Neural Text-to-Speech Adaptation from Low Quality Public Recordings

Qiong Hu, Erik Marchi, David Winarsky, Yannis Stylianou, Devang Naik, Sachin Kajarekar

Apple Inc. USA

qiong@apple.com

Abstract

Neural Text-to-Speech (TTS) synthesis is able to generate high-quality speech with natural prosody. However, these systems typically require a large amount of data, preferably recorded in a clean and noise-free environment. We focus on creating character voices from low quality public recordings and our findings show that even with a large amount of data from a specific speaker, it is challenging to train a speaker-dependent neural TTS model. In order to improve the voice quality, while simultaneously reducing the amount of data required, we introduce meta-learning to adapt the neural TTS front-end. We propose three approaches for multi-speaker systems: (1) a lookup-table-based system, (2) a speaker representation derived from the Personalized Hey Siri (PHS) system, and (3) a system with no speaker encoder. Results show that: i) By using a significantly smaller number of target voice recordings, the proposed system based on embeddings trained from the PHS system can generate comparable quality and speaker similarity to the speaker-dependent model trained solely on the target voice. ii) Applying meta-learning to Tacotron can effectively learn a representation of an unseen speaker. iii) For low quality public recordings, the adaptation based on the multi-speaker corpus can generate a cleaner target voice in comparison with the speaker-dependent model.

Index Terms: speech synthesis, speaker adaptation, multi-speaker training, meta-learning

1. Introduction

Recent advancements in neural TTS systems, which are fully based on end-to-end (E2E) models [1, 2, 3, 4, 5, 6, 7, 8], can generate high-quality speech with natural prosody. These approaches mainly focus on clean and noise-free recordings, which generally require a large amount of studio-recorded speech by a professional talent. We investigate whether we can leverage the E2E model capabilities to create character voices with various personalities and accents. There are a number of publicly-available datasets from TV shows, news and other sources [9]. These corpora of under-utilized recordings can be used to create various identities and character voices.

Many existing works [10, 11] have used audiobooks as the training corpus, however, it is much more challenging to utilize low quality public recordings. The drawback of harnessing such data is three-fold: 1) Voices extracted from such datasets are often casual and conversational [12], as opposed to professionally produced content or audiobooks. Disfluencies, repetitions, and pauses are frequent and some utterances are challenging to understand and transcribe phonetically. 2) Voices extracted from TV shows or interviews are often recorded in uncontrolled environments. Most recordings contain ambient noise and reverberation. It is difficult to assess how robust a typical neural TTS system will be with such data. 3) Extracted voices may contain an expressive style for which no prosodic annotation is available. The average pitch and duration of characters are more variable, which makes the synthesis using conversational speech more challenging. Although neural TTS systems are capable of generating high-quality speech [2, 1], no study has assessed the impact of using such types of corpora for training. The main goal of this work is to train a neural TTS system on public recordings, which is limited in size, noisy, and of low quality, to generate high-quality TTS output. Preprocessing this type of dataset is an expensive process, particularly when transcriptions are not available. Therefore, we further address how to generate a character voice with limited data. More specifically, we propose and compare adaptation methods for neural TTS to understand which generates the most natural and similar voice compared to the target speaker.

For the low-resource setting for neural TTS, [13] presents a meta-learning method applied to Wavenet. The task-independent parameters for mapping text-to-speech are first learned by the network, then the target speaker data is applied to fine-tune the Wavenet model and the speaker embedding. However, the output waveform is still predicted from hand-designed linguistic features and fundamental frequency, which require significant domain expertise and may introduce additional errors. Direct application of meta-learning to Tacotron has yet to be tested. We do not have a corresponding clean version of the data to train a separate network for denoising the target voice. Furthermore, noise conditions in actual low quality public recordings are more complicated. Adding an additional encoder to decrease the noise [14] makes the network difficult to optimize. [15] has shown the effectiveness of decoupling the speaker encoder from speech synthesis by training a speaker verification model. However, the authors show the difficulty in generating high-similarity voices for unseen speakers. All those experiments are conducted on corpora where written transcriptions are available. In our case, the task is more challenging since the data contains variable speaking styles, mismatched background conditions, and content significantly different from that in the training set.

