PERSONALIZATION STRATEGIES FOR END-TO-END SPEECH RECOGNITION SYSTEMS

*Aditya Gourav** *Linda Liu** *Ankur Gandhe* *Yile Gu*
Guitang Lan *Xiangyang Huang* *Shashank Kalmane* *Gautam Tiwari*
Denis Filimonov *Ariya Rastrow* *Andreas Stolcke* *Ivan Bulyko*

Amazon Alexa

ABSTRACT

The recognition of personalized content, such as contact names, remains a challenging problem for end-to-end speech recognition systems. In this work, we demonstrate how first- and second-pass rescoring strategies can be leveraged together to improve the recognition of such words. Following previous work, we use a shallow fusion approach to bias towards recognition of personalized content in the first-pass decoding. We show that such an approach can improve personalized content recognition by up to 16% with minimum degradation on the general use case. We describe a fast and scalable algorithm that enables our biasing models to remain at the word-level, while applying the biasing at the subword level. This has the advantage of not requiring the biasing models to be dependent on any subword symbol table. We also describe a novel second-pass de-biasing approach: used in conjunction with a first-pass shallow fusion that optimizes on oracle WER, we can achieve an additional 14% improvement on personalized content recognition, and even improve accuracy for the general use case by up to 2.5%.

Index Terms— language modeling, automatic speech recognition, rescoring, shallow fusion, personalization

1. INTRODUCTION

The successful recognition of personalized content, such as a user's contacts or custom smart home device names, is essential for automatic speech recognition (ASR). Personalized content recognition is challenging as such words can be very rare or have low probability for the user population overall. For instance, a user's contact list may contain foreign names or unique nicknames, and they may freely name their smart home devices.

This problem is exacerbated for end-to-end (E2E) systems, such as those based on CTC [1], LAS [2], or RNN-T [3]. Unlike hybrid ASR systems, which include acoustic and language model (LM) components that are trained separately, E2E systems use a single network that is trained end-to-end. Whereas in a hybrid system, the LM component can be trained separately on any written text, in an E2E system, the training is generally restricted to acoustic-text pairs. As a result, E2E systems are often trained with less data than hybrid systems, making personalized content recognition particularly challenging given the limited representation during training. Furthermore, hybrid systems are able to incorporate personal content into the decoding search graph, i.e., via class-based language models and on-the-fly composition of biasing phrases and n-grams [4, 5, 6, 7, 8].

Various approaches have been proposed for improving personalized content recognition for E2E models, including model fine-tuning with real or synthesized audio data [9], incorporating personalized content directly during E2E training using a separate bias-encoder module [10], using a token passing decoder with efficient token recombination during inference [11], and shallow fusion (e.g., [12, 8, 13, 14]).

In this work, we describe a novel approach to address this problem using a combination of first-pass shallow fusion and second-pass rescoring. We first provide a comparison of a few shallow fusion approaches: shallow fusion applied at the word-level and subword level, as well as contextual shallow fusion. We describe a novel algorithm that uses grapheme-level lookahead to perform subword-level rescoring, thus bypassing the need to build subword-level language models that are dependent on the wordpiece model that generates the subwords. We show the benefit of contextual shallow fusion in capturing improvement in personalized content recognition. Finally, we describe a novel de-biasing approach in which we treat the second-pass rescoring as an optimization problem to optimally combine scores from the E2E model, the shallow fusion model, and second-pass LMs. Apart from improving recognition for the personalized content, it also improves the general recognition.

2. PREVIOUS WORK

One popular approach to improve personalized content recognition is via shallow fusion [15]. In shallow fusion, the scores from an external language model $Score_{SF}(y)$ scaled by a factor λ are combined with the main decoding scores $P_{RNNT}(y \mid x)$ during beam search:

$$\hat{y} = \arg\max_{y}(\log P_{RNNT}(y \mid x) + \lambda \log Score_{SF}(y)) \qquad (1)$$

This biasing can be applied at word boundaries [8], at the grapheme level [11, 13, 10], or at the subword level [13, 14]. Given that E2E models generally used a constrained beam [16], applying biasing only at word boundaries cannot improve performance if the relevant word does not already appear in the beam. As a result, compared to grapheme-level biasing which tends to keep the relevant words on the beam, word-level biasing results in less improvement on proper nouns such as contact names [10]. Applying biasing at the subword level, which would result in sparser matches at each step of the beam compared to the grapheme level, results in further improvements [13].

