Rapid RNN-T Adaptation Using Personalized Speech Synthesis and Neural Language Generator

Yan Huang, Jinyu Li, Lei He, Wenning Wei, William Gale, and Yifan Gong

Microsoft Corporation

{yanhuang, jinyli, helei, wennwei, wigale,ygong}@microsoft.com

Abstract

Rapid unsupervised speaker adaptation in an E2E system posits us new challenges due to its end-to-end unified structure in addition to its intrinsic difficulty of data sparsity and imperfect label [1]. Previously we proposed utilizing the content relevant personalized speech synthesis for rapid speaker adaptation and achieved significant performance breakthrough in a hybrid system [2]. In this paper, we answer the following two questions: First, how to effectively perform rapid speaker adaptation in an RNN-T. Second, whether our previously proposed approach is still beneficial for the RNN-T and what are the modification and distinct observations. We apply the proposed methodology to a speaker adaptation task in a state-of-art presentation transcription RNN-T system. In the 1 min setup, it yields 11.58 % or 7.95 % relative word error rate (WER) reduction for the sup/unsup adaptation, comparing to the negligible gain when adapting with 1 min source speech. In the 10 min setup, it yields 15.71 % or 8.00 % relative WER reduction, doubling the gain of the source speech adaptation. We further apply various data filtering techniques and significantly bridge the gap between sup/unsup adaptation.

Index Terms: rapid speaker adaptation, unsupervised adaptation, RNN-T, personalization

1. Introduction

End-to-end models (E2E) adopting a unified framework with joint optimization has made significant progress in recent years [3–9]. Among the various forms of E2E models [10–14], RNN Transducer (RNN-T) [11] has gained popularity and been developed extensively due to its convenient streaming [7, 8]. Personalization is a widely practiced strategy in industry speech recognition systems [1, 15, 16]. Personalizing an E2E system [16–19] posits us new challenges due to its end-to-end unified structure in addition to the intrinsic difficulty of the data sparsity and imperfect label [1].

Rapid speaker adaptation refers to adapting a speech model to a specific speaker with limited data (e.g. less than 10 min). It has been studied substantially in the hybrid systems [20–26]. Previously we proposed utilizing the personalized speech synthesis and neural language generator for rapid speaker adaptation and achieved significant performance gain in a hybrid system [2]. In this paper, we would like to answer the following two questions: First, how to effectively perform rapid adaptation in an RNN-T. Second, whether our previously proposed methodology is still beneficial in the RNN-T and what are the specific modification and distinct observations.

We first compare different adaptation architectures in the RNN-T. We found that adapting the encoder network performs significantly better than the prediction network, consistent with [17, 27]. In the supervised setup adapting the encoder, joint, and softmax yields the best performance, while in the unsupervised setup it is better to remove the softmax adaptation to alleviate its sensitivity to the labeling errors.

We then apply our previously proposed rapid speaker adaptation using content relevant synthesized personalized speech [2] to the RNN-T. Through leveraging the speaker trait distilled from small amount of source speech and the general phonological and morphological information embedded in the synthesis model and the neural language generator, this approach fundamentally alleviates the data sparsity. Furthermore, when the synthesized personalized speech is consumed for adaptation, the original unsupervised adaptation is effectively converted to a pseudo-supervised one as the synthesized speech seldom exhibits perceptible mismatch with the input text. The imperfect label in the source speech is often rendered as less perceptible spectrum distortion in the synthesized speech.

In a state-of-art presentation transcription RNN-T system, with 1 min source speech, our proposed approach yields 11.58 % or 7.95 % relative word error rate (WER) reduction for the supervised and unsupervised adaptation respectively, while adapting with the 1 min source speech only yields negligible gain. In the 10 min setup, it yields 15.71 % or 8.00 % relative WER reduction, roughly doubling the gain of the 10 min source speech adaptation. To further improve the unsupervised adaptation, we apply various data filtering to remove the source speech possibly mislabeled and the TTS speech of lesser quality, achieving 10.46 % and 11.73 % relative WER reduction for the 1 min and 10 min setup .

To the best of our knowledge, we are not aware of any previous work reporting similar significant gain in rapid unsupervised RNN-T adaptation, especially in the 1 min adaptation setup. We also compare with applying the similar approach to the hybrid model and discuss the distinct observations.

