# FACTORIZED NEURAL TRANSDUCER FOR EFFICIENT LANGUAGE MODEL ADAPTATION

*Xie Chen, Zhong Meng, Sarangarajan Parthasarathy, Jinyu Li*

Microsoft Speech and Language Group

## ABSTRACT

In recent years, end-to-end (E2E) based automatic speech recognition (ASR) systems have achieved great success due to their simplicity and promising performance. Neural Transducer based models are increasingly popular in streaming E2E based ASR systems and have been reported to outperform the traditional hybrid system in some scenarios. However, the joint optimization of acoustic model, lexicon and language model (LM) in neural Transducer also brings about challenges in adapting ASR using just adaptation text. This drawback might prevent their potential applications in practice. In order to address this issue, we propose a novel model, factorized neural Transducer, by factorizing the blank and vocabulary prediction, and adopting a standalone language model for the vocabulary prediction. It is expected that this factorization can transfer the improvement of the standalone language model to the Transducer for speech recognition, which allows various language model adaptation techniques to be applied. We demonstrate that the proposed factorized neural Transducer yields 15.4% to 19.4% WER improvements when out-of-domain text data is used for language model adaptation, at the cost of a minor degradation in WER on a general test set.

***Index Terms—*** factorized neural Transducer, Transformer Transducer, language model adaptation, speech recognition

## 1. INTRODUCTION

In recent years, end-to-end (E2E) based models [1, 2, 3, 4, 5, 6, 7, 8] have attracted increasing research interest in automatic speech recognition (ASR) systems. Compared to traditional HMM based models, where the acoustic model, lexicon and language model are built and optimized separately, a single neural network is used in E2E models to directly predict the word sequence. Nowadays, neural Transducer [9, 10, 11] and Attention-based Encoder-Decoder (AED) [1, 12, 13] are two most popular choices for E2E based ASR systems. AED models achieved very good performance by adopting the attention mechanism and fusing the acoustic and linguistic information at the early stage. However, they are not streamable models in nature. There are some efforts to allow AED models to work in streaming mode, such as monotonic chunk-wise attention [14] and triggered attention [15, 16]. In contrast, the neural Transducer model provides a more attractive solution for streaming ASR and has been reported to outperform traditional hybrid systems [17, 18, 19] in some scenarios. Therefore, in this work, we mainly focus on the Transducer model in light of the streaming scenario in practice.

However, the simplicity of E2E models also brings some sacrifice. There are no individual acoustic and language models in a neural Transducer. Although the predictor looks similar to a language model in terms of model structure and an internal language model [20, 21] could be extracted from the predictor and joint network, it does not perform as a language model because the predictor needs to coordinate with the acoustic encoder closely. Hence, it is not straightforward to utilize text-only data to adapt the Transducer model from the source domain to the target domain. As a result, effective and efficient language model adaptation remains an open research problem for E2E based ASR models.

There are continuous efforts in the speech community to address this issue. One research direction is to adopt Text-to-Speech (TTS) techniques to synthesize audio with the target-domain text [17, 22, 23, 24], and then fine-tune the Transducer model on the synthesized audio and text pairs. However, this approach is computationally expensive. It is not flexible and practical for scenarios requiring fast adaption. LM fusion is another popular choice to incorporate external language models trained on target-domain text, such as shallow fusion [25] and density ratio based LM integration [20, 21, 26, 27, 28]. However, the interpolation weight is task-dependent and needs to be tuned on dev data. The performance might be sensitive to the interpolation weight. There are some recent efforts to fine-tune the predictor [29] or the internal language model [30] with an additional language model loss, and then make it behave similar to a language model. Nevertheless, the predictor in neural Transducer is not equivalent to a language model in nature. It needs to coordinate with the acoustic encoder, and predict the blank to prevent outputting repetitive word [31].

As discussed above, most previous work on LM adaptation adopted the standard neural Transducer architecture [9, 6]. In this paper, we propose a modified model architecture to explicitly optimize the predictor towards a standard neural language model during training. We name it factorized neural Transducer, which factorizes the blank and vocabulary prediction, allowing the vocabulary predictor to work as a standalone language model. As a result, various language model adaptation [32, 33, 34] techniques could be simply applied to the factorized Transducer model. The improvement of the standalone language model is expected to yield consistent performance gain for speech recognition, which is similar to the effect of language model in the HMM based ASR system. We hope this work could shed some light on the re-design of model architecture, by disentangling the fusion of AM and LM in E2E models for efficient language model adaptation and customization.

## 2. NEURAL TRANSDUCER

### 2.1. Neural Transducer Architecture

The neural Transducer model consists of three components, an acoustic encoder, a label predictor and a joint network, as shown in Figure 1. The encoder consumes the acoustic feature $\mathbf{x}_1^t$ and generates the acoustic representation $\mathbf{f}_t$. The predictor computes the label representation $\mathbf{g}_u$ given the history of the label sequence $\mathbf{y}_1^u$. The outputs of encoder and predictor are then combined in the joint network and fed to the output layer to compute the probability dis-

tribution over the output layer. The computation formulas in neural Transducer could be written as below,

$$\begin{aligned} \mathbf{f}_t &= \text{encoder}(\mathbf{x}_1^t) \\ \mathbf{g}_u &= \text{predictor}(\mathbf{y}_1^u) \\ \mathbf{z}_{t,u} &= \mathbf{W}_o * \text{relu}(\mathbf{f}_t + \mathbf{g}_u) \\ P(\hat{y}_{t+1}|\mathbf{x}_1^t, \mathbf{y}_1^u) &= \text{softmax}(\mathbf{z}_{t,u}) \end{aligned} \quad (1)$$

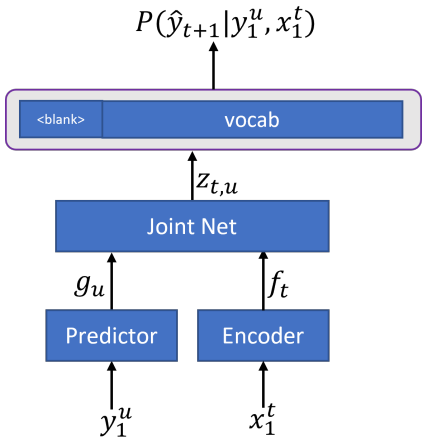


**Fig. 1**. Flowchart of neural Transducer

In order to address the length differences between the acoustic feature $\mathbf{x}_1^T$ and label sequences $\mathbf{y}_1^U$, a special blank symbol, $\phi$, is added to the output vocabulary to represent a null token. Each alignment $\alpha$ contains $T+U$ output tokens, $\hat{y}_1, ..., \hat{y}_{T+U}$, where each output token is an element of the set $\{\phi, \mathcal{V}\}$. The objective function of the Transducer model is to minimize the negative log probability over all possible alignments, which could be written as,

$$\mathcal{J}_t = -\log P(\mathbf{y} \in \mathcal{Y}^*|\mathbf{x}) = -\log \sum_{\alpha \in \beta^{-1}(\mathbf{y})} P(\alpha|\mathbf{x}) \quad (2)$$

where $\beta : \bar{\mathcal{Y}}^* \longrightarrow \mathcal{Y}^*$ is the function to convert the alignment $\alpha$ to label sequence $\mathbf{y}$ by removing the blank $\phi$.

In this paper, we choose the Transformer Transducer (T-T) model as the backbone model which uses Transformer as the acoustic encoder, LSTM as the predictor, by considering the performance and computational costs. The T-T model can be trained and evaluated efficiently as described in [35].

### 2.2. Rethinking the Blank Symbol in Neural Transducer

In the standard neural Transducer model, the predictor looks like a language model in terms of model structure. There are some studies claiming that an internal language model could be extracted from the Transducer model by using predictor and joint network and excluding the blank $\phi$ connection in the output layer [20, 21]. Although this internal language model yields a relatively low perplexity [27], it does not actually perform as a standalone language model. As shown in Equation 1, given the label history $\mathbf{y}_1^u$, the predictor needs to coordinate with the encoder outputs $\mathbf{f}_t$ to predict the output token $\hat{y}_{t+1}$. In addition, it also needs to avoid generating repeated label tokens as the duration of each label normally consists of multiple acoustic frames [31]. Therefore, the predictor plays a special and important role in neural Transducer rather than merely predicting the next vocabulary token as language model.

Here, we use a simple example to further illustrate why the predictor is not working as a language model. In Figure 2, the label of the acoustic feature $\mathbf{x}_1^T$ consists of three characters, "$C$ $A$ $T$". Normally, the acoustic feature sequence is much longer than the label sequence, i.e. $T >> U$. Figure 2 gives an example alignment $\alpha$. In the training stage, at the $t$-th frame, the target is "$A$", given the encoder output $\mathbf{f}_t$ and the label history "<s> $C$". While at the $t+1$-th frame, the target is $\phi$, given the encoder output $\mathbf{f}_{t+1}$ and label history "<s> $C$ $A$". It is safe to assume that the encoder outputs $\mathbf{f}_{t+1}$ and $\mathbf{f}_t$ are similar since there is only one acoustic frame difference and they both lie in the middle of the pronunciation of "$A$" as illustrated in Figure 2. In the $t$-th frame, the predictor predicts "$A$" given the label history "<s> $C$", which is consistent with the language model task. However, in the $t+1$-th frame, the predictor helps to predict "$\phi$" given label history "<s> $C$ $A$", which is different to the language model task where the vocabulary token $T$ is predicted as target. Furthermore, there is no blank $\phi$ in the language model. By considering the large amounts of blank $\phi$ in each alignment, the predictor is not only working as a language model. More importantly, it needs to coordinate with the acoustic encoder output and label history to generate the neural Transducer alignment. Therefore, the job of predictor is not only predicting normal vocabulary tokens but also generating the blank $\phi$ for the co-ordination job. Because of this reason, the predictor cannot be considered as a pure LM [31].

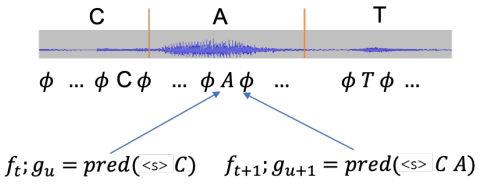


**Fig. 2**. An example alignment in neural Transducer for an audio with label "C A T". The label at time $t$ is $A$ and at time $t+1$ is $\phi$.

## 3. FACTORIZED NEURAL TRANSDUCER

As discussed in Section 2.2, the predictor has two jobs, which are predicting the blank and normal vocabulary tokens according to the Transducer alignment, which hinders the direct use of text-only data for LM adaptation on the predictor. In this section, we introduce a novel model architecture, which is called factorized neural Transducer. Instead of using one predictor to predict both blank and vocabulary tokens, the proposed model factorizes the original prediction network into two separate networks, predicting vocabulary tokens and blank respectively. A standard language model could be used as the vocabulary predictor since there is no blank in the output layer. As a result, the vocabulary predictor could be optimized with the standard LM loss, i.e. cross entropy, on the label sequence during training, and further improved via various LM adaptation techniques given the target-domain text in test time.

The architecture of the proposed factorized neural Transducer model is illustrated in Figure 3. Two predictors are adopted, one is dedicated to predict the blank $\phi$, which is called blank predictor; and the other is to predict the label vocabulary excluding $\phi$, which is called vocabulary predictor. The vocabulary predictor is the same as a normal language model, using history words as input and the log probability of each word as output. The acoustic encoder output $\mathbf{f}_t$ is shared by these two predictors to extract the acoustic representation, but with slightly different combinations. For the prediction of the

blank $\phi$ , it is important to fuse the acoustic and label information as early as possible. Therefore, we adopt the same combination as [6] with a joint network. While for the vocabulary part, we would like to keep a separate language model module. Hence, the acoustic and label information are combined in the logit level, which is similar to the original Transducer paper [9][1]. The exact computation formulas could be written as below,

$$\begin{aligned}
\mathbf{f}_t &= \text{encoder}(\mathbf{x}_1^t) \\
\mathbf{g}_u^b &= \text{predictor}^b(\mathbf{y}_1^u) \\
\mathbf{z}_{t,u}^b &= \mathbf{W}_o^b * \text{relu}(\mathbf{f}_t + \mathbf{g}_u^b) \\
\mathbf{g}_u^v &= \text{predictor}^v(\mathbf{y}_1^u) \\
\mathbf{z}_t^v &= \mathbf{W}_{enc}^v * \text{relu}(\mathbf{f}_t) \\
\mathbf{z}_u^v &= \text{log_softmax}(\mathbf{W}_{pred}^v * \text{relu}(\mathbf{g}_u^v)) \\
\mathbf{z}_{t,u}^v &= \mathbf{z}_t^v + \mathbf{z}_u^v \\
P(\hat{y}_{t+1}|\mathbf{x}_1^t, \mathbf{y}_1^u) &= \text{softmax}([\mathbf{z}_{t,u}^b; \mathbf{z}_{t,u}^v]) \qquad (3)
\end{aligned}$$

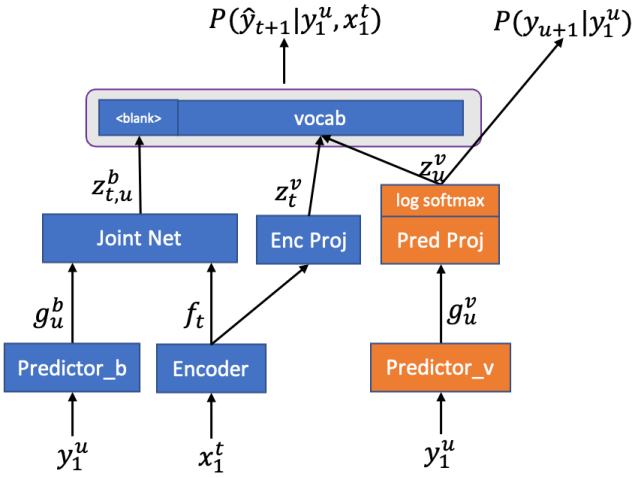


**Fig. 3**. Architecture of factorized neural Transducer

The loss function of the factorized Transducer can be written as

$$\mathcal{J}_f = \mathcal{J}_t - \lambda \log P(\mathbf{y}_1^U) \qquad (4)$$

where the first term is the Transducer loss as defined in Equation 2 and the second term is the language model loss with cross entropy. $\lambda$ is a hyper-parameter to tune the effect of language model loss.

The orange part, denoted as vocabulary predictor, in Figure 3 has the same structure as a standalone language model, which can be viewed as the internal language model in the factorized Transducer. Note that its output is the log probability over the vocabulary. Hence in theory this internal language model could be replaced by any language model trained with the same vocabulary, e.g. LSTM and n-gram LMs. There is no large matrix computation in the factorized Transducer model in the joint network as the standard Transducer model. As a result, the training speed and memory consumption can be improved compared to the standard Transducer model, although there is a slight increase of the model parameters due to the additional vocabulary predictor.

In the training stage, the factorized Transducer model is trained from scratch using the loss function defined in Equation 4. In the adaptation stage, since the vocabulary predictor works as a language model, we could apply any well-studied language model adaptation techniques to adapt the language model on the target-domain text data. For simplicity, in this paper, the language model is directly fine-tuned with the adaptation text data for a specified number of sweeps.

[1]we also tried to apply the log_softmax function on the vocabulary encoder (i.e. Enc Proj) in Figure 3, it presented similar performance, while introduced additional computational cost.

| test set | adapt words(utts) | test words(utts) |
|---|---|---|
| Librispeech | 9.4M(281k) | 210k(11k) |
| call-center | 1.4M(76k) | 77k (4k) |

**Table 1**. Statistics of two test sets for language model adaptation

## 4. EXPERIMENTS

In this section, we investigate the effectiveness of the proposed factorized Transducer model on three test sets, the first one is a general test set to verify the impact of architecture change, and two domain-specific test sets, one is from the public Librispeech corpus, the other is an in-house call center task, are used to evaluate the language model adaptation of the factorized Transducer.

### 4.1. Experiment Setup

64 thousand (K) hours of transcribed Microsoft data are used as the training data. The general test set is used to evaluate the benchmark performance of the standard and factorized neural Transducer, which covers 13 different application scenarios including dictation, far-field speech and call center, consisting of a total of 1.8 million (M) words. The word error rate (WER) averaged over all test scenarios is reported. Two test sets are used for adaptation, the first one is from the public Librispeech data, where the acoustic transcription of the 960-hour training data is used as the adaptation text data, and the standard dev and test sets are adopted for evaluation. The other test set is an internal Microsoft call center test set. The statistics of these two test set is summarized in Table 1. All the in-house training and test data are anonymized data with personal identifiable information removed. 4000 sentence pieces trained on the training data transcription was used as vocabulary. We applied a context window of 8 for the input frames to form a 640-dim feature as the input of Transducer encoder and the frame shift is set to 30ms. Transformer-Transducer (T-T) models are adopted for all Transducer models, where the encoder consists of 18 transformer layers and predictor consists of two LSTM layers. The total number of parameters for standard and factorized Transducer models are 84M and 103M respectively. Note that the encoder of these two Transducer models are the same and the increase of model parameter in the factorized Transducer is mainly from the additional vocabulary predictor. Utterances longer than 30 second were discarded from the training data. The T-T models are trained using the chunk based masking approach as described in [35], with an average latency of 360ms. In this paper, we trained the whole model from scratch without pretraining on the encoder or predictor.

The first experiment is to evaluate the benchmark performance of the factorized T-T model and standard T-T model on the general test set. The factorized Transducer with different $\lambda$ as defined in Equation 4 are also investigated. According to the results shown in

| model | $\lambda$ | PPL | WER |
|---|---|---|---|
| std T-T | - | | 8.10 |
| Factorized T-T | 0.0 | 109.2 | 8.21 |
| | 0.1 | 31.0 | 8.23 |
| | 0.2 | 29.3 | 8.25 |
| | 0.5 | 28.0 | 8.32 |
| | 1.0 | 27.7 | 8.40 |

**Table 2**. PPL and WER results of standard and factorized T-T models on the general test set, note that PPLs are computed over sentence piece level.

| model | PPL | WER | | | |
|---|---|---|---|---|---|
| | | dev | | test | |
| | | clean | other | clean | other |
| std T-T | - | 5.90 | 13.31 | 5.86 | 13.38 |
| +shallow fusion | | 5.29 | 12.03 | 5.20 | 12.36 |
| factorized T-T | 88.9 | 6.18 | 13.76 | 6.24 | 14.14 |
| +adapt text | 45.8 | 4.98 | 11.49 | 5.27 | 11.96 |
| (rel. WERR) | | -19.4% | -16.5% | -15.5% | -15.4% |

**Table 3**. PPL and WER results of factorized T-T on Librispeech test set for language model adaptation.

Table 2, the factorized Transducer models degrade the WER performance slightly compared to the standard Transducer model, which is expected due to the factorization of acoustic and label information for vocabulary prediction. The WER of the factorized T-T increases marginally with increase of $\lambda$. In contrast, the PPL drops dramatically when we include the LM loss in Equation 4 by setting $\lambda$ larger than 0. An LSTM-LM with same model structure trained on the text data results in a PPL of 27.5, which indicates that the internal LM extracted from the factorized T-T presents similar PPL compared to the standard LM. Note that the PPL reported in Table 2 is computed on the sentence piece vocabulary of 4000 tokens. In the following experiment, we adopt the factorized T-T with $\lambda = 0.5$ as the seed model for LM adaptation on text data.

The next experiment investigates the language model adaptation of the factorized T-T model on the Librispeech data. The experiment results are reported in Table 3. Similar to the result on the general test set, the performance of the factorized Transducer is slightly worse than the standard Transducer without adaptation. By using the target-domain text data, significant WER reductions can be achieved, resulting in a relative 15.4% to 19.4% WER improvement on the dev and test sets. The gain over the standard Transducer model is also larger than 10% in spite of the improvement of the baseline model. Compared to the WER of the shallow fusion on the standard T-T model, which is shown in the second row of Table 3, the adapted factorized T-T model still outperforms on most test sets and gives similar performance on the test clean set.

Another experiment is conducted to reveal the relationship between the improvement of vocabulary predictor and the performance of factorized Transducer. The plot of the PPL and WER trends with different amounts of adapt text data is given in Figure 4. It could be seen that the WER gain of the Transducer is highly correlated to the PPL improvement of the vocabulary predictor, which is consistent with the impact of LM in HMM based ASR systems. Note that a sweep of the adaptation text data consists of about 10k steps, it could be seen that four sweeps are enough for the convergence of PPL and WER improvements in the Librispeech task.

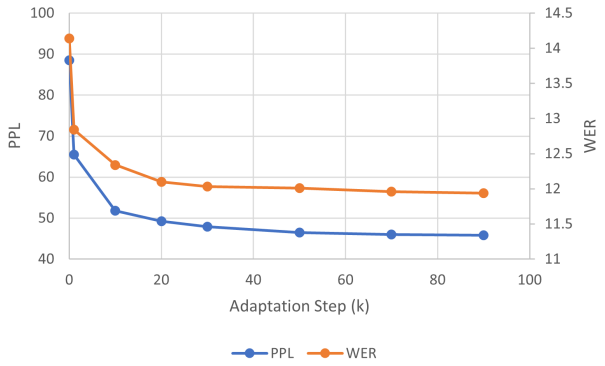


**Fig. 4**. PPL and WER with the increase of LM adaptation step for factorized Transducer in Librispeech test-other set. One sweep of the adaptation text data consists of about 10k steps.

| model | PPL | WER |
|---|---|---|
| std T-T | - | 37.03 |
| +shallow fusion | | 34.36 |
| factorized T-T | 42.0 | 38.14 |
| +adapt | 24.1 | 32.18 |
| (rel. WERR) | | -15.6% |

**Table 4**. PPL and WER results of factorized T-T on the in-house call center test set for language model adaptation

The last experiment aims to validate the effectiveness of the language model for factorized Transducer on an in-house call center test set. The experiment results can be found in Table 4. By fine-tuning the vocabulary predictor of the factorized Transducer on the adapt text, it yields a relative 15.6% WER improvement, which is consistent to the observation on Librispeech, outperforming the shallow fusion on the standard T-T model. This demonstrates that the improvement of the vocabulary predictor could be transferred to the Transducer model. This is of great practical value as it is much easier to collect a large scale of text data compared to the labelled speech data in practice.

## 5. CONCLUSION

Recent years have witnessed the great success of the E2E based models in speech recognition, especially the neural Transducer based model due to their streaming capability and promising performance. However, how to utilize the out-of-domain text for efficient language model adaptation remains an active research topic for neural Transducer. In this paper, we proposed a novel model architecture, factorized neural Transducer, by separating the blank and vocabulary prediction using two predictor networks, and adopting a standalone language model as the vocabulary predictor. Thanks to the factorization, we could adapt the vocabulary predictor with text-only data in the same way as conventional neural language model adaptation. The improvement of the language model is able to be transferred to the Transducer performance on speech recognition. The experiment results demonstrate that significant WER improvements can be achieved by using the target-domain text data, outperforming the shallow fusion on standard neural Transducer model.

## 6. REFERENCES


[1] William Chan, Navdeep Jaitly, Quoc Le, and Oriol Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *Proc. ICASSP*, 2016.

[2] Eric Battenberg, Jitong Chen, et al., "Exploring neural transducers for end-to-end speech recognition," in *Proc. ASRU*, 2017.

[3] Kanishka Rao, Haşim Sak, and Rohit Prabhavalkar, "Exploring architectures, data and units for streaming end-to-end speech recognition with RNN-transducer," in *Proc. ASRU*, 2017.

[4] Chung-Cheng Chiu, Sainath, et al., "State-of-the-art speech recognition with sequence-to-sequence models," in *Proc. ICASSP*, 2018.

[5] Jinyu Li, Guoli Ye, Amit Das, Rui Zhao, and Yifan Gong, "Advancing acoustic-to-word CTC model," in *Proc. ICASSP*, 2018.

[6] Yanzhang He, Tara Sainath, et al., "Streaming end-to-end speech recognition for mobile devices," in *Proc. ICASSP*, 2019.

[7] Jinyu Li, Rui Zhao, Hu Hu, and Yifan Gong, "Improving RNN transducer modeling for end-to-end speech recognition," in *Proc. ASRU*, 2019.

[8] Yangyang Shi, Yongqiang Wang, et al., "Emformer: Efficient memory transformer based acoustic model for low latency streaming speech recognition," *arXiv preprint arXiv:2010.10759*, 2020.

[9] Alex Graves, "Sequence transduction with recurrent neural networks," *arXiv preprint arXiv:1211.3711*, 2012.

[10] Ching-Feng Yeh, Jay Mahadeokar, et al., "Transformer-transducer: End-to-end speech recognition with self-attention," *arXiv preprint arXiv:1910.12977*, 2019.

[11] Qian Zhang, Han Lu, et al., "Transformer transducer: A streamable speech recognition model with transformer encoders and RNN-T loss," in *Proc. ICASSP*, 2020.

[12] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio, "Neural machine translation by jointly learning to align and translate," in *Proc. ICLR*, 2015.

[13] Jan K Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio, "Attention-based models for speech recognition," in *Proc. NIPS*, 2015.

[14] Chung-Cheng Chiu and Colin Raffel, "Monotonic chunkwise attention," in *Proc. ICLR*, 2018.

[15] Niko Moritz, Takaaki Hori, and Jonathan Le Roux, "Triggered attention for end-to-end speech recognition," in *Proc. ICASSP*, 2019.

[16] Chengyi Wang, Yu Wu, et al., "Reducing the latency of end-to-end streaming speech recognition models with a scout network," in *Proc. Interspeech*, 2020.

[17] Jinyu Li, , Rui Zhao, Zhong Meng, et al., "Developing RNN-T models surpassing high-performance hybrid models with customization capability," in *Proc. Interspeech*, 2020.

[18] Tara Sainath, Yanzhang He, et al., "A streaming on-device end-to-end model surpassing server-side conventional model quality and latency," in *Proc. ICASSP*, 2020.

[19] Mahaveer Jain, Kjell Schubert, Jay Mahadeokar, et al., "RNN-T for latency controlled ASR with improved beam search," *arXiv preprint arXiv:1911.01629*, 2019.

[20] Ehsan Variani, David Rybach, Cyril Allauzen, and Michael Riley, "Hybrid autoregressive transducer (HAT)," in *Proc. ICASSP*, 2020.

[21] Zhong Meng, Sarangarajan Parthasarathy, et al., "Internal language model estimation for domain-adaptive end-to-end speech recognition," in *SLT workshop*, 2021.

[22] Khe Chai Sim, Francoise Beaufays, et al., "Personalization of end-to-end speech recognition on mobile devices for named entities," in *ASRU Workshop*. IEEE, 2019.

[23] Xianrui Zheng, Yulan Liu, Deniz Gunceler, and Daniel Willett, "Using synthetic audio to improve the recognition of out-of-vocabulary words in end-to-end ASR systems," in *Proc. ICASSP*. IEEE, 2021.

[24] Yan Deng, Rui Zhao, Zhong Meng, Xie Chen, Bin Liu, Jinyu Li, Yifan Gong, and Lei He, "Improving RNN-T for domain scaling using semi-supervised training with neural TTS," in *Proc. Interspeech*, 2021.

[25] Anjuli Kannan, Yonghui Wu, et al., "An analysis of incorporating an external language model into a sequence-to-sequence model," in *Proc. ICASSP*, 2018.

[26] Erik McDermott, Hasim Sak, and Ehsan Variani, "A density ratio approach to language model fusion in end-to-end automatic speech recognition," in *Proc. ASRU*, 2019.

[27] Zhong Meng, Naoyuki Kanda, et al., "Internal language model training for domain-adaptive end-to-end speech recognition," in *ICASSP*, 2021.

[28] Zhong Meng, Yu Wu, et al., "Minimum word error rate training with language model fusion for end-to-end speech recognition," *arXiv preprint arXiv:2106.02302*, 2021.

[29] Janne Pylkkonen, Antti Ukkonen, Juho Kilpikoski, Samu Tamminen, and Hannes Heikinheimo, "Fast text-only domain adaptation of RNN-transducer prediction network," *arXiv preprint arXiv:2104.11127*, 2021.

[30] Zhong Meng, Yashesh Gaur, et al., "Internal language model adaptation with text-only data for end-to-end speech recognition," in *submitted to Proc. ICASSP*, 2022.

[31] Mohammadreza Ghodsi, Xiaofeng Liu, James Apfel, Rodrigo Cabrera, and Eugene Weinstein, "RNN-transducer with stateless prediction network," in *Proc. ICASSP*, 2020.

[32] Jerome R Bellegarda, "Statistical language model adaptation: review and perspectives," *Speech communication*, vol. 42, no. 1, pp. 93–108, 2004.

[33] Xie Chen, Tian Tan, et al., "Recurrent neural network language model adaptation for multi-genre broadcast speech recognition," in *INTERSPEECH*, 2015.

[34] Ke Li, Hainan Xu, Yiming Wang, Daniel Povey, and Sanjeev Khudanpur, "Recurrent neural network language model adaptation for conversational speech recognition.," in *INTERSPEECH*, 2018.

[35] Xie Chen, Yu Wu, Zhenghao Wang, Shujie Liu, and Jinyu Li, "Developing real-time streaming transformer transducer for speech recognition on large-scale dataset," in *Proc. ICASSP*, 2021.