Improving RNN-T for Domain Scaling Using Semi-Supervised Training with Neural TTS

Yan Deng[1], *Rui Zhao*[2], *Zhong Meng*[2], *Xie Chen*[2], *Bing Liu*[1], *Jinyu Li*[2], *Yifan Gong*[2] *and Lei He*[1]

[1]Microsoft, China
[2]Microsoft, USA

{yaden, ruzhao, zhme, xieche, liu.bing, jinyli, ygong, helei}@microsoft.com

Abstract

Recurrent neural network transducer (RNN-T) has shown to be comparable with conventional hybrid model for speech recognition. However, there is still a challenge in out-of-domain scenarios with context or words different from training data. In this paper, we explore the semi-supervised training which optimizes RNN-T jointly with neural text-to-speech (TTS) to better generalize to new domains using domain-specific text data. We apply the method to two tasks: one with out-of-domain context and the other with significant out-of-vocabulary (OOV) words. The results show that the proposed method significantly improves the recognition accuracy in both tasks, resulting in 61.4% and 53.8% relative word error rate (WER) reductions respectively, from a well-trained RNN-T with 65 thousand hours of training data. We do further study on the semi-supervised training methodology: 1) which modules of RNN-T model to be updated; 2) the impact of using different neural TTS models; 3) the performance of using text with different relevancy to target domain. Finally, we compare several RNN-T customization methods, and conclude that semi-supervised training with neural TTS is comparable and complementary with Internal Language Model Estimation (ILME) or biasing.

Index Terms: RNN-T, customization, semi-supervised training, neural TTS

1. Introduction

End-to-end (E2E) models, which adopt a unified neural network to learn mapping between speech and word sequences, have been widely used for automatic speech recognition (ASR). In recent years, significant progress has been made for E2E models [1, 2, 3, 4, 5, 6]. Among these models, RNN-T is very promising to replace conventional hybrid models due to its streaming nature. It has been shown that RNN-T can be optimized to surpass a well-trained hybrid model for large-scale ASR in real scenarios, in terms of both accuracy and latency [7, 8].

A main challenge for RNN-T is how to deploy the model into a new domain which has only text data, as customization is relatively difficult for RNN-T which learns language model (LM) implicitly and the vocabulary is not explicitly defined. The RNN-T model suffers from performance degradation in mismatched target domain with words or phrases unseen from training data. Researchers have proposed several methods to leverage text-only data from target domain. The most straightforward method is to integrate an external language model (LM) [9, 10] or do biasing [11, 12, 13] with domain-specific text, as in traditional hybrid models. LM fusion is further improved by training with text-only data [14, 15]. The second way is to generate speech data for the text using neural TTS models and update the RNN-T model using paired speech-text data [7, 16, 17]. All these methods have shown to be helpful in adapting RNN-T model to new domains or scenarios, using only text data.

In this paper, we explore the semi-supervised training with neural TTS to leverage text-only data for RNN-T domain adaptation. Different from the method of generating speech data offline, semi-supervised training generates the acoustic features on-the-fly via integrating a well-trained neural TTS model into RNN-T training. This kind of paired data generation is faster and more flexible as no neural vocoder is needed, and it increases data diversity that can be helpful for RNN-T training. Similar methods have been used in previous studies, which built a TTS→ASR architecture to leverage large amounts of in-domain unpaired text for data augmentation [18, 19, 20]. Here, we borrow the idea to increase the customization capability of RNN-T. Our main contributions are in the following:

- We verify the effectiveness of semi-supervised training with neural TTS for domain adaptation using an RNN-T model well-trained with 65 thousand (K) hours of anonymized data. The method resolves the challenge for RNN-T in new scenarios with little performance degradation on our general test set with 1.8 million (M) words.
- We give a comprehensive study on several key points that lead to the success of semi-supervised training, including comparison of using different neural TTS models, which has not been fully investigated in previous studies.
- We combine the proposed method with ILME [10] or biasing [11] for different tasks to obtain further gains in new domains, with final ASR accuracy comparable to that of hybrid models.

To the best of our knowledge, there is few such work reporting similar significant gains for an RNN-T model pre-trained on a large-scale training corpus in domain adaptation.