The rest of this paper is organized as: Section 2 introduces the methodology; Section 3 presents the experiments and results; Section 4 concludes the paper.

2. Methodology

In this section, we describe our proposed RNN-T adaptation using personalized synthesis and neural language generator.

2.1. System Architecture

We adopt a similar architecture we proposed previously for the hybrid model adaptation [2]. It consists of personalized synthesis, neural language generator, and RNN-T adaption as in Fig 1.

We first use small amount of source speech to train the personalized text-to-speech (TTS) model through model refinement; then use the neural language generator to generate content relevant text to be synthesized. Alternatively, random conversational speech text can be used. Lastly, the synthesized speech is added to the source speech for adaptation. In the supervised

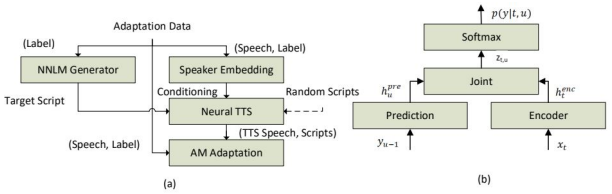

Figure 1: *(a) System architecture; (b) RNN-T model structure.*

setup, human transcription of the source speech is used for the personalized TTS training, content relevant text generation, and model adaptation. Otherwise, in the unsupervised setup, the first-pass decoding result is used throughout the pipeline.

2.2. RNN-T adaptation

RNN-T models the acoustic, language, pronunciation, and acoustic-language score fusion within an end-to-end framework. Rapid adaptation with as little as 1 min speech in the RNN-T is extremely prone to overfitting. Therefore identifying the key network component for adaptation is critical.

An RNN-T consists of an encoder, a prediction, and a joint network as in Fig 1 (b). The encoder network converts the acoustic feature x_t into a high-level representation h_t^{enc}, where t is the index of time:

$$h_t^{enc} = f^{enc}(x_t). \tag{1}$$

The prediction network generates a high-level representation h_{pre}^u by conditioning on the previous non-blank target y_{u-1} predicted by the RNN-T, where u is the index of the label:

$$h_u^{pre} = f^{pre}(y_{u-1}). \tag{2}$$

The joint network $z_{t,u}$ is a feed-forward network that combines the encoder output h_{enc}^t and the prediction output h_u^{pre}:

$$z_{t,u} = f_{joint}(h_t^{enc}, h_u^{pre}). \tag{3}$$

$z_{t,u}$ is connected to the output layer with a linear transform followed by a softmax. The posterior of each output token k is

$$P(k|t,u) = softmax(W_y z_{t,u}^k + b_y). \tag{4}$$

The modeling of acoustics including the speaker voice trait largely resides in the encoder. The prediction network, generally believed to carry primarily language-level information or simply handling speech alignment according to a recent study [27] , is expected to be less relevant in rapid adaptation with limited amount of adaptation data. The joint network specifies the fusion of acoustic and language information, which can also potentially be optimized per speaker basis. The softmax layer with the linear projection directly modeling the posterior of the word piece unit can be efficiently and effectively adapted, though it is expected to be most sensitive to labeling errors.

Regularization is usually applied to address overfitting in rapid adaptation [16, 20, 28]. We experiment with different regularization approaches, but found its limited impact especially for this work where large amount of TTS speech is added.

2.3. Personalized Speech Synthesis

We use the same multi-speaker neural TTS system for personalized speech synthesis [2]. As depicted in Figure 2, it consists of a spectrum predictor, a neural vocoder, and a speaker embedding. The spectrum predictor converts the input text to the Mel

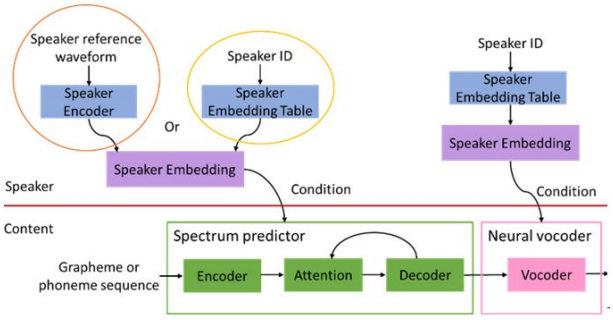

Figure 2: *Diagram of personalized speech synthesis.*

spectral using an encoder-decoder with attention model. The vocoder generates the waveform conditioning on the Mel spectral using a WaveNet neural vocoder [29]. The speaker embedding is introduced to model multi speaker latent space and concatenated with the encoder output as input to the attention layer, then jointly trained with the spectrum predictor [30]. We use an in-house TTS corpus with 30 professional en-US speakers and more than 200 hours phonetic rich recordings for training. The spectrum predictor is adapted to each target speaker with a new speaker embedding optimized to the enrollment data, while a universal WaveNet model is adopted without adaptation.

One particular challenge is that, in unsupervised adaptation, only the first-pass decoding result is available. The imperfect transcription may affect the personalized TTS model training. Furthermore, rapid adaptation with as little as 1 min speech makes robust estimation of personalized TTS more challenging.

2.4. Neural Language Generator

We use an LSTM language model [31] with a beam search algorithm [32] to generate content relevant target text as in [2]. Each sentence is provided as a prompt to the neural language generator to generate various continuations of the prompts. Similar to [33], we impose diversity constraints during the beam search, namely by penalizing repeated tokens, restricting the number of beams that end with the same bigram, and preventing n-gram repetitions within a beam. The language model has a vocabulary size of 59K BPE tokens [33] and three LSTM layers, with a total of 220M parameters. We trained the language model to convergence on 3B words of paragraph level web-crawled data.

3. Experiments and Results

We present experiments in a presentation transcription system. All experiments were conducted on anonymized data with personally identifiable information removed.

3.1. Experimental Setup

The baseline RNN-T is trained from 65K hour anonymized speech with the cross-entropy criteria. It consists of six layer-normalized LSTM layers for the encoder and two layers of the same structure for the prediction network. Each LSTM layer has 1600 hidden units and the output size is reduced to 800 using a linear projection layer. The acoustic feature is formed by stacking three 80-dimension log Mel filter bank calculated for every 10 millisecond speech. The output layer models 4000 word pieces plus an additional blank label.

The speaker adaptation task consists of six speakers, each with 10 min for training and 20 min for testing. As in [2], we configured 8 setups, specified by the source data amount (1 or

10 min), source label type (human or ASR), and TTS script type (random or target).

3.2. RNN-T Model Adaptation

We compare adapting different components of the RNN-T and various combination of these components. Table 1 presents the 1 min and 10 min supervised adaptation results.

For the 10 min setup, the encoder adaptation is much more effective than adapting other component of the network. Adding the joint and subsequently the softmax layer adaptation yields incrementally more gains. Nevertheless, further adding the prediction network to allow the full model to be adapted does not yield further improvement. 1 min setup presents sever data scarcity challenge with small gain only for the encoder adaptation. We also experiment with different regularization methodologies. They are found to be slightly beneficial in extreme data scarcity and in full model adaptation. When using large amount of TTS speech for adaptation, regularization only has limited benefit and therefore is not adopted in this paper.

Table 1: *Comparison of supervised adaptation results of adapting the encoder (E), predictor (P), joint (J), softmax (S), and various combination of these components, including the full network (ALL). WER.R refers to the relative WER reduction.*

Model	1 min	**WER.R**	10 min	**WER.R**
baseline	14.15	**NA**	14.15	**NA**
E	13.91	**1.68**	13.43	**5.08**
P	14.16	**-0.07**	14.14	**0.06**
J	14.15	**-0.01**	14.24	**-0.62**
S	14.27	**-0.82**	14.05	**0.74**
$E+J$	14.10	**0.35**	13.39	**5.36**
$E+J+S$	14.29	**-0.97**	13.34	**5.71**
ALL	14.31	**-1.13**	13.51	**4.55**

We select four representative structures as presented in Table 2 and proceed to unsupervised adaptation. The encoder and joint adaptation ($E+J$) performs the best with no further gain observed when adding the softmax in unsupervised adaptation. The softmax layer directly modeling the word-piece target is particularly sensitive to the labeling error. We therefore choose the encoder, joint, and softmax adaptation ($E+J+S$) as the default for the supervised and the encoder and joint adaptation ($E+J$) for the unsupervised adaptation for the rest of this paper.

