustic features during the training procedure, they can provide a high-quality synthetic sound. However, the synthesis speed is slow due to the nature of sequential generation. In contrast, non-AR approach-based models, such as FastSpeech, can generate acoustic features in parallel [3, 4]. Thus, their generation speed is significantly faster than that of AR models and more suitable for real-time TTS applications. However, due to the limited capacity of non-AR modeling, there is room for improvement of their synthesis quality, especially when the training database is not sufficient.

To improve the quality of non-AR TTS, we propose a TTS-driven data augmentation method. In this system, the database for training ***target*** non-AR TTS (i.e., text-waveform pairs with phoneme duration) is generated by a well-designed ***source*** AR TTS system. First, we collect a large amount of text scripts while maintaining the recording script's phoneme distribution. Second, the Tacotron 2-based acoustic model generates acoustic features and phoneme durations from the collected texts. In detail, we adopted Tacotron 2 with a duration predictor [6, 7] because it has the capacity to accurately match the alignment between phonemes and acoustic features. Finally, a neural excitation vocoder synthesizes the speech waveforms from the generated features. Among the various types of vocoders, we chose an LP-WaveNet vocoder due to its good quality with stable generation [8]. After generating large-scale synthetic TTS corpora, these are used to train the target TTS system. As a large amount of text scripts enables the model to simulate various phoneme combinations, the target model's stability to the unseen text can be significantly improved.

We evaluated the proposed method via subjective listening tests. Specifically, the target non-AR TTS systems consisting of the FastSpeech 2 acoustic model [4] with a Parallel WaveGAN vocoder [9–11] were trained by five hours of recorded data and 179 hours of augmented data. Consequently, our system achieved a mean opinion score (MOS) of 3.74, which is 40% higher than that of systems trained without augmented data.

2. RELATIONSHIP TO PRIOR WORK

As the quality of recent TTS systems has reached a natural level, several attempts have been made to apply TTS-synthesized speech databases to speech applications. For instance, Laptev et al. [12] and Jia et al. [13] improved the performance of automatic speech recognition and speech translation systems by training models with synthetic speech databases generated by Tacotron. In the TTS applications, Sharma et al. [14] showed that the AR WaveNet-driven data augmentation is effective for improving the quality of Parallel WaveNet system [15]. Note that we mainly investigate the effectiveness of TTS-driven data augmentation on the performance of *acoustic model*, which was not considered in Sharma et al. [14].

On the other hand, the FastSpeech [3] adopted the idea of using the generated output from AR model to train the non-AR model. Even though this and our methods commonly transfer the quality of AR model to the non-AR model, there are clear differences: Our method uses the AR TTS model to increase the size of training database for *data augmentation* purpose; whereas the FastSpeech uses it to re-generate training set's acoustic parameters for the purpose of *knowledge distillation* [16].

3. TTS-DRIVEN DATA AUGMENTATION

As illustrated in Fig. 1, the training framework of the proposed system consists of three processes. First, a well-designed AR TTS model is trained by the recorded data (Fig. 1-(a)). Then, the synthetic speech corpus with phoneme durations is generated by feeding

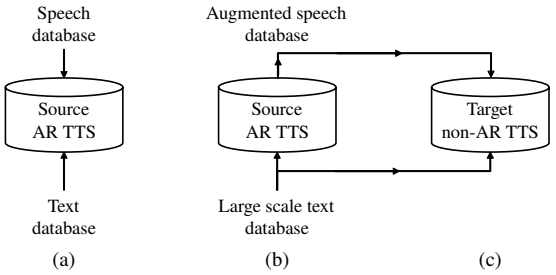


Fig. 1. The proposed training process with data augmentation: (a) source TTS training, (b) data augmentation, and (c) target TTS training.

Table 1. Summary of TTS systems.

System	Acoustic model	Neural vocoder
Source AR TTS	Tacotron 2 with duration predictor [6]	LP-WaveNet [8]
Target non-AR TTS	FastSpeech 2 [4]	Parallel WaveGAN [9]

collected text scripts to the source TTS system (Fig. 1-(b)). Finally, the target non-AR TTS model is trained using the augmented data (Fig. 1-(c)).