One challenge in applying biasing at the subword level, particularly for personalization, is that each of the biasing models needs to be built at the subword level and include all possible segmentations of a given word. This can be expensive when we have one or more models per user, particularly if the wordpiece model used to train the first-pass model often changes. Unlike previous work,

*Equal contribution

which generally relies on composition with a speller FST to transduce a sequence of wordpieces into the corresponding word (e.g., [13, 10, 17]), we describe a novel prefix-matching algorithm in Section 3.2.2 that enables the language models to be kept at the word level and applies the subword decomposition at inference time.

Another challenge these shallow fusion approaches to personalization is how to improve recognition of personalized content while not degrading performance on general non-personalized content; to this end, several strategies for applying contextual biasing have been proposed [13, 14, 11]. Many of these strategies reveal that applying shallow fusion in context minimizes, but does not completely remove, the degradation observed on general data, and do not discuss the potential impact of second-pass rescoring. For example, [13] finds that applying contextual shallow fusion decreases the negative impact on general content while maintaining performance on the shallow fusion content; however, even in this best case, they report a slight degradation of 5.8% (6.9 to 7.3 WER, cf. 12.5 for non-contextual shallow fusion) on general data.

On one hand, a more aggressive shallow fusion model enables more personalized content to appear in the n-best hypotheses but on the other hand, it is also more likely to cause false recognitions of the biased personalized content. To address this, we present a strategy in which we optimize shallow fusion for the n-best, as opposed to the 1-best, hypotheses, thereby maximizing the personalized content present in the n-best. To recover the correct 1-best, we explore a novel second-pass de-biasing approach that optimizes the combination of the E2E, shallow fusion, and second-pass scores.

3. METHODS

3.1. Baseline RNN-T model

Following [18], our baseline RNN-T model consists of an encoder comprised of five LSTM layers of size 1024, and a two-layer LSTM prediction network of size 1024 with an embedding layer of 512 units. The softmax layer consists of 4k (subword) output units. Our model was trained on over 200k hours of anonymized utterances from interactions with a voice assistant according to the minimum word error rate criterion [19, 18].

3.2. First-pass shallow fusion

3.2.1. Personalized models

For each anonymized user in our test set, we construct three personalized models, corresponding to (1) contact names (2) smart home device names and (3) enabled application names. Each of these models is represented as a word-level weighted finite state transducer (FST). An example is shown in Figure 1a. For simplicity, in our experiments, each word level arc has the same weight of -1. On average, each user has 600 personalized contact names, 50 device names, and 70 enabled applications.

3.2.2. Subword rescoring with lookahead

We describe our approach to biasing at the subword level using our word-level personalized models (Algorithm 1). We leverage ideas similar to [20] for subword level lookahead weight pushing and start with a word-level model represented as an FST, such as the one shown in Figure 1a. In this case, there are three paths associated with this FST, containing the words "play", "player", and "playground". In Figure 1b, we show the subword breakdown for these words, based on some wordpiece model. The weights on each path

Algorithm 1 On-the-fly subword rescoring with lookahead. T and s represent word level FST and a non-final state of it. i denotes the starting state of subword level FST and i[e] the input symbol string for a transition e. W is a sequence of subword input. R denotes a set of weights. E_s denotes all transitions starting from s in T. $\pi(a, b)$ represents a path from a to b. w_e is the net weight for transition e. t is the previous state in subword FST.