The rest of the paper is organized as follows. Section 2 introduces the methodology of semi-supervised training through combining neural TTS with RNN-T. Sections 3 and 4 show the experiments and results on adapting a well-trained RNN-T to new domains. We give a brief conclusion in Section 5.

2. Methodology

In this section, we describe the semi-supervised training framework via combining neural TTS, to improve the performance of RNN-T model in new scenarios.

2.1. Semi-supervised Training

We adopt a similar architecture like TTS→ASR chain for RNN-T domain scaling, in which semi-supervised training is used to leverage text-only data from a new target domain [19, 20]. It consists of neural TTS and RNN-T as in Figure 1. The output

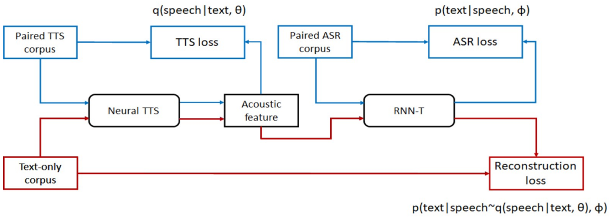

Figure 1: *Diagram of semi-supervised training with neural TTS. The red line represents data flow in training with text-only corpus, the blue line represents training with paired TTS/ASR data.*

of neural TTS is feed into RNN-T directly for joint optimization of both models.

We first pre-train both neural TTS and RNN-T models on separate general paired corpus, which includes real speech data with vast diversity in content and acoustics. Then we use the well-trained neural TTS model to generate acoustic feature for text from target domain, with a randomly selected speaker embedding. Finally, the synthetic data is combined with the paired data to further fine-tune either the RNN-T model or both neural TTS and RNN-T models using following loss function:

$$\mathcal{L} = \alpha\mathcal{L}_{TTS} + \beta\mathcal{L}_{RNN-T}^{paired} + \gamma\mathcal{L}_{RNN-T}^{unpaired} \qquad (1)$$

where $\mathcal{L}_{TTS}$ is the Transformer TTS loss defined in [21] or FastSpeech loss defined in [22], depending on which neural TTS model is used. α is set to 0 if we only update the RNN-T model. $\mathcal{L}_{RNN-T}^{paired}$ is actually the loss used in RNN-T modeling [1]. And for $\mathcal{L}_{RNN-T}^{unpaired}$, the format is the same as $\mathcal{L}_{RNN-T}^{paired}$, the only difference is that input speech features are generated by a neural TTS model instead of extracting from real speech.

Generally, a large amount of paired speech-text data can be generated if we exhaust all possible permutations and combinations of available speakers and sentences from target domain. But it's time-consuming for offline data generation as neural vocoder is used [7, 16, 17], and some permutations may not bring meaningful diversity to improve RNN-T on the target domain. With a neural TTS that generates data on-the-fly, the combination of speakers and sentences is performed randomly in each batch, which is more effective in finding the most meaningful combinations that are beneficial for RNN-T training. We will show the superiority of our method in results part.

2.2. Neural TTS

We use a multi-speaker neural TTS model to generate acoustic feature for text-only data in semi-supervised training. The multi-speaker modeling framework is similar with [23] but no neural vocoder is used here, as only acoustic feature is needed in our training. This is much simpler than the offline data generation pipeline which converts mel-spectrogram to speech and then extracts new mel-spectrogram. In our work, the E2E model with encoder-decoder structure is adopted to learn mapping from phoneme sequences to mel-spectrograms. We try both Transformer TTS and FastSpeech [21, 22] models for training to see which one generates data with closer distribution to real speech, based on the performance of RNN-T. We give a brief description of both models in experimental setup. Different from original FastSpeech, we use the forced alignment from a pre-trained ASR model to get the duration, instead of the teacher-student training pipeline.

In multi-speaker neural TTS, a speaker embedding is extracted from an internal speaker model, which is pre-trained on 7363 speakers from VoxCeleb [24, 25]. We use different strategies to add speaker embedding to different neural TTS systems:

- Multi-speaker Transformer TTS: the speaker embedding is concatenated to the encoder output, before being fed to the decoder.
- Multi-speaker FastSpeech: besides concatenating the speaker embedding with the encoder output, we also apply conditional layer normalization to both encoder and decoder with speaker embeddings determining the scales and biases [26].

During training with paired corpus, the speaker embedding vector is extracted from the speech in each text-speech pair. For unpaired corpus with only text, we select an utterance randomly from the paired corpus to get the speaker embedding.

2.3. Adapting RNN-T to New Domain

The RNN-T model [1] consists of an encoder, a predictor and a joint network. The encoder converts an acoustic feature $x(t)$ into a high-level representation $h_{enc}(t)$. And the predictor produces another high-level representation $h_{dec}(u)$, based on the previous non-blank target $y(u-1)$ predicted by the RNN-T model. The joint network is a feed-forward network that combines the encoder and predictor output to get $z(t, u)$, which is used to calculate the final posterior of each output token, with a linear transformation followed by a softmax.

Adaptation [27] is usually applied to fit speech models to new scenarios or speakers. There have been detailed investigations on which components of the RNN-T model should be updated to get the best results for different tasks [7, 28]. Following previous studies, we adapt an RNN-T model to new domains by updating different components of RNN-T without losing performance on the general domain test set.

2.4. Training Strategy

Our semi-supervised training is implemented using the objective function defined in Eq. (1) by combining both paired and unpaired data. We use the corpus for model pre-training as paired corpus, and $\mathcal{L}_{TTS}$ is calculated for batched data from TTS corpus, $\mathcal{L}_{RNN-T}^{paired}$ is calculated for batched data from ASR corpus. The paired data is added to prevent the adapted models from straying too far away by synthetic data, which has a distribution different from that of real speech [19]. Here, the consistency loss is not used as it provides only slight improvements for a well-trained ASR model [20], which is also observed in our experiments.

In our work, we adopt the iterative training method to leverage both paired and unpaired data, that is, using paired data in one batch and unpaired data in the following batch. The switching frequency can be tuned to make a trade-off in performance between source and target domain. We compare the iterative training with other methods which mix paired and unpaired data in one batch, and find that there is no difference in performance when training converges. But our method can be used to avoid the effort of tuning weighting coefficients in Eq. (1), as it's a bit tricky to pre-define the coefficients without knowing the numerical range of each loss before training. We just use paired and unpaired data in an iterative way, letting the model learn by itself during training until converge.

There are always robustness issues in speech generated by neural TTS models, especially for Transformer TTS in which attention is used. We introduce a data filtering mechanism to remove bad data on-the-fly, based on the focus rate which measures how an attention is close to diagonal [22]. Here, we use the maximal focus rate of all heads at all layers to do filter. The threshold for filtering is obtained based on a development set. We also set a lower bound on the number of samples to be used in each batch, to avoid filtering too many utterances. The data filtering mechanism is only applied on Transformer TTS, not on FastSpeech which has few robustness issues as shown in [22].

3. Experimental Setup

3.1. Datasets

In our experiments, the RNN-T model is well-trained on 65 thousand (K) hours of transcribed Microsoft data until full convergence [7]. The neural TTS model is well-trained on about 1K hours of data including our internal TTS corpus, LibriTTS, VCTK and LJSpeech [29, 30, 31]. We evaluate the effectiveness of our method on two different test sets: one has 800 utterances from a new domain consisting of common words but with out-of-domain (OOD) context (OOD task); the other has 11K utterances from conversational data containing significant amount of out-of-vocabulary (OOV) words (OOV task). All training and test data are anonymized data with personally identifiable information removed.

The domain-specific text data are prepared based on the characteristics of different tasks. For the OOD task, the training texts are obtained by randomly parsing the grammar in the new domain and also using the crowd sourcing method as described in [32], 75K sentences are generated for training. For the OOV task, we pre-define two lists: one is an OOV word list with 2.5K names of entities, the other is a pattern list with 509 frequently used patterns. We generate in total 1.27 million (M) sentences by doing all possible permutations and combinations of the OOV words and patterns. We also enlarge the OOV word list and show the impact in results part.

3.2. Model Configurations

The baseline RNN-T model consists of a 6-layer unidirectional LSTM for the encoder and 2 layers of the same structure for the predictor. The LSTM layer has 1024 hidden units in each layer for regular-sized model and 768 hidden units with singular value decomposition (SVD) [33] for a small-sized model. The acoustic feature for the encoder is formed by stacking eight 80-dimension log Mel filter bank features calculated for every 10 milliseconds (ms) speech. The output layer in joint network models 4K word piece units.

We follow [21] for the configuration of Transformer TTS. The encoder and decoder of our multi-speaker Transformer TTS consists of 6 layers, with 512 hidden units in each layer, and we use 8 heads in each multi-head attention block. The model of multi-speaker FastSpeech follows the basic structure of the original FastSpeech [22], which consists of 6 feed-forward Transformer blocks in both phoneme encoder and mel-spectrogram decoder, the number of attention heads is set to 4. Other model configurations are the same as original papers unless otherwise stated. For all neural TTS models, the size of phoneme vocabulary is 110 and the output acoustic feature is an 80-dimension log Mel filter bank calculated for every 10ms of speech. The speaker embedding is a 512-dimension vector extracted from a well-trained speaker model, which is reduced to 128 dimensions using a linear projection layer and is then fed to either Transformer TTS or FastSpeech.

For multi-speaker Transformer TTS, we add guided attention loss [34] to stabilize alignment on training data with big diversity in acoustics and prosody. As the loss introduces strong constrain on the attention to fit a diagonal alignment, hurting prosody, we remove it after the alignment is good enough, and further fine-tune the model to get more natural synthetic speech.

Table 1: *Performance of semi-supervised training in new domains. WERR is the relative WER reduction. OOD Task has out-of-domain context; OOV Task has out-of-vocabulary words.*

Method	WER (%)	WERR (%)
OOD Task		
Baseline	16.52	
+ Semi-supervised Training	6.37	61.4
OOV Task		
Baseline	27.50	
+ Semi-supervised Training	12.70	53.8

4. Results

In this section, we conduct evaluations on two different test sets to verify the effectiveness of semi-supervised training with neural TTS: one containing out-of-domain context and the other containing OOV words. We perform beam search inference with a beam size of 5 for all evaluations. We also give detailed analyses on several key points: 1) RNN-T updating; 2) naturalness and diversity of synthetic data; 3) relevancy of training text to target domain. Finally, we do comprehensive study on different methods for customization, and combine them to get more gains for each task.

4.1. Adaptation to New Domains

In this experiment, we adopt multi-speaker Transformer TTS for synthetic data generation, and several common pronunciations are included for each OOV word. The small-sized RNN-T model is used for the OOD task, and the regular-sized model is used for the OOV task. We update the 2 upper layers of encoder, all layers of predictor and joint network during training. From results in Table 1, we observe significant improvements for both tasks, with 61.4% and 53.8% relative WER reductions for the OOD and OOV task, respectively, which is comparable or even better than previous studies [7, 16, 17, 18, 19, 20]. It shows that semi-supervised training with neural TTS is helpful in adapting RNN-T to new scenarios with only text data, even for a well-trained RNN-T model. The consistent improvements in two scenarios also show that our Transformer TTS model can generate very natural speech with vast diversity in acoustics given text with either out-of-domain context or OOV words.

We also compare different methods of synthetic data generation, and find that on-the-fly generation (WER=6.37%) leads to more gains than offline generation (WER=9.33%), as more diversity is introduced when we combine speaker embeddings and sentences randomly in one batch. Furthermore, we evaluate the regular-sized model on our general test set with 1.8M words, there is only slight degradation after using semi-supervised training, WER increases from 9.64% to 10.26%.

4.2. Method Analysis

We first conduct ablation study in OOV task to verify the effectiveness of each module when updating RNN-T with synthetic domain-specific data. Table 2 lists the results of fixing

Table 2: *Comparison of adapting different modules and their combinations, **only the 2 upper layers of encoder** are updated in all experiments. Y: update, N: fixed.*

Encoder	Predictor	Joint	WER (%)
N	N	N	27.50
Y	Y	Y	12.70
Y	N	Y	15.07
Y	N	N	18.33
N	Y	Y	19.88
N	N	Y	16.83
N	Y	N	26.41

Table 3: *Comparison of using different neural TTS models.*

System	WER (%)
Baseline	27.50
Transformer TTS + RNN-T	12.70
FastSpeech + RNN-T	14.59

each component during adaption. From Table 2, we can see that fixing more components leads to decrease in performance, especially when we fix encoder or joint network. But there is an exception that updating predictor & joint network (19.88%) performs worse than updating only joint network (16.83%). This could be attributed to the training text being generated using pre-defined patterns, which makes the context in training and test not fully matched.

Then we analyse if a better neural TTS, which generates more natural speech with larger diversity, can help reduce the mismatch between synthetic and real speech. We adopt Transformer TTS and FastSpeech for comparison. Table 3 lists the WER results in OOV task. It can be seen that better performance is obtained when using Transformer TTS, about 12.9% relative WER reduction compared with using FastSpeech. This demonstrates that attention mechanism contributes to more flexible duration prediction, which helps generating speech with larger diversity that is closer to real speech. Although there are a few attention errors, we can remove such data using the proposed filtering mechanism.

We further study the performance of using text corpus covering different number of OOV words for the OOV task. The newly added OOV words are selected randomly from a very large pool. Table 4 shows that less gain is obtained when we add more OOV words that are irrelevant to test set. There is only an 11% relative WER reduction when we use 1M OOV words, which is enlarged by about 400 times to have very low relevancy ratio. We can draw the conclusion that text with high relevancy (the percentage of sentences which include OOV words that occur in test set) is crucial for our semi-supervised training.

4.3. Comprehensive Study on Customization Methods

Several methods have been proposed to improve E2E models for customization, such as ILME [10] or biasing [16]. We compare the effectiveness of different methods and combine them to get more gains in target domain. For the OOD task which includes out-of-domain context, we apply semi-supervised training, ILME and splicing data, where splicing data means generating paired data by searching the speech segment of each word in general training corpus and concatenating them to form new utterances in the target domain [32]. For OOV task which includes OOV words, we add biasing that is similar as [35].

Table 4: *Comparison of using text corpus covering different number of OOV words (names of entities). The relevant OOV words are heavily diluted with many irrelevant OOVs added.*

# of OOV Words	WER (%)
2.5K	12.70
10K	16.38
50K	19.96
200K	22.97
1M	24.47

Table 5: *Comparison & combination of different customization methods. WERR is the relative WER reduction. OOD Task has out-of-domain context; OOV Task has out-of-vocabulary words.*

Method	WER (%)	WERR (%)
OOD Task		
Baseline	16.52	
+ Semi-supervised Training	6.37	61.4
+ ILME	5.71	65.4
+ Splicing Data	5.02	69.2
+ Semi-supervised Training	4.33	73.8
+ ILME	3.74	77.4
OOV Task		
Baseline	27.50	
+ Semi-supervised Training	12.70	53.8
+ Biasing	11.30	58.9

All results are shown in Table 5. We have several observations: 1) semi-supervised training is comparable to ILME (here we use much less text data for training than ILME, 75K VS 10M). 2) both semi-supervised training and ILME are worse than using splicing data, which demonstrates that using real paired data is the most effective method. However, this method is not always feasible especially when OOV words exist. 3) the combination of all methods by adding one after another can achieve more than 10% relative WER reduction from each method, which demonstrates that the semi-supervised training is complementary to others via bringing in more meaningful diversity that is helpful for RNN-T in target domain.

5. Conclusions

In this paper, we have demonstrated that using semi-supervised training with neural TTS is an effective way to improve RNN-T performance in new domains that only text data are available. There are relative WER reductions of 61.4% and 53.8% for the tasks with out-of-domain context and OOV words, respectively. We further investigate some key reasons leading to the significant improvements. We find that the quality of neural TTS model and the relevancy of text to target domain are most important. We finally compare semi-supervised training with other customization methods, and get more gains after combining all of them, about 77.4% relative WER reduction for the task with out-of-domain context, and 58.9% relative WER reduction for the task with OOV words. In the future, we will further improve our neural TTS on larger corpus with more acoustic and prosody diversity, and try to combine different TTS models for multi-agent joint training.

6. References

[1] A. Graves, "Sequence transduction with recurrent neural networks," in *International Conference of Machine Learning (ICML) 2012 Workshop on Representation Learning*, Edinburgh, Scotland, Jun./Jul. 2012.

[2] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *Proc. ICASSP*, Shanghai, China, Mar. 2016, pp. 4960–4964.

[3] E. Battenberg, J. Chen, R. Child, and et al., "Exploring neural transducers for end-to-end speech recognition," in *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, Okinawa, Japan, Dec. 2017, pp. 206–213.

[4] K. Rao, H. Sak, and R. Prabhavalkar, "Exploring architectures, data and units for streaming end-to-end speech recognition with RNN-transducer," in *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, Okinawa, Japan, Dec. 2017.

[5] C.-C. Chiu, T. N. Sainath, Y. Wu, and et al., "State-of-the-art speech recognition with sequence-to-sequence models," in *Proc. ICASSP*, Calgary, AB, Canada, Apr. 2018.

[6] J. Li, Y. Wu, Y. Gaur, C. Wang, R. Zhao, and S. Liu, "On the comparison of popular end-to-end models for large scale speech recognition," in *Proc. Interspeech*, Shanghai, China, Oct. 2020, pp. 1–5.

[7] J. Li, R. Zhao, Z. Meng, Y. Liu, W. Wei, S. Parthasarathy, V. Mazalov, Z. Wang, L. He, S. Zhao, and Y. Gong, "Developing RNN-T models surpassing high-performance hybrid models with customization capability," in *Proc. Interspeech*, Shanghai, China, Oct. 2020, pp. 3590–3594.

[8] T. N. Sainath, Y. He, B. Li, and et al., "A streaming on-device end-to-end model surpassing server-side conventional model quality and latency," in *Proc. ICASSP*, Barcelona, Spain, May. 2020, pp. 6059–6063.

[9] C. Gulcehre, O. Firat, K. Xu, and et al., "On using monolingual corpora in neural machine translation," *arXiv preprint arXiv:1503.03535*, 2015.

[10] Z. Meng, S. Parthasarathy, E. Sun, Y. Gaur, N. Kanda, L. Lu, X. Chen, R. Zhao, J. Li, and Y. Gong, "Internal language model estimation for domain-adaptive end-to-end speech recognition," in *2021 IEEE Spoken Language Technology Workshop (SLT)*, Shenzhen, China, Jan. 2021.

[11] K. Hall, E. Cho, C. Allauzen, and et al., "Composition-based on-the-fly rescoring for salient n-gram biasing," in *Proc. Interspeech*, Dresden, Germany, Sep. 2015, pp. 1418–1422.

[12] P. Aleksic, C. Allauzen, D. Elson, and et al., "Improved recognition of contact names in voice commands," in *Proc. ICASSP*, South Brisbane, QLD, Australia, Apr. 2015, pp. 5172–5175.

[13] I. McGraw, R. Prabhavalkar, R. Alvarez, and et al., "Personalized speech recognition on mobile devices," in *Proc. ICASSP*, Shanghai, China, Mar. 2016, pp. 5955–5959.

[14] Z. Meng, N. Kanda, Y. Gaur *et al.*, "Internal language model training for domain-adaptive end-to-end speech recognition," in *Proc. ICASSP*, Toronto, Ontario, Canada, Jun. 2021.

[15] Z. Meng, Y. Wu, N. Kanda, L. Lu, X. Chen, G. Ye, E. Sun, J. Li, and Y. Gong, "Minimum word error rate training with language model fusion for end-to-end speech recognition," *arXiv preprint arXiv:2106.02302*, 2021.

[16] K. C. Sim, F. Beaufays, A. Benard, and et al., "Personalization of end-to-end speech recognition on mobile devices for named entities," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, Sentosa, Singapore, Dec. 2019.