Table 2: *Comparison of 10 min sup/unsup adaptation for four selected structures: predictor (P), encoder + joint ($E+J$), encoder + joint + softmax ($E+J+S$), full network (ALL).*

Model	SUP	**WER.R**	UNSUP	**WER.R**
baseline	14.15	**NA**	14.15	**NA**
P	14.14	**0.06**	14.25	**-0.71**
$E+J$	13.39	**5.36**	13.51	**4.55**
$E+J+S$	13.34	**5.71**	13.66	**3.44**
ALL	13.51	**4.55**	13.63	**3.67**

3.3. Adaptation with Personalized TTS and NNLM

Table 3 presents the RNN-T adaptation with personalized TTS and NNLM generator. We focus on the target script and leave the comparison with the random script in Section 3.4.

For the supervised setup, 1 min source speech adaptation can barely yield any gain. After adopting our proposed methodology, adapting with 100 min content relevant personalized synthesis speech yields 5.91 % relative WER reduction. The 1 min source speech is translated into large amount of synthesized speech, incorporating the speaker voice trait from the source speech with the general phonological and morphological information embedded in the TTS and neural language model. Blending in the 1 min source speech with TTS speech results in additional gain. This suggests that the source speech is still particularly valuable. We subsequently introduce weighting to boost the representativeness of the source speech, yielding 10.75 % and 11.25 % relative WER reduction with 100 min and 200 min synthesized speech. The weight is set to roughly balance the source/TTS data amount. In the 10 min setup, we observe similar pattern. Adapting with 200 min TTS speech and 10 min source speech yields 15.71 % relative WER reduction, comparing to 5.71 % for the 10 min source speech adaptation.

Table 3: *Performance of 1 min and 10 min sup/unsup adaptation with synthesis speech. (T) and (R) refer to the target or random text; (W) refers to applying weighting to original speech; + refers to adding TTS speech to the source for adaptation. (f, *, *), (*, f, *), and (*, *, f) refer to applying filtering only to the source speech when used for embedding training, to the source speech when used for adaptation, or to the TTS speech.*

Model	1 min	**WER.R**	10 min	**WER.R**
baseline	14.15	**NA**	14.15	**NA**
SUP_{org}	14.29	**-0.97**	13.34	**5.71**
SUP_{R100}	14.19	**-0.27**	13.73	**2.94**
SUP_{T100}	13.31	**5.91**	13.00	**8.14**
SUP_{+T100}	13.18	**6.89**	12.74	**10.00**
$SUP_{+T100(W)}$	12.63	**10.75**	12.26	**13.38**
$SUP_{+R200(W)}$	13.15	**7.06**	12.30	**12.29**
$SUP_{+T200(W)}$	12.56	**11.25**	11.93	**15.71**
$UNSUP_{org}$	14.04	**0.75**	13.51	**4.55**
$UNSUP_{R100}$	14.36	**-1.45**	14.10	**0.33**
$UNSUP_{T100}$	13.42	**5.17**	13.41	**5.24**
$UNSUP_{+T100}$	13.47	**4.81**	13.50	**6.44**
$UNSUP_{+T100(W)}$	13.39	**5.39**	13.16	**6.97**
$UNSUP_{+R200(W)}$	14.14	**0.11**	13.37	**3.75**
$UNSUP_{+T200(W)}$	13.03	**7.95**	13.02	**8.00**
$UNSUP^{(f,*,*)}_{+T200(W)}$	12.91	**8.76**	12.80	**9.54**
$UNSUP^{(f,f,*)}_{+T200(W)}$	12.81	**9.47**	12.69	**10.32**
$UNSUP^{(f,f,f)}_{+T200(W)}$	12.78	**9.68**	12.59	**11.02**

The unsupervised adaptation exhibits similar performance pattern with generally smaller amount of gains. For example, adapting with 200 min target script synthesized personalized speech combined with weighted source speech yields 7.95 % relative WER reduction for the 1 min setup and 8.00 % for the 10 min setup, which compares to 0.75 % and 4.55 % when only using source speech for adaptation.