$\mathbf{Expand}(T, s, i, W)$:
1. Initialize : $R \leftarrow \phi$; $prefix \leftarrow \epsilon$; $w_{prev} \leftarrow 0$
2. Sort : E_s by $i[E_s]$ in lexicographical order
3. for sw in W **do**
4. **if** sw is $delimiter$ **then**
5. **if** $prefix \in i[E_s]$ **then**
6. Return $R \cup (w_e - w_{prev})$
7. **else**
8. Return $R \cup w^{-1}(\pi(i, t))$
9. $prefix \leftarrow \mathbf{Concatenate}(prefix, sw)$
10. $E_s \leftarrow$ **BinarySearch**$(prefix \in \mathbf{Prefix}(i[E_s]))$
11. **if** E_s is empty **then**
12. Return $R \cup w^{-1}(\pi(i, t))$
13. **else**
14. $N \leftarrow$ max string length in $i[E_s]$
15. $L \leftarrow$ prefix string length
16. $w_{lookahead} \leftarrow \bigoplus w(e \in E_s))$
17. $w_{pushed} \leftarrow w_{lookahead} \cdot L/N$
18. Append:$R \leftarrow R \cup (w_{pushed} - w_{prev})$
19. $w_{prev} \leftarrow w_{pushed}$
20. Return R

are determined via Algorithm 1. Notice that the net weight for each path remains the same: i.e., the weight between state 0 and 5 (representing the word "player") in the subword-level FST is (-1.6) +(-1.6)+(-4.8) = -8, which is the same as the weight for the same word in the word-level FST. The weight w_{pushed} for each transition state is determined as follows: $w_{pushed} = (\frac{L}{N})(w_{lookahead})$, where L is the length of the prefix so far and N is the longest length of all matched words. In our example, given the input sequence "play" (pl, ay, _) from Figure 1, we can see there are three arcs prefixed with the subword "pl": thus, we have $L = 2$, $N = 10$, and the pushed weight is -8 * 2 / 10 = -1.6. Additionally, similar to [10, 17], we add fallback arcs for each non-final state with a weight equal to the negation of the current total weight up to that point.

This approach is beneficial as it avoids unnecessary arc expansion and provides a heuristic approach to perform subword-level rescoring without the need to build the biasing FST itself directly at the subword level. Additionally, this prefix matching approach enables us to consider any possible subword sequences for a word. To optimize the search for arcs that have a common prefix string, we sort the input arc in lexicographic order so that we can use binary search to find the lower and upper bound of arc indices. As we continue to process subword input, we are able to narrow down the search range quickly. We also cache all newly created states in subword level FST S, which results in efficient weight evaluation.

3.2.3. Contextual boosting model

Following previous work such as [4, 11, 5, 14], we construct a class-based language model containing three classes: contact names, home automation device names, and application names. To build the contextual biasing LM, we identified all utterances containing words that were annotated with aforementioned classes. We then

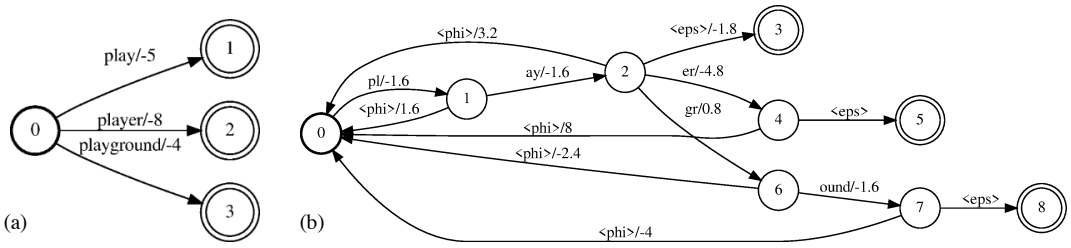

Fig. 1: A illustration of (a) word-level biasing FST (b) subword-level FST with transition state weights evaluated using a grapheme-level lookahead. Additional phi self loop at the start state is not shown.

replaced the word(s) in the utterance with the corresponding class tag (e.g., @contactname). All utterances with the replaced class tags that occurred a minimum of 10 times were included in the contextual biasing FST. Unlike a typical class-based model, all arcs on the class-based model are unweighted. Weights only appear in the corresponding personalized models, which are injected at each class tag. Both the class-based LM and personalized models operate at the subword level using the algorithm described in Section 3.2.2.

3.3. Datasets

We evaluate on (1) a 20k utterance contact name test set and (2) a 20k utterance test set representing the general use case. Both test sets consist of anonymized data from real user interactions with a personal assistant device.