[17] X. Zheng, Y. Liu, D. Gunceler, and D. Willett, "Using synthetic audio to improve the recognition of out-of-vocabulary words in end-to-end ASR systems," in *Proc. ICASSP*, Toronto, Ontario, Canada, Jun. 2021.

[18] M. K. Baskar, S. Watanabe, R. Astudillo, and et al., "Semi-supervised sequence-to-sequence ASR using unpaired speech and text," in *Proc. Interspeech*, Graz, Austria, Sep. 2019, pp. 3790–3794.

[19] G. Wang, A. Rosenberg, Z. Chen, and et al., "Improving speech recognition using consistent predictions on synthesized speech," in *Proc. ICASSP*, Barcelona, Spain, May. 2020.

[20] Z. Chen, A. Rosenberg, Y. Zhang, and et al., "Improving speech recognition using GAN-based speech synthesis and contrastive unspoken text selection," in *Proc. Interspeech*, Shanghai, China, Oct. 2020, pp. 556–560.

[21] N. Li, S. Liu, Y. Liu, S. Zhao, M. Liu, and M. Zhou, "Neural speech synthesis with transformer network," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, Honolulu, Hawaii, USA, Feb. 2019, pp. 6706–6713.

[22] 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 32 (NeurIPS 2019)*, Vancouver CANADA, Dec. 2019, pp. 3171–3180.

[23] Y. Deng, L. He, and F. Soong, "Modeling multi-speaker latent space to improve neural tts: Quick enrolling new speaker and enhancing premium voice," *arXiv preprint arXiv:1812.05253*, 2018.

[24] A. Nagrani, J. S. Chung, and A. Zisserman, "VoxCeleb: a large-scale speaker identification dataset," in *Proc. Interspeech*, Stockholm, Sweden, Aug. 2017, pp. 2616–2620.

[25] J. S. Chung, A. Nagrani, and A. Zisserman, "VoxCeleb2: Deep speaker recognition," in *Proc. Interspeech*, Hyderabad, India, Sep. 2018, pp. 1086–1090.

[26] 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 *2021 International Conference on Learning Representations (ICLR)*, Vienna, Austria, May. 2021.

[27] P. Bell, J. Fainberg, O. Klejch, J. Li, S. Renals, and P. Swietojanski, "Adaptation algorithms for neural network-based speech recognition: An overview," *IEEE Open Journal of Singal Processing*, vol. 2, 2021.

[28] Y. Huang, J. Li, L. He, W. Wei, W. Gale, and Y. Gong, "Rapid RNN-T adaptation using personalized speech synthesis and neural language generator," in *Proc. Interspeech*, Shanghai, China, Oct. 2020, pp. 1256–1260.

[29] H. Zen, V. Dang, R. Clark, and et al., "LibriTTS: A corpus derived from LibriSpeech for text-to-speech," in *Proc. Interspeech*, Graz, Austria, Sep. 2019, pp. 1526–1530.

[30] Y. Junichi, V. Christophe, MacDonald, and Kirsten, "CSTR VCTK corpus: English multi-speaker corpus for CSTR voice cloning toolkit (version 0.92)," *https://doi.org/10.7488/ds/2645*, 2019.

[31] K. Ito, "The LJ speech dataset," *https://keithito.com/LJ-Speech-Dataset/*, 2017.

[32] R. Zhao, J. Xue, J. Li, W. Wei, L. He, and Y. Gong, "On addressing practical challenges for RNN-Transducer," in *submit to Interspeech*, 2021.

[33] J. Xue, J. Li, and Y. Gong, "Restructuring of deep neural network acoustic models with singular value decomposition." in *Proc. Interspeech*, Lyon, France, Aug. 2013, pp. 2365–2369.

[34] H. Tachibana, K. Uenoyama, and S. Aihara, "Efficiently trainable text-to-speech system based on deep convolutional networks with guided attention," in *Proc. ICASSP*, Calgary, AB, Canada, Apr. 2018.

[35] P. Aleksic, M. Ghodsi, A. Michaely, and et al., "Bringing contextual information to google speech recognition," in *Proc. Interspeech*, Dresden, Germany, Sep. 2015.