3.1. Source AR TTS model

The source and target TTS systems used for data augmentation experiments are summarized in Table. 1. To generate a *high-quality* synthetic TTS database, it is important to ensure that the speech generated by the source TTS model is aligned with the phonemic pronunciation. Thus, we adopt a Tacotron 2 decoder with a phoneme alignment approach [6], which has the capacity to accurately align the phoneme sequence with the acoustic features, as an acoustic model. In this method, an external duration model predicts the phoneme durations from the linguistic features, and the Tacotron 2 decoder generates the corresponding acoustic features. Then, the LP-WaveNet-based neural excitation vocoder [8] synthesizes the speech signals from these acoustic features. In this vocoder, the speech waveform is generated by the WaveNet-based mixture density network [17] within the framework of the human speech production mechanism [18]. As a result, it can stably generate more accurate speech signals than plain WaveNet models [19, 20].

3.2. Data augmentation

To prepare text scripts for data augmentation, we crawled 124,134 text scripts from news articles on the NAVER website[1]. As shown in Fig. 2, a total of 6,288,422 phonemes were collected, which was 40 times larger than the recorded database. Assuming that the recorded database had a balanced phoneme distribution, we tried to crawl text scripts to follow its phoneme distribution. Because a large number of phoneme sets enable the TTS model to learn various phoneme combinations, the target TTS can generate more stable synthetic speech in the condition of unseen text.

3.3. Target non-AR TTS model

As the acoustic model of the target TTS, we adopt the state-of-the-art FastSpeech 2 model thanks to its fast inference speed and good

[1] https://news.naver.com

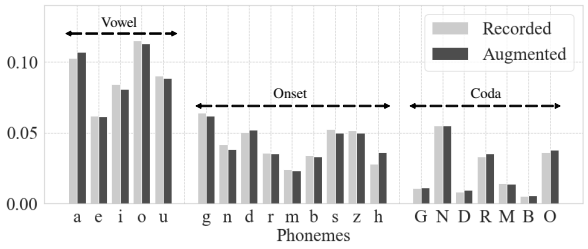


Fig. 2. Normalized histograms of phoneme distributions obtained from the recorded and augmented TTS databases. A number of phonemes used in the recorded and augmented databases were 155,715 and 6,288,422, respectively Note that lowercase and uppercase letters denote onset and coda consonants defined as Korean pronunciation, respectively.

quality [4]. There are several differences from its original version in our implementation. First, instead of using forced alignment to predict the phoneme duration [21], we use the phoneme duration used for source TTS system because it is already matched with the synthetic speech waveform. Second, to avoid the synthetic artifacts as reported in FastPitch [22], the pitch and energy modeled in the variance adaptor are averaged over every input symbol by using the given durations. Finally, the PostNet module of Tacotron 2 [2] is used to improve the generation accuracy of acoustic features.

To synthesize the speech waveform from the generated acoustic features, we use the Parallel WaveGAN vocoder [9], which is a non-causal WaveNet model that generates a speech waveform within a generative adversarial network framework [23]. As adversarial training enables realistic waveform generation, the WaveNet model can efficiently generate a speech signal of good quality faster than real time. The detailed configurations of the source and target TTS systems are described in Sec. 4.2.

4. EXPERIMENTS

4.1. Speech database

To train the source TTS model, a phonetically and prosodically balanced TTS corpus recorded by a female Korean professional speaker was used. The speech signals were sampled at 24 kHz with 16-bit quantization. In total, 2,970 utterances (five hours), 590 utterances (one hour), and 290 utterances (30 minutes) were used for the training, validation, and test sets, respectively.

As described in Sec. 3.2, the augmented TTS corpus was used to train the target TTS model. In total, 118,734 synthetic utterances (179 hours) and 5,400 synthetic utterances (eight hours) were used for the training and validation sets, respectively.

4.2. Model configuration

In all model training, the input and output features were normalized to have zero mean and unit variance. The weights were first initialized by the *Xavier* initializer [24], and then trained using an *Adam* optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-6}$ [25]. The initial learning rate was set to 10^{-3}, and we exponentially reduced it to 10^{-4} with a decaying rate of 0.33 per 100,000 iterations. Neural vocoders trained by the recorded database only were used in all experiments[2].

[2] It has been reported that using large size of training data is not crucial for neural vocoder [26]. In addition, in our preliminary experiments, we

4.2.1. Source AR TTS system

In the source TTS system, the acoustic features of the improved time-frequency trajectory excitation vocoder were extracted every 5 ms [27], which included 40-dimensional line spectral frequencies, fundamental frequency, energy, voicing flag, 32-dimensional slowly evolving waveform, and 4-dimensional rapidly evolving waveform, all of which composed a total 79-dimensional feature vector.

The acoustic model of the source TTS consists of three sub-modules : a context analyzer, a context encoder, and a Tacotron decoder [7]. In the context analyzer, 354-dimensional phoneme-level linguistic feature vectors consisting of 330 categorical and 24 numerical contexts were first extracted from the input text. Then, the duration predictor, which consists of three fully connected (FC) layers with 1,024, 512, and 256 units, and a long short-term memory (LSTM) layer with 128 memory blocks, estimated the duration of each phoneme. Based on the estimated durations, the phoneme-level linguistic features were upsampled to that of the frame level. In the context encoder, high-level context features were further extracted by feeding the frame-level linguistic features to the three convolution layers with 10×1 kernels and 512 channels, bidirectional LSTM with 512 memory blocks, and FC layers with 512 units. Then, the Tacotron decoder, which consists of PreNet, PostNet, and main unidirectional LSTM, generated the acoustic features. First, the previously generated acoustic features were fed into PreNet, which consists of two FC layers with 256 units. Then, the outputs of PreNet and a context-embedding module were passed through two unidirectional LSTM layers with 1,024 memory blocks, followed by two projection layers with 79 units to generate the acoustic features. Finally, PostNet, which consists of five convolution layers with 5×1 kernels and 512 channels, was used to add the residual elements of the generated acoustic features for more accurate generation.

In the configuration of LP-WaveNet, the dilations were set to $[2^0, 2^1, ..., 2^9]$ and repeated three times, resulting in 30 layers of residual blocks and 3,071 samples of the receptive field. In each residual block, 128 channels of convolution layers were used. The number of output dimensions was set to two to generate the mean and standard deviation of Gaussian distribution. The weight normalization technique, which normalizes the weight vectors to have a unit length, was applied [28].

To improve the spectral clarity of the synthesized speech, the spectral domain sharpening filter with a coefficient of 0.95 [27] was applied as a post-processing technique. In addition, to generate a cleaner speech sound, LP-WaveNet's generated scale parameter on the voiced region was reduced by a factor of 0.85 [8].

4.2.2. Target non-AR TTS system

In the target TTS system, an 80-dimensional Mel-spectrogram extracted every 10 ms was used as acoustic features [2]. Like the source TTS, the acoustic model of the target TTS consisted of three sub-modules: a feed-forward Transformer (FFT) encoder, a variance adaptor, and an FFT decoder [4].

First, the phoneme sequence defined by 55 vocabulary passed through a 256-dimensional embedding layer. Four FFT blocks were used in both the encoder and the decoder. In each FFT block, the hidden size of the self-attention layer and the number of attention heads were 384 and 2, respectively. The kernel sizes of convolution layer in the two-layer convolutional network after the self-attention layer were set to 9 and 1, with input/output sizes of 384/1,024 for the first layer and 1,024/384 for the second layer. In the decoder, the output FC layer converted the 256-dimensional hidden states into 80-dimensional Mel-spectrograms with residual components predicted by PostNet. The variance adaptor consisted of three variance predictors estimating duration, pitch, and energy components, respectively. The variance predictor was composed of five convolution blocks, each containing 1D convolution and rectified linear unit (ReLU) activation, followed by layer normalization and dropout with a probability of 0.5. The numbers of dimensions and the kernel size of convolution layer were set to 256 and 5, respectively. The final FC layer converted 256-dimensional hidden features to the output variance parameter.

Similar to LP-WaveNet, Parallel WaveGAN consisted of 30 layers of dilated residual convolution blocks with three dilation cycles. A number of residual and skip channels were set to 64, and the convolution filter size was set to 5. The resulting receptive field of the model was 12,277. Weight normalization was applied to all convolutional layers [28]. The discriminator configuration was the same as that of Parallel WaveGAN [9].

could not confirm quality improvements when the neural vocoder is trained by augmented database.

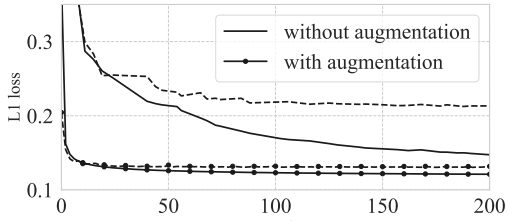


Fig. 3. L1 loss obtained during the training process of the FastSpeech 2 model with and without augmentation. The solid and dashed lines show the training and validation losses, respectively.

4.3. Results

4.3.1. Training efficiency

Fig. 3 shows the training and validation losses obtained during the training process of the FastSpeech 2 model. Our observations are summarized as follows: (1) with data augmentation, the model exhibited significantly less loss. This indicated that data augmentation was beneficial to accurately estimate the acoustic features. (2) The gap between the training and validation losses in the augmentation case was significantly narrower than the case without augmentation. This indicated that model's generalization performance was also improved where it provided the consistent estimation results to both the seen (train) and unseen (validation) data.

4.3.2. Subjective listening tests

To evaluate the perceptual quality of the proposed system, MOS listening tests were performed[3]. Eighteen native Korean listeners were asked to score the randomly selected 20 synthesized utterances from the test set using a following possible 5-point MOS responses: 1 = Bad, 2 = Poor, 3 = Fair, 4 = Good, 5 = Excellent.

Table 2 summarizes the MOS test results, whose trends can be analyzed as follows: (1) the AR TTS (TTS_{AR}) system performed better than the non-AR TTS (TTS_{NAR}) system because of the AR model's better capacity to capture the temporal variation of the speech signal. (2) When the non-AR TTS system was trained by the augmented database, its perceptual quality was significantly improved (B4 vs. P1). (3) When both the recorded and augmented databases were used to train the non-AR TTS system, its perceptual

[3]Generated audio samples are available at the following URL: https://min-jae.github.io/icassp2021/

Table 2. Subjective MOS test results with 95% confidence intervals. Note that the score of the analysis/synthesis system can be considered the upper bound of the TTS system

System	Model	Analysis / Synthesis	Training Database		MOS
			Recorded	Augmented	
R	Recorded	–	–	–	4.56±0.13
B1	TTS_{AR}	Yes	–	–	4.11±0.17
B2		–	Yes	–	3.99±0.16
B3	TTS_{NAR}	Yes	–	–	3.84±0.16
B4		–	Yes	–	2.68±0.34
P1		–	–	Yes	3.55±0.25
P2		–	**Yes**	**Yes**	**3.74±0.20**

R: recording; Bi: i^{th} baseline; Pi: i^{th} proposed model; TTS_{AR}: source AR TTS system; TTS_{NAR}: target non-AR TTS system. The non-AR TTS system with the highest score is shown in boldface.

Table 3. Subjective MOS test results with 95% confidence intervals of augmentation applied to the end-to-end Tacotron 2 model instead of the FastSpeech 2 model.

System	Model	Training database		MOS
		Recorded	Augmented	
B4-T	Tacotron 2 + PWG	Yes	–	2.89±0.36
P1-T		–	Yes	3.70±0.26
P2-T		**Yes**	**Yes**	**3.72±0.32**

Bi: i^{th} baseline; Pi: i^{th} proposed model; T: Tacotron; PWG: Parallel WaveGAN. The system with the highest score is shown in boldface.

Table 4. Subjective MOS test results with 95% confidence intervals with the recorded data increased from 5 to 20 hours.

System	Model	Training database		MOS
		Recorded	Augmented	
B2-L	TTS_{AR}	Yes	–	4.22±0.15
B4-L	TTS_{NAR}	Yes	–	3.47±0.43
P1-L		–	Yes	3.80±0.27
P2-L		**Yes**	**Yes**	**3.95±0.23**

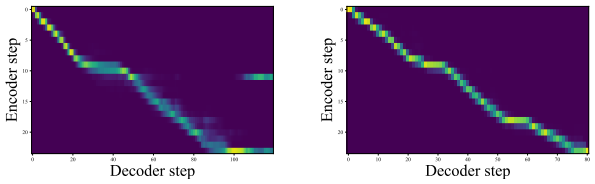
Bi: i^{th} baseline; Pi: i^{th} proposed model; L: trained by a large database; TTS_{AR}: source AR TTS system; TTS_{NAR}: target non-AR TTS system. The non-AR TTS system with the highest score is shown in boldface.

quality was further improved (P1 vs. P2). In particular, even though only five hours of a natural database were used, the quality of the non-AR TTS system achieved 3.74 MOS, which is 40% higher than the system without augmentation (B4 vs. P2).

4.4. Additional experiments

4.4.1. Data augmentation for AR TTS system

To examine whether the proposed method works well with attention mechanism in the end-to-end AR model, we replaced the non-AR FastSpeech 2 model with the Tacotron 2 model [2]. The structure of Tacotron 2 was similar to our source TTS model, but it followed its original version. First, instead of using the external duration predictor, we used a location-sensitive attention mechanism [29]. Second, instead of using the linguistic features, the phoneme sequence was used as input. The detailed configuration followed the ESPnet-TTS toolkit [30]. Note that this architecture contained a potential alignment failures including attention skip or collapse.



(a) (b)

Fig. 4. Attention alignments generated by Tacotron 2 acoustic models (a) without and (b) with augmentation.

As shown in Fig. 4, the attention alignments generated by the augmentation method were clearer than those of the model without augmentation. We conjecture that the data augmentation was beneficial to improve the robustness to the unseen text patterns. In the subjective evaluation[4] results summarized in Table. 3, we figured out that the proposed data augmentation further improved the perceptual quality of Tacotron 2 acoustic model by achieving 3.72 MOS, which is 28% higher than the results without augmentation (B4-T vs. P1-T and P2-T).

4.4.2. Data augmentation with enough recordings

To verify the effectiveness of the proposed method with a sufficient amount of recorded data, we conducted additional experiments by increasing the size of recorded data from 2,970 (five hours) to 11,890 utterances (20 hours). We re-trained the source AR TTS system with the larger database and re-generated augmentation data. Then, the target non-AR TTS system was also re-trained using the generated database.

The subjective evaluation[4] results are shown in Table. 4. The perceptual quality of source TTS was improved as the amount of training data was increased (B2 vs. B2-L). Moreover, the perceptual quality of the non-AR TTS system was significantly improved when the augmented database was included in the training process (B4-L vs. P1-L and P2-L). In particular, when both recorded and augmented database were used, the target TTS achieved 3.95 MOS which is higher score than the case to train the model with smaller database (P2 vs. P2-L).

5. CONCLUSION

In this paper, we proposed a TTS-driven data augmentation method to improve a quality of non-AR TTS system. Using a large-scale synthetic TTS database generated by a high-quality AR TTS system, we successfully improved the quality of the target TTS system. The experimental results verified that the proposed data augmentation was effective in various experimental conditions, especially when the training data were insufficient. As we collected the text scripts during augmentation by keeping the recorded data's phoneme distribution, the future studies should test the augmentation with various phoneme distributions, such as uniformly distributed case.

6. ACKNOWLEDGEMENTS

This work was supported by Clova Voice, NAVER Corp., Seongnam, Korea. The authors would like to thank SungJun Choi and Sangkil Lee at NAVER Corp., Seongnam, Korea, for their support.

[4]The experimental settings for the MOS test were the same as described in Sec. 4.3.2

7. REFERENCES

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

[2] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan, R. A. Saurous, Y. Agiomvrgiannakis, and Y. Wu, "Natural TTS synthesis by conditioning WaveNet on Mel spectrogram predictions," in *Proc. ICASSP*, 2018, pp. 4779–4783.

[3] Y. Ren, Y. Ruan, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T. Liu, "FastSpeech: Fast, robust and controllable text to speech," in *Proc. NeurIPS*, 2019, pp. 3165–3174.

[4] Y. Ren, C. Hu, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, "FastSpeech 2: Fast and high-quality end-to-end text-to-speech," in *Proc. ICLR (in press)*, 2021.

[5] W. Ping, K. Peng, and J. Chen, "ClariNet: Parallel wave generation in end-to-end text-to-speech," in *Proc. ICLR*, 2019.

[6] T. Okamoto, T. Toda, Y. Shiga, and H. Kawai, "Tacotron-based acoustic model using phoneme alignment for practical neural text-to-speech systems," in *Proc. ASRU*, 2019, pp. 214–221.

[7] E. Song, M.-J. Hwang, R. Yamamoto, J.-S. Kim, O. Kwon, and J.-M. Kim, "Neural text-to-speech with a modeling-by-generation excitation vocoder," in *Proc. Interspeech*, 2020, pp. 3570–3574.

[8] M.-J. Hwang, F. Soong, E. Song, X. Wang, H. Kang, and H.-G. Kang, "LP-WaveNet: Linear prediction-based WaveNet speech synthesis," in *Proc. APSIPA*, 2020, pp. 810–814.

[9] R. Yamamoto, E. Song, and J. Kim, "Parallel WaveGAN: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram," in *Proc. ICASSP*, 2020, pp. 6199–6203.

[10] E. Song, R. Yamamoto, M.-J. Hwang, J.-S. Kim, O. Kwon, and J.-M. Kim, "Improved Parallel WaveGAN vocoder with perceptually weighted spectrogram loss," in *Proc. SLT*, 2021, pp. 470–476.

[11] R. Yamamoto, E. Song, M.-J. Hwang, and J.-M. Kim, "Parallel waveform synthesis based on generative adversarial networks with voicing-aware conditional discriminators," in *Proc. ICASSP (in press)*, 2021.

[12] A. Laptev, R. Korostik, A. Svischev, A. Andrusenko, I. Medennikov, and S. Rybin, "You do not need more data: Improving end-to-end speech recognition by text-to-speech data augmentation," in *Arxiv*, 2020.

[13] Y. Jia, M. Johnson, W. Macherey, R. J. Weiss, Y. Cao, C. Chiu, N. Ari, S. Laurenzo, and Y. Wu, "Leveraging weakly supervised data to improve end-to-end speech-to-text translation," in *Proc. ICASSP*, 2019, pp. 7180–7184.

[14] M. Sharma, T. Kenter, and R. Clark, "StrawNet: Self-training WaveNet for TTS in low-data regimes," in *Proc. Interspeech*, 2020, pp. 3550–3554.

[15] A. van den Oord, Y. Li, I. Babuschkin, K. Simonyan, O. Vinyals, K. Kavukcuoglu, G. van den Driessche, E. Lockhart, L. C. Cobo, F. Stimberg, N. Casagrande, D. Grewe, S. Noury, S. Dieleman, E. Elsen, N. Kalchbrenner, H. Zen, A. Graves, H. King, T. Walters, D. Belov, and D. Hassabis, "Parallel WaveNet: Fast high-fidelity speech synthesis," in *Proc. ICML*, 2018, pp. 3915–3923.

[16] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," in *Proc. NIPS*, 2015.

[17] C. M. Bishop, "Mixture density networks," Tech. Rep., 1994.

[18] T. F. Quatieri, *Discrete-time speech signal processing: principles and practice*. Pearson Education India, 2006.

[19] A. Tamamori, T. Hayashi, K. Kobayashi, K. Takeda, and T. Toda, "Speaker-dependent WaveNet vocoder," in *Proc. Interspeech*, 2017, pp. 1118–1122.

[20] A. van den 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," in *Arxiv*, 2016. [Online]. Available: https://arxiv.org/pdf/1609.03499.pdf

[21] M. McAuliffe, M. Socolof, S. Mihuc, M. Wagner, and M. Sonderegger, "Montreal Forced Aligner: Trainable Text-Speech Alignment Using Kaldi," in *Proc. Interspeech*, 2017, pp. 498–502.

[22] A. Lańcucki, "FastPitch: Parallel text-to-speech with pitch prediction," in *Proc. ICASSP (in press)*, 2021.

[23] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. NIPS*, 2014, pp. 2672–2680.

[24] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. AISTATS*, 2010, pp. 249–256.

[25] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Arxiv*, 2014. [Online]. Available: http://arxiv.org/abs/1412.6980

[26] K. Matsubara, T. Okamoto, R. Takashima, T. Takiguchi, T. Toda, Y. Shiga, and H. Kawai, "Investigation of training data size for real-time neural vocoders on CPUs," *Acoust. Sci. Tech.*, vol. 42, no. 1, pp. 65–68, 2021.

[27] E. Song, F. K. Soong, and H.-G. Kang, "Effective spectral and excitation modeling techniques for LSTM-RNN-based speech synthesis systems," *IEEE/ACM Trans. Audio, Speech, and Lang. Process.*, vol. 25, no. 11, pp. 2152–2161, 2017.

[28] T. Salimans and D. P. Kingma, "Weight normalization: A simple reparameterization to accelerate training of deep neural networks," in *Proc. NIPS*, 2016, pp. 901–909.

[29] J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *Proc. NIPS*, 2015, pp. 577–585.

[30] 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," in *Proc. ICASSP*, 2020, pp. 7654–7658.