3.4. Second-pass rescoring

We rescore 8-best hypotheses from the first-pass shallow fusion as described in Section 3.2. Each n-best hypothesis y_i can be assigned a score based on the following equation:

$$Score(y_i) = \log P_{RNNT}(y_i \mid x) + \alpha Score_{SF}(y_i) + \beta \log P_{RLM}(y_i) \quad (2)$$

$P_{RLM}(y_i)$ is the probability of the hypothesis y_i assigned by the rescoring LM, $Score_{SF}(y_i)$ is the shallow fusion score of y_i from the first-pass. α and β are the tunable scaling factors. In the tuning stage, we resort to a simulated annealing algorithm as described in [21] to find the optimal values of α and β. The objective of the optimization is to minimize the overall WER of the dev set. This approach enables us to optimally combine multiple rescoring LMs with the first-pass scores.

To select the rescoring LM, we use the domain aware rescoring framework described in [22] to differentiate between the utterances with contact names and generic ones. For the generic utterances, we use an NCE based neural LM (NLM) [23] trained on 80 million utterances from live traffic. The model consists of two LSTM layers, each with 512 hidden units. For the utterances with contact names, we use a KN-smoothed 4-gram class based LM [24], with a single ContactName class, trained on utterances with word annotations.

4. RESULTS AND DISCUSSION

We report on word error rate reduction (WERR) and oracle WERR to the baseline RNN-T model throughout. The oracle WERR is computed by finding the hypothesis in the 8-best that minimizes WER for each utterance.

	Contacts		General	
Model	WERR	Oracle	WERR	Oracle
RNN-T	–	–	–	–
+word(1.0)	-6.5	-2.3	0.4	-0.2
+word(1.5)	-6.5	-2.5	1.4	0.1
+word(2.0)	-5.0	-1.6	3.0	0.1
+noctxt-subwd(1.0)	-13.3	-10.9	-0.5	0.0
+noctxt-subwd(1.5)	-14.2	-14.4	0.1	0.0
+noctxt-subwd(2.0)	-12.7	-16.9	2.0	0.6
+noctxt-subwd(2.5)	-7.8	-18.4	5.0	0.8
+noctxt-subwd(3.0)	-0.4	-18.1	9.6	1.7
+ctxt-subwd(1.0)	-14.0	-10.9	-0.5	0.0
+ctxt-subwd(1.5)	-16.3	-13.6	-0.3	0.0
+ctxt-subwd(2.0)	-16.5	-16.9	0.6	0.3
+ctxt-subwd(2.5)	-14.3	-16.9	2.7	1.1
+ctxt-subwd(3.0)	-10.8	-16.7	5.0	2.5

Table 1: Results using only the contact names personalized model, with different biasing weights, and comparing word-level biasing(word), with subword-level biasing with (ctxt-subwd) and without context(noctxt-subwd)

4.1. Comparing shallow fusion approaches

In Table 1, we report results comparing word-level biasing, to subword-level biasing with and without context, using different biasing weights. We report results using only the personalized contact names model for shallow fusion. We find significant improvements in WERR and oracle WERR when applying biasing at the subword level (best WERR improvement: 14.2%) compared to the word level (best WERR improvement: 6.5%). This aligns with previous work [10, 13], which found that applying biasing at the subword level allows more critical words to stay on the beam.

Comparing the subword results with and without context, we observe larger improvements in WERR on contact names at higher biasing weights when using the contextual biasing model. For example, at a weight of 2.5, we observe improvements of 14.3% with context, but only 7.8% without context. Additionally, we observe that constraining shallow fusion with context decreases the impact on the general WERR at higher biasing weights.

Finally, we observe that increasing the biasing weight leads to improvements in oracle WERR on contact names, *even when overall WERR improvements decrease*. This suggests that a higher weight allows for more personalized content to appear in the n-best hypotheses, even as it increases the number of false recognitions in the 1-best hypothesis. We return to this point later.