To further reduce the gap of sup/unsup adaptation, we use multiple system decoding to filter the source speech possibly mislabeled and the TTS speech of lesser quality. We use a hybrid system with comparable performance to generate alternative hypothesis and measure the agreement level as basis for data filtering. When applying this filtering only to the source speech for personalized TTS training (f, *, *), the unsupervised adaptation gain increases to 8.76 % and 9.54 % for the 1 min and 10 min setup respectively; when further applying it to the source speech when consumed for adaptation (f, f, *), the gain

increases to 9.47 % and 10.32 %; when in additionally applying it to TTS speech (f, f, f), the gain increases to 9.68 % and 11.02 %. Finally, we can achieve closer-to supervised adaptation performance. We also tried other approaches based on alignment score, which was not found to be as effective.

Next we will discuss how TTS script type, source data amount, label quality, TTS data amount affect the performance.

3.4. Random Text versus Target Text

Previously we found that in the hybrid model adaptation using content relevant target text for TTS yields consistent but small additional gain comparing to using the random text [2]. For the RNN-T, as shown in Table 3, adapting with the target text synthesized speech consistently outperforms using the random text and the benefit of using the content relevant script for TTS is much more significant. Despite that we don't explicitly adapt the prediction network, the end-to-end framework and the word-piece modeling in the RNN-T makes it more sensitive to the content of the adaptation data.

3.5. Source Speech Data Amount

The source data amount can affect the adaptation in two ways: first, it determines whether personalized speech synthesis model can be robustly estimated and thus affects the TTS quality; second, it directly affects the performance when consumed for RNN-T adaptation. Generally more source speech is expected to result in larger gain. Practically we need to find a good operating point as a trade-off to minimize the user enrollment effort. As presented in Fig. 3, with as little as 0.5 min source speech, the synthesized personalized speech already starts to yield improved performance. Increasing source speech benefits both sup/unsup adaptation, but the gain is notably larger for supervised adaptation. Understandably without the ground truth label, increasing source data amount is not as valuable. Comparing adapting with TTS speech only or with the source and TTS speech, we find that the gain primarily comes from the direct consumption of the increased amount of source speech for RNN-T adaptation. The personalized TTS can benefit from more source speech, but only moderately in comparison.

3.6. Source Speech Label Quality

The imperfect label results in poor personalized TTS model estimation and thus degrades the quality of synthesis speech. When consuming the source speech for RNN-T adaptation, the incorrect source data label can generate catastrophic gradient update and result in poor adaptation performance. We simulate transcription at different quality levels. The human transcription is empirically treated as the golden standard with 0.00 % WER. As presented in Fig. 3 (b), the adaptation performance degrades as the source speech label becomes less accurate. The personalized TTS training is robust to minor labeling errors (e.g below 10 %). As the label quality continues to degrade surpassing a certain limit, we observe sharp performance degradation. This also explains why it is important to perform effective data filtering, especially for speakers with higher WERs.

3.7. TTS Data Amount

Fig. 3 (c-d) presents sup/unsup adaptation with increased amount of TTS data. No data filtering is applied for unsupervised adaptation. In all cases, the adaptation performance improves as more TTS data is added, with generally larger slope for the 1 min setup (v.s. 10 min), for the supervised setup (v.s. supervised), and using the target script (v.s. random). The performance is plateaued earlier for the 1 min setup after adding 100 min TTS speech. After applying weighting to the source speech, all cases continue to improve without being plateaued with 200 min TTS speech. We observe that the rate of improvement becomes smaller when further adding more TTS speech.

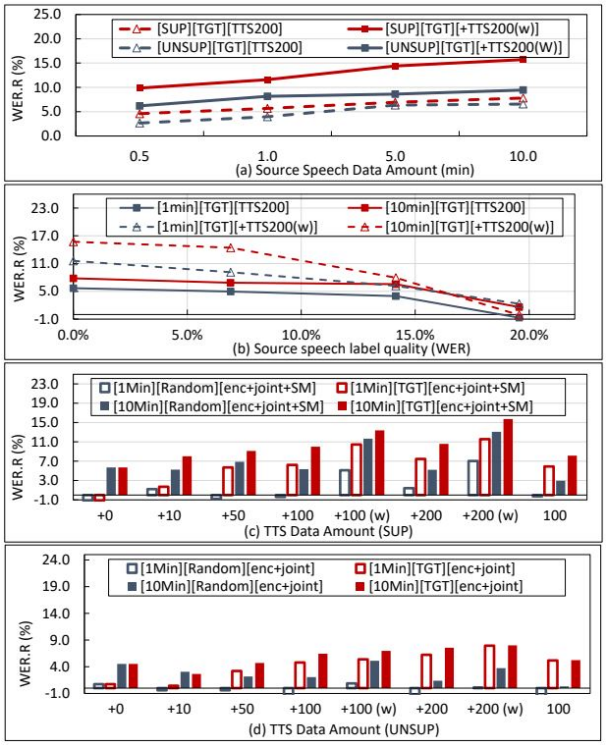

Figure 3: *(a) Adaptation performance with respect to source data amount; (b) source label quality; (c-d) TTS data amount.*

3.8. Comparison with Hybrid Model

Finally we compare with applying it to the hybrid model [2]. The RNN-T and the hybrid model have comparable baseline performance[34]. Synthesizing content relevant speech is critical for its success in the RNN-T despite the fact that the prediction network is not adapted, while it is less critical in the hybrid model. RNN-T is more sensitive to the labeling error of the source speech and the spectrum distortion in the TTS speech, while the hybrid model appears to be less impacted. Overall, the proposed rapid speaker adaptation yields significant gain on both models. With careful choice of content relevant text and effective source/TTS data selection, we can harvest even greater gain in the RNN-T.

4. Conclusions

In conclusion, we proposed an effective rapid RNN-T adaptation methodology using personalized synthesis and NNLM generator. By leveraging the speaker trait distilled from the source speech and the general phonological and morphological information from the TTS and neural language model, it initiates a new perspective in how to consume the unlabeled speech for unsupervised adaptation. In a state-of-art presentation transcription RNN-T system, our proposed approach achieves 10.46 % and 11.73 % relative WER reduction for 1 min and 10 min unsupervised adaptation.

5. References

[1] Y. Huang and Y. Gong, "Acoustic model adaptation for presentation transcription and intelligent meeting assistant systems," in *Proceedings of ICASSP*, 2020.

[2] Y. Huang, L. He, W. Wei, W. Gale, Y. Li, and Y. Gong, "Using personalized speech synthesis and neural language generator for rapid speaker adaptation," in *Proceedings of ICASSP*, 2020.

[3] H. Soltau, H. Liao, and H. Sak, "Neural speech recognizer: Acoustic-to-word LSTM model for large vocabulary speech recognition," in *arXiv preprint arXiv:1610.09975*, 2016.

[4] A. Kim, T. Hori, and W. S., "Joint ctc-attention based end-to-end speech recognition using multi-task learning," in *Proceedings of ICASSP*, 2017.

[5] C. C. Chiu, T. N. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen, K. Kannan, R. J. Weiss, R. K., and K. Goninaetal, "State-of-the-art speech recognition with sequence-to-sequence models," in *Proceedings of ICASSP*, 2018.

[6] K. Rao, H. Sak, and R. Prabhavalkar, "Exploring architectures, data and units for streaming end-to-end speech recognition with rnn-transducer," in *Proceedings of ASRU*, 2017.

[7] Y. He, T. Sainath, R. Prabhavalkar, I. McGraw, R. Alvarez, D. Zhao, D. Rybach, A. Kannan, Y. Wu, R. Pang, and et al., "Streaming end-to-end speech recognition for mobile devices," in *ICASSP*, 2019.

[8] J. Li, R. Zhao, H. Hu, and Y. Gong, "Improving RNN transducer modeling for end-to-end speech recognition," in *Proceedings of ASRU*, 2019.

[9] K. Hu, T. Sainath, R. Pang, and R. Prabhavalkar, "Deliberation model based two-pass end-to-end speech recognition," in *Proceedings of ICASSP*, 2020.

[10] A. Graves, S. Fernandez, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of ICML*, 2006.

[11] A. Graves, "Sequence transduction with recurrent neural networks," in *CoRR, vol. abs/1211.371*, 2012.

[12] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *Proceedings of ICASSP*, 2016.

[13] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio, "End-to-end attention-based large vocabulary speech recognition," in *Proceedings of ICASSP*, 2017.

[14] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proceedings of NIPS*, 2017.

[15] I. McGraw, R. Prabhavalkar, R. Alvarez, M. Gonzalez, K. Rao, D. Rybach, O. Alsharif, H. Sak, A. Gruenstein, F. Beaufays, and C. Parada, "Personalized speech recognition on mobile devices," in *ICASSP*, 2016.

[16] K. C. Sim, L. Johnson, G. Motta, and H. Zhang, "Personalization of end-to-end speech recognition on mobile devices for named entities," in *Proceedings of ASRU*, 2019.

[17] K. C. Sim, P. Zadrazil, and F. Beaufays, "An investigation into on-device personalization of end-to-end automatic speech recognition models," in *Proceedings of INTERSPEECH*, 2019.

[18] I. Williams, A. Kannan, P. Aleksic, D. Rybach, and T. N. Sainath, "Contextual speech recognition in end-to-end neural network systems using beam search," in *Interspeech*, 2018.

[19] G. Pundak, T. N. Sainath, R. Prabhavalkar, A. Kannan, and D. Zhao, "Deep context: End-to-end contextual speech recognition," in *SLT*, 2018.

[20] D. Yu, H. Su, G. Li, and F. Seide, "KL-divergence regularized deep neural network adaptation for improved large vocabulary speech recognition," in *Proceedings of ICASSP*, 2013.

[21] J. Xue, J. Li, D. Yu, M. Seltzer, and Y. Gong, "Singular value decomposition based low-footprint speaker adaptation and personalization for deep neural network," in *Proceedings of ICASSP*, 2014.

[22] P. Swietojanski, J. Li, and S. Renal, "Learning hidden unit contributions for unsupervised acoustic model adaptation," *IEEE/ACM Transaction on Audio Speech Language Processing*, pp. 1450–1463, 2016.

[23] O. Abdel-Hamid and H. Jiang, "Fast speaker adaptation of hybrid NN/HMM model for speech recognition based on discriminative learning of speaker code," in *Proceedings of ICASSP*, 2013.

[24] G. Saon, H. Soltau, D. Nahamoo, and M. Picheny, "Speaker adaptation of neural network acoustic models using i-vectors," in *Proceedings of ASRU*, 2013.

[25] T. Tan, Y. Qian, and K. Yu, "Cluster adaptive training for deep neural network based acoustic model," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, p. 459–468, 2016.

[26] L. Samarakoon and K. C. Sim, "Factorized hidden layer adaptation for deep neural network based acoustic modeling," in *Proceedings of ICASSP*, 2014.

[27] M. Mohammadreza Ghodsi, X. Liu, J. Apfel, R. Cabrera, and E. Weinstein, "Rnn-transducer with stateless prediction network," in *Proceedings of ICASSP*, 2020.

[28] J. Kirkpatrick, R. Pascanu, N. C. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, D. Hassabis, C. Clopath, D. Kumaran, and R. Hadsell, "Overcoming catastrophic forgetting in neural networksms," in *arXiv:1612.00796*, 2016.

[29] Y. Deng, L. He, and F. K. Song, "Modeling multi-speaker latent space to improve neural TTS: quick enrolling new speaker and enhancing premium voice," in *arXiv preprint arXiv:1812.05253*, 2016.

[30] J. Shen, P. Pang, R. J. Weiss, M. Schuster, M. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. J. Skerry-Ryan, R. A. Saurous, Y. Agiomyrgiannakis, and Y. Wu, "Natural TTS synthesis by conditioning WaveNet on MEL spectrogram predictions," in *Proceedings of ICASSP*, 2018.

[31] M. Sundermeyer, R. Schluter, and H. Ney, "LSTM neural networks for language modeling," in *Proceedings of Interspeech*, 2010.

[32] K. Vijayakumar, A, M. Cogswell, R. S. Ramprasath, Q. Sun, S. Lee, D. Crandall, and D. Batra, "Diverse beamsearch: Decoding diverse solutions from neural sequence models," in *arXiv preprint arXiv:1610.02424*, 2016.

[33] R. Sennrich, B. Haddow, and B. Alexandra, "Neural machine translation of rare words with subword units," in *arXiv preprint arXiv:1508.07909*, 2015.

[34] J. Li, R. Zhao, Z. Meng, and et al., "Developing rnn-t models surpassing high-performance hybrid models with customization capability," in *Proceedings of Interspeech*, 2020.