To address all the issues mentioned above, we first train a neural-TTS system on a single speaker whose material is collected from a conversational interview of 19 hours. Next, we propose various methods for adaptation. We introduce two different speaker encoders to extract a speaker representation. The first is based on a look-up table and the second uses a dense speaker verification model trained for PHS. We further propose applying meta-learning to a Tacotron system with three different approaches for the new speaker. Data cleaning may boost the performance of a speaker-dependent model using the entire 19 hours of recordings. However, it is not a focus for this paper.

This paper is organized as follows: We first introduce our neural TTS frontend and backend in Section 2. Various adaptation methods and proposed systems are explained in Section 3. Experiments and results are shown in Section 4. The discussion and conclusions are listed in Section 5.

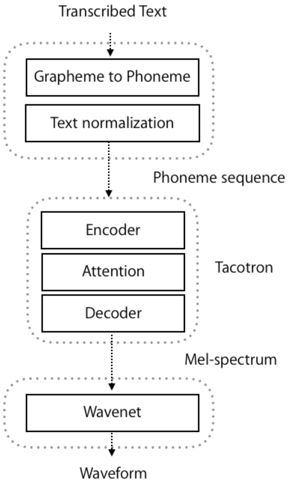

Figure 1: *Neural TTS baseline*

2. Baseline description

2.1. System structure

Compared to the unit selection system presented in [16], our neural TTS system is capable of generating natural speech quality with high flexibility [17, 15, 13, 18, 19]. The front-end system is based on Tacotron [1], where the Mel-spectrum is predicted from phoneme sequences and the waveform is generated by the neural vocoder in the backend. The input is text, which first undergoes text normalization and Grapheme-to-Phoneme conversion [20, 16, 21]. The Tacotron itself is an encoder-decoder with an attention network [4] with zoneout regularization [22] and location-sensitive attention [23]. For speaker-dependent training, only the phoneme input is converted into hidden units, and the acoustic output is predicted from the auto-regressive decoder. The output Mel-spectrum is a block of 2 frames computed from a 25 ms window with a 10 ms shift. For the backend, we use a modified Wavenet architecture [2] that, instead of linguistic features and fundamental frequency, uses the Mel-spectrum as input to predict time-domain samples [1]. The architecture is shown in Figure 1.

3. Adaptation systems

To generate the voice of the target speaker with a small amount of training data, a multi-speaker Tacotron is utilized in [17, 13, 18, 15, 3]. The main difference compared to the baseline Tacotron described in Section 2 is that an additional speaker encoder is added to explicitly model the speaker's identity. The auto-regressive decoder is trained to encode the concatenation of the embeddings obtained from the phoneme encoder and the embeddings extracted with the speaker encoder.

3.1. Speaker encoder

Speaker encoder has been widely used in the multi-speaker generative model for capturing the representation from target speaker such as speaker characteristics, speaking style and accent. In order to retrieve these characteristics, a known approach is to use a speaker identifier and repeat it for the entire input text followed by a look-up table [24, 25, 18]. The concatenation of the speaker embedding and the phoneme encoder output is used as input for the attention layer. To further distinguish the speaker, [15, 17, 26] have introduced a trainable speaker encoder to map the speaker's voice into a latent speaker space. Instead of using speaker ID, the speech is used to extract embeddings as a fixed dimensional vector in the speaker encoder, which can be either optimized together with the synthesis training [17] or learned from a pretrained speaker verification network [15]. The advantage of the latter method is that no transcribed audio is needed to train the speaker encoder [15]. Here, we used the speaker verification system introduced in [27], which is used for PHS [28] but can also extract speaker information independently of the variable linguistic inputs. Since the speaker discriminative transform used in PHS models contains Siri requests, which are short and offer less text variations compared to the one presented in the traditional TTS training set, the network may not be able to capture accurate speaker identities from training utterances. So our first question is whether the speaker verification network for PHS can be used as a speaker encoder?

Table 1: *Different speaker embeddings and adaptation methods for Tacotron (When there is no meta-learning, target voice and other speakers are trained at the same time. Otherwise, we pretain on other speakers and then fine-tune the model to the target voice)*

ID	Speaker embedding	meta-learning
onehot	lookup table embedding	no
sv	speaker verification output	no
onehot-pf	lookup table embedding	yes
sv-pf	speaker verification system	yes
noid-pf	none	yes

To answer the question, we use a LSTM system which maps a sequence of MFCCs extracted from the speech utterance into a speaker embedding vector via our pretrained speaker encoder [27]. The parameters of the speaker encoder are optimized via a curriculum learning procedure [29] to improve robustness under various acoustic conditions and text variability. The input for the LSTM is 20 dimensional MFCCs extracted from the waveform. The layout consists of a recurrent layer containing 512 LSTM units followed by a fully connected linear layer with 128 units. Embeddings from the linear layer are calculated for every utterance from the corpus, and then averaged to obtain the final speaker embeddings. A dataset containing more than 18k English speakers is used to train the encoder.

3.2. Meta-learning

An advange of using an auxiliary speaker encoder is that the system is able to generate a voice for unseen speakers without retraining the Tacotron model. However, compared to the target voice, the speaker similarity from the synthesized speech drops significantly especially for unseen speakers from a different dataset [15]. Here we propose several ways to use meta-learning [30] to first train a multi-speaker Tacotron, and then adapt it on the target character. In [13], during adaptation, a target speaker voice, linguistic features, and corresponding fundamental frequency are used as input to adapt the speaker embedding vector and Wavenet parameters. However for Tacotron, fundamental frequency is not used as input, and the speaker encoder may not be well-adapted in the fine-tuning process. Therefore, our second question is whether meta-learning can be applied to Tacotron, and if so, how should the system be designed.

Based on the speaker encoder type, we present three ways to train a multi-speaker Tacotron with fine-tuning. We can first pretrain a multi-speaker Tacotron using a lookup table (*onehot* in Table 1) based on a large database. In the fine-tuning stage, supposing M^{target} , $M^{predict}$, and L^{target} are the target, predicted Mel-spectrum, and linguistic features for the target voice, our aim is to minimize the difference between target and predicted speaker Mel-spectrum by optimizing the weights of both the speaker embedding $W_{speaker}^{onehot}$ and Tacotron parameters $W_{Tacotron}$ (*onehot-pf* with pretraining-fine-tuning (*pf*)),

$$min\mathcal{L}\{f(M^{target}, M^{predict})|L^{target}; W_{speaker}^{onehot}, W_{Tacotron}\}.$$

Similar to Section 3.1, we can also apply the speaker verification system for PHS to predict the speaker reference vector E^{sv}. As this speaker encoder is trained separately from Tacotron, during the adaptation period, only $W_{Tacotron}$ needs to be refined on the target speaker (*sv-pf*),

$$min\mathcal{L}\{f(M^{target}, M^{predict})|L^{target}, E^{sv}; W_{Tacotron}\}.$$

By investigating the meta-learning process, we find in the case of *onehot-pf*, both the speaker encoder and Tacotron weights are learned in the pretraining process. Based on the findings reported in [13, 17], by optimizing the weights for the speaker encoder and the entire synthesis model, the system performs better than when only the speaker encoder is adapted. In other words, instead of treating the weights from the speaker encoder and the synthesis model as task-dependent and task-independent parameters respectively, we have optimized the entire model weights for the two tasks simultaneously. Under such conditions, our hypothesis is that Tacotron without a speaker encoder can also perform well in the meta-learning process. Thus, our third question is: is a speaker encoder still needed in the pretraining-fine-tuning process for Tacotron?

In the pretraining process, we first train a multi-speaker Tacotron, without conditioning on speaker identity, and use only linguistic features as input. This can be viewed as our first step toward learning a general text-to-speech mapping. During the fine-tuning stage, we adapt the Tacotron weights learned from the mutli-speaker dataset to the target character. The difference between adaptation methods are shown in Table 1. Architectures for the proposed multi-speaker Tacotron are shown in Figure 2. Our new optimization function becomes (*noid-pf*):

$$min\mathcal{L}\{f(M^{target}, M^{predict})|L^{target}; W_{Tacotron}\}.$$

4. Experiments

4.1. Corpus

Our character recordings are based on a set of spontaneous interviews taken over the course of 3 days, and therefore qualities such as data volume and background noise are not consistent, due to varying configurations and microphone locations. There are multiple voices in the recording, and one male speaker with an American English accent is selected as our target character in this preliminary investigation. Corresponding text, punctuation, and lexicon are manually transcribed by linguists. Overall, 19 hours of data without overlapping speakers were collected at a sampling frequency of 16kHz and then denoised. The corpus contains a high speaking rate and energy variance with stammer and hesitations. 1.5 hours of speech with a stable performance (e.g., speaking style, rate and etc.) are manually selected from the 19 hours for the adaptation experiment. 100 speakers from the VCTK corpus [31] are used for training the multi-speaker system. The whole corpus consists of around 40 hours of audio, with each speaker reading the same script for about 20 minutes each.

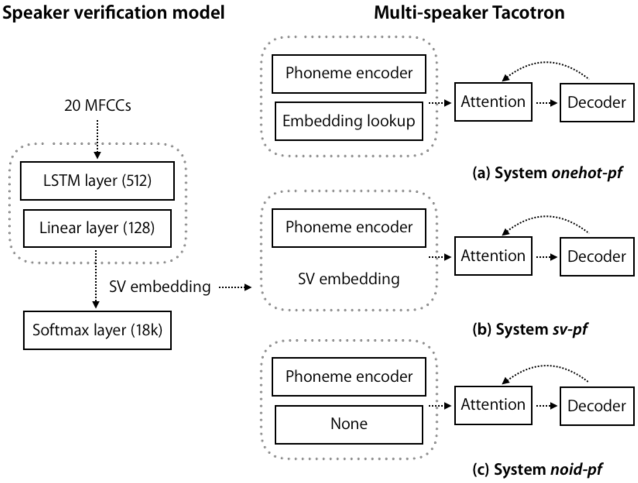

Figure 2: *Different multi-speaker systems for meta-learning ((a) speaker encoder based on the embedding lookup table; (b) speaker encoder based on the speaker verification model trained separately; (c) no speaker encoder)*

4.2. Evaluations

To understand how the neural TTS system performs for a character voice based on low quality public recordings, we first train a speaker-dependent Tacotron and Wavenet with the entire 19 hours of the single speaker dataset. To evaluate the influence of the quality of the corpus on neural TTS, we also choose a separate 13 hour high-quality Siri corpus from a professional English male talent [32]. This corpus is used to build a speaker-dependent model with the same configuration. 20 test sentences are generated for each system and each sentence is evaluated by 30 native listeners. Synthesized voices are evaluated from three aspects: 1) the speech naturalness Mean Opinion Score (MOS) (from 1 to 5) [33] 2) speaker similarity MOS (from 1 to 5) compared to the target character voice 3) Signal Noise Ratio (SNR). Table 2 shows results for the natural target voice (*natural_target*), generated speaker-dependent Siri voice (*gen_siri*), and generated speaker-dependent target voice (*gen_target*).

We can see that our neural TTS system based on the traditional Siri corpus (*gen_siri*) achieves a naturalness MOS score of 4.25, which shows the ability of the system to generate high-quality speech. However, when we switch the training corpus to low quality public recordings with more data (*gen_target*), even under the same setup, the naturalness MOS drops significantly to 3.56. The character dataset was recorded in an uncontrolled environment, the SNR for the denoised natural voice (*natural_target*) is around 40.5 dB, which is lower than the generated Siri voice (61.9 dB[1]). Furthermore, the test shows that the neural TTS is not robust when training on the noisy corpus, and the SNR for the speaker-dependent model decreases to

[1] SNR from natural Siri sentences is around the same range.

Table 2: *Naturalness, similarity MOS with 95% confidence interval and SNR (mean with variance) for the natural and synthesized voice (second and third rows are trained under the same configuration)*

System	Naturalness	Similarity	SNR (dB)
natural_target	4.32 ± 0.04	–	40.5 ± 4.2
gen_siri	4.25 ± 0.04	1.26 ± 0.04	61.9 ± 2.1
gen_target	**3.56 ± 0.04**	**3.77 ± 0.04**	**35.2 ± 0.8**

35.2 dB for the target voice. We draw the conclusions that 1) The performance of the neural TTS system is sensitive to the training corpus type. 2) It is difficult to generate a high-quality voice from low quality public recordings, and the neural TTS system is not robust when trained on a noisy dataset. 3) Simply increasing the corpus size for the dataset will not always improve neural TTS performance.

Next, we evaluate whether we can use adaptation to achieve a similar or higher voice quality. To answer the first question of whether the PHS-based speaker verification output can be used to represent the speaker identity, we trained a multi-speaker Tacotron using the output from the speaker encoder (*sv-1.5h*). Both VCTK and the 1.5 hour target character's voice are used to train the frontend and backend (no speaker ID is used in Wavenet [34, 35]). For comparison, a multi-speaker Tacotron based on the lookup table trained on VCTK and 1.5 hour target voice is also tested (*onehot-1.5h*). From results in Table 3, we can see that the *sv-1.5h* and *onehot-1.5h* system can generate a similar level of quality and similarity for the target voice (MOS difference between *onehot-1.5h* and *sv-1.5h* is not statistically significant, but *gen_target* is statistically better than both of them). This supports our first hypothesis. While the naturalness and similarity MOS of adaptation systems decreases compared to *gen_target*, the mean SNR for generated samples increases. This indicates that training the multi-speaker model can boost the speech quality for target voices.

For the systems in Table 3, to synthesize a new target speaker's voice, a multi-speaker model based on VCTK and the target speaker needs to be trained. Therefore, we next address the second and third questions: can meta-learning be applied to Tacotron in the case of an unseen speaker and can it generate a good quality voice without a speaker encoder? For the pretraining process, three different multi-speaker Tacotron systems are trained: 1) without conditioning on the speaker encoder (*noid-pf-1.5h*), 2) based on the lookup table (*onehot-pf-1.5h*), and 3) based on the verification output as a speaker reference (*sv-pf-1.5h*). Only the VCTK database is utilized in this stage. Next, the target character's voice is utilized as an unseen speaker to fine-tune the entire system based on the pretrained model.

Results are shown in Table 4. Surprisingly, with meta-learning, SNR values from all adaptation systems are further increased compared to the ones in Table 3. For system *onehot-pf-1.5h* and *noid-pf-1.5h*, SNR is improved almost 3 dB. In terms of speech naturalness, *sv-pf-1.5h* generates the highest quality, but it is not statistically significant compared with system *gen_target*, *onehot-pf-1.5h* and *noid-pf-1.5h*. This indicates that during the fine-tuning period, Tacotron weights are not only capable of learning the general text-to-speech mapping, but they are also adjusted to learn from the target's voice identity. Meanwhile, system *gen_target* and *sv-pf-1.5h* can generate a voice with higher speaker similarity compared to system *noid-pf-1.5h* and *onehot-pf-1.5h*. Results in Table 4 are promising, as system *sv-pf-1.5h* has achieved comparable speech naturalness and similarity as the system trained on the whole dataset. The noise in the generated voices has been improved while no additional noise encoders have been applied in our dataset. We would like to investigate the capability of the proposed method for higher noise conditions and test the system's performance on zero-short learning as a next step.

Table 3: *Naturalness, similarity MOS with 95% confidence interval and SNR (mean with variance) for the synthesized target character's voice from multi-speaker model tacotron based on VCTK and target voice 1.5 hour data.*

System	Naturalness	Similarity	SNR (dB)
onehot-1.5h	3.21 ± 0.04	3.53 ± 0.06	36.0 ± 1.1
sv-1.5h	3.12 ± 0.05	3.35 ± 0.06	35.5 ± 0.8

Table 4: *Naturalness, similarity MOS with 95% confidence interval and SNR (mean with variance) for the synthesized target character's voice from 1.5 hour target voice based on meta-learning.*

System	Naturalness	Similarity	SNR (dB)
onehot-pf-1.5h	3.52 ± 0.04	3.52 ± 0.06	**38.0 ± 0.7**
sv-pf-1.5h	**3.60 ± 0.04**	**3.69 ± 0.06**	36.5 ± 0.5
noid-pf-1.5h	3.56 ± 0.04	3.60 ± 0.06	**38.2 ± 0.7**

5. Discussion and conclusions

Although a large number of neural-based TTS systems have been proposed, few studies have measured the performance after training on recordings from low quality public recordings. We have shown it is a challenge to train a neural-based TTS system with this data even with a standard-sized corpus. Therefore, we use adaptation with a small amount of target voice material. Systems with various adaptation methods and configurations were proposed. We first introduce that using embeddings from a dense speaker verification model used for PHS data can generate quality comparable to the system based on a lookup table. Both systems can generate a less noisy voice compared to the network which is trained only on character recordings. Three approaches for multi-speaker systems with pretraining and fine-tuning are introduced to further improve the performance of adaptation systems. We prove that meta-learning is effective to learn speaker parameters of unseen characters for Tacotron by pretraining the model on a large multi-speaker database. Results show that by using a significantly smaller number of target voice recordings, the proposed system based on embeddings extracted from the PHS system can generate comparable quality and speaker similarity to the speaker-dependent model trained solely on the target voice. For future work, we will focus on designing early stopping in the fine-tuning on zero-short learning and other methods for corpus denoising.

6. Acknowledgments

Authors greatly appreciate helpful discussions with Greg Townsend, Laurence Schwarz, Yannis Pantazis, Nagaraj Adiga, Alistair Conkie, Ramya Rasipuram, Ladan Golipour, Jiangchuan Li, Tuomo Raitio, Sivanand Achanta, Maxwell Jordan, and Srikanth Vishnubthotla.

7. References

[1] J. Shen, R. Pang, R. Weiss, et al., "Natural tts synthesis by conditioning wavenet on mel spectrogram predictions," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018.

[2] A. Oord van den, S. Dieleman, H. Zen, et al., "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[3] W.Ping, K. Peng, A. Gibiansky, et al., "Deep voice 3: 2000-speaker neural text-to-speech," in *International Conference on Learning Representations*, 2018.

[4] Y. Wang, RJ Skerry-Ryan, D. Stanton, et al., "Tacotron: Towards end-to-end speech synthesis," *arXiv preprint arXiv:1703.10135*, 2017.

[5] S. Arik, M. Chrzanowski, A. Coates, G. Diamos, et al., "Deep voice: Real-time neural text-to-speech," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017.

[6] A. Gibiansky, S. Arik, G. Diamos, J. Miller, K. Peng, et al., "Deep voice 2: Multi-speaker neural text-to-speech," in *Advances in neural information processing systems*, 2017.

[7] A. Oord, Y. Li, I. Babuschkin, K. Simonyan, O. Vinyals, K. Kavukcuoglu, et al., "Parallel wavenet: Fast high-fidelity speech synthesis," *arXiv preprint arXiv:1711.10433*, 2017.

[8] N. Kalchbrenner, E. Elsen, K. Simonyan, S. Noury, N. Casagrande, E. Lockhart, F. Stimberg, A. Oord, S. Dieleman, and K. Kavukcuoglu, "Efficient neural audio synthesis," *arXiv preprint arXiv:1802.08435*, 2018.

[9] A. Mansikkaniemi, P. Smit, M. Kurimo, et al., "Automatic construction of the finnish parliament speech corpus.," in *Conference of the International Speech Communication Association(Interspeech)*, 2017.

[10] S. King, J. Crumlish, A. Martin, and L. Wihlborg, "The blizzard challenge 2018," in *Proc. Blizzard Challenge*, 2018.

[11] S. King and V. Karaiskos, "The blizzard challenge," in *Proc. Blizzard Challenge*, 2011.

[12] N. Campbell, "Towards conversational speech synthesis; lessons learned from the expressive speech processing project.," *ISCA Speech Synthesis Workshop*, 2007.

[13] Y. Chen, Y. Assael, B. Shillingford, et al., "Sample efficient adaptive text-to-speech," in *International Conference on Learning Representations*, 2019.

[14] W. Hsu, Y Zhang, R. Weiss, et al., "Disentangling correlated speaker and noise for speech synthesis via data augmentation and adversarial factorization," in *Advances in Neural Information Processing Systems*, 2018.

[15] Y. Jia, Y. Zhang, R. Weiss, et al., "Transfer learning from speaker verification to multispeaker text-to-speech synthesis," in *Advances in Neural Information Processing Systems*, 2018.

[16] T. Capes, P. Coles, A. Conkie, et al., "Siri on-device deep learning-guided unit selection text-to-speech system.," in *Conference of the International Speech Communication Association(Interspeech)*, 2017.

[17] S. Arik, J. Chen, K. Peng, W. Ping, and Y. Zhou, "Neural voice cloning with a few samples," in *Advances in Neural Information Processing Systems*, 2018.

[18] J. Latorre, J. Lachowicz, J. Lorenzo-Trueba, et al., "Effect of data reduction on sequence-to-sequence neural tts," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019.

[19] Y. Chung, Y. Wang, W. Hsu, et al., "Semi-supervised training for improving data efficiency in end-to-end speech synthesis," *arXiv preprint arXiv:1808.10128*, 2018.

[20] U. Reichel and F. Schiel, "Using morphology and phoneme history to improve grapheme-to-phoneme conversion," in *Ninth European Conference on Speech Communication and Technology*, 2005.

[21] R. Sproat, "Multilingual text analysis for text-to-speech synthesis," *Natural Language Engineering*, 1996.

[22] D. Krueger, T. Maharaj, J. Krama, M. Pezeshki, N. Ballas, N. Ke, A. Goyal, Y. Bengio, et al., "Zoneout: Regularizing rnns by randomly preserving hidden activations," in *International Conference on Learning Representations*, 2017.

[23] J.K.Chorowski, D.Bahdanau, D.Serdyuk, K.Cho, Y.Bengio, et al., "Attention-based models for speech recognition," in *Advances in Neural Information Processing Systems*, 2015.

[24] Y. Wang, D. Stanton, Y. Zhang, et al., "Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis," *arXiv preprint arXiv:1803.09017*, 2018.

[25] R. Skerry-Ryan, E. Battenberg, Y. Xiao, Yuxuan Wang, et al., "Towards end-to-end prosody transfer for expressive speech synthesis with tacotron," *arXiv preprint arXiv:1803.09047*, 2018.

[26] R. Doddipatla, N. Braunschweiler, and R. Maia, "Speaker adaptation in dnn-based speech synthesis using d-vectors.," in *INTERSPEECH*, 2017.

[27] E. Marchi, S. Shum, K. Hwang, et al., "Generalised discriminative transform via curriculum learning for speaker recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018.

[28] Siri Team, "Personalized hey siri," *Apple Machine Learning Journal*, vol. 1, April 2018.

[29] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proceedings of the 26th annual international conference on machine learning*. ACM, 2009.

[30] R. Vilalta and Y. Drissi, "A perspective view and survey of meta-learning," *Artificial intelligence review*, 2002.

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

[32] Siri Team, "Deep learning for siri voice," *Apple Machine Learning Journal*, vol. 1, August 2017.

[33] R. Streijl, S. Winkler, and D. Hands, "Mean opinion score (mos) revisited: methods and applications, limitations and alternatives," *Multimedia Systems*, 2016.

[34] T. Hayashi, A. Tamamori, K. Kobayashi, K. Takeda, and T. Toda, "An investigation of multi-speaker training for wavenet vocoder," in *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2017.

[35] N. Adiga, V. Tsiaras, and Y. Stylianou, "On the use of wavenet as a statistical vocoder," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018.