Model	Contacts WERR	Contacts Oracle	General WERR	General Oracle
RNN-T	–	–	–	–
+noctxt-subwd(1.0)	-14.2	-11.6	0.1	0.0
+noctxt-subwd(1.5)	-13.1	-15.3	2.6	0.5
+noctxt-subwd(2.0)	-8.8	-17.3	8.5	1.7
+noctxt-subwd(2.5)	2.9	-18.4	18.7	3.4
+ctxt-subwd(1.0)	-14.3	-11.1	-0.5	0.0
+ctxt-subwd(1.5)	-16.5	-13.4	0.5	0.6
+ctxt-subwd(2.0)	-16.5	-16.1	2.6	1.1
+ctxt-subwd(2.5)	-14.2	-16.5	5.7	2.5
+ctxt-subwd(3.0)	-14.3	-16.7	5.7	2.5

Table 2: Results using three personalized models, with different biasing weights. Biasing with context helps to avoid general degradation at the same level of biasing weights

Model	2P no de-biasing Contacts	2P no de-biasing General	2P w/ de-biasing Contacts	2P w/ de-biasing General
+noctxt-subwd(2.5)	-21.2	0.6	-28.5	-2.7
+ctxt-subwd(2.0)	-25.3	-1.7	-27.4	-2.5

Table 3: Results of de-biasing the shallow fusion scores for contact name personalized model in second-pass(2P).

4.2. Adding additional personalized content in shallow fusion

In Table 2, we show that biasing in context helps to avoid degradation on general use cases particularly as the number of classes increases. For these results, we use additional personal models (devices, applications). We can observe that degradation on the general test set is more pronounced when the amount of biasing content increases. This is in line with previous work (e.g., [8]). Specifically, using a biasing weight of 2.5, we observe an 8.5% degradation on the general test set without context, but only 2.6% with context. Critically, we observe that the WERR for contact names is preserved.

4.3. Second Pass Rescoring

A trend seen in Table 1 is that the 1-best WERR for both the Contacts and the General test sets degrades as the shallow fusion biasing factor increases. However, Oracle WERR for the Contacts test set improves. We address this divergence using second-pass rescoring.

We note that second-pass rescoring *without* shallow fusion provides an improvement of 16.5% and 2.3% on the Contacts and General test sets. Following sections elicit that we see an additional 10-15% improvement on the Contacts test set when shallow fusion is used along with second-pass rescoring. To the best of our knowledge, no previous work has shown this synergy.

4.3.1. De-biasing shallow fusion scores

We observe that re-weighting the shallow fusion scaled scores from the first-pass helps us achieve better WERR compared to adding it with the first-pass RNN-T scores. i.e., setting $\alpha = 1$ during optimization in Equation 2. It helps in achieving better WERR for shallow fusion with or without context, as can be seen in Table 3. We call this method de-biasing in second-pass and use it in the results reported in the subsequent sections.

We observe that de-biasing is especially useful when there is no context-based biasing in the first pass. It improves recognition for personalized content without compromising the WER of the general test set.

Model	First-pass Contacts	First-pass General	2P w/ de-biasing Contacts	2P w/ de-biasing General
One biasing model				
+noctxt-subwd(2.5)	-7.8	5.0	-28.5	-2.7
Three biasing models				
+noctxt-subwd(2.5)	2.9	18.4	-29.1	-2.7

Table 4: Results of second-pass rescoring when more personalized models are added in shallow fusion

Model	Contacts WERR	General WERR
+noctxt-subwd(2.0)	-27.2	-2.3
+noctxt-subwd(2.5)	-28.5	-2.7
+noctxt-subwd(3.0)	-29.2	-2.9
+noctxt-subwd(3.5)	-29.3	-2.5
+ctxt-subwd(2.0)	-27.4	-2.5
+ctxt-subwd(2.5)	-28.4	-2.2
+ctxt-subwd(3.0)	-30.0	-2.3

Table 5: Results of second-pass rescoring over the contact names personalized model, with different biasing weights, with and without context

4.3.2. Adding additional personalized models in shallow fusion

Second-pass optimization can not only recover from degradation in WERR but can also improve WERR when additional personalized models are added to first-pass shallow fusion without context (Table 4). The first-pass degradation can be seen in Table 2 and is reproduced in Table 4: we observe that in general, incorporating more biasing models without context results in larger degradations on the general test set. However, second-pass rescoring with de-biasing enables us to completely recover from these degradations, while continuing to improve overall contact name WERR. This aligns with our reasoning that second-pass can improve the first-pass 1-best degradation as long the first-pass oracle WERR continues to improve.

In Table 5, we report the WERR post second-pass rescoring for various weights of shallow fusion biasing, with and without context. As the biasing weight increases, we improve WERR for both the Contacts and General test sets.

5. CONCLUSION

In this work, we have presented several strategies to improve personal content recognition for end-to-end speech recognition systems. We have outlined a novel algorithm for efficient biasing of personalized content on the subword level at inference time. This helps us improve on personal content recognition by 14% - 16% compared to RNN-T. We also describe a novel second-pass optimization to improve recognition by an additional 13% - 15% without degrading the general use case. Combining the two strategies, we achieve 27% - 30% improvement overall in personal content recognition and about 2.5% improvement on the general test set. We also elucidate ways to tackle degradation on the general test set when biasing the RNN-T model in the absence of any context.

6. REFERENCES

[1] Alex Graves, "Sequence transduction with recurrent neural networks," *CoRR*, vol. abs/1211.3711, 2012.

[2] William Chan, Navdeep Jaitly, Quoc V. Le, and Oriol Vinyals, "Listen, attend and spell," *CoRR*, vol. abs/1508.01211, 2015.

[3] Alex Graves and Navdeep Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *International conference on machine learning*, 2014, pp. 1764–1772.

[4] Petar Aleksic, Cyril Allauzen, David Elson, Aleksandar Kracun, Diego Melendo Casado, and Pedro J Moreno, "Improved recognition of contact names in voice commands," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 5172–5175.

[5] Peter F Brown, Vincent J Della Pietra, Peter V Desouza, Jennifer C Lai, and Robert L Mercer, "Class-based n-gram models of natural language," *Computational linguistics*, vol. 18, no. 4, pp. 467–480, 1992.

[6] Ian McGraw, Rohit Prabhavalkar, Raziel Alvarez, Montse Gonzalez Arenas, Kanishka Rao, David Rybach, Ouais Alsharif, Haşim Sak, Alexander Gruenstein, Françoise Beaufays, et al., "Personalized speech recognition on mobile devices," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 5955–5959.

[7] Keith Hall, Eunjoon Cho, Cyril Allauzen, Francoise Beaufays, Noah Coccaro, Kaisuke Nakajima, Michael Riley, Brian Roark, David Rybach, and Linda Zhang, "Composition-based on-the-fly rescoring for salient n-gram biasing," in *Interspeech 2015, International Speech Communications Association*, 2015.

[8] Ian Williams, Anjuli Kannan, Petar S Aleksic, David Rybach, and Tara N Sainath, "Contextual speech recognition in end-to-end neural network systems using beam search.," in *Interspeech*, 2018, pp. 2227–2231.

[9] Khe Chai Sim, Françoise Beaufays, Arnaud Benard, Dhruv Guliani, Andreas Kabel, Nikhil Khare, Tamar Lucassen, Petr Zadrazil, Harry Zhang, Leif Johnson, 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)*. IEEE, 2019, pp. 23–30.

[10] Golan Pundak, Tara N Sainath, Rohit Prabhavalkar, Anjuli Kannan, and Ding Zhao, "Deep context: end-to-end contextual speech recognition," in *2018 IEEE spoken language technology workshop (SLT)*. IEEE, 2018, pp. 418–425.

[11] Zhehuai Chen, Mahaveer Jain, Yongqiang Wang, Michael L Seltzer, and Christian Fuegen, "End-to-end contextual speech recognition using class language models and a token passing decoder," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6186–6190.

[12] Shubham Toshniwal, Anjuli Kannan, Chung-Cheng Chiu, Yonghui Wu, Tara N Sainath, and Karen Livescu, "A comparison of techniques for language model integration in encoder-decoder speech recognition," in *2018 IEEE spoken language technology workshop (SLT)*. IEEE, 2018, pp. 369–375.

[13] Ding Zhao, Tara N Sainath, David Rybach, Pat Rondon, Deepti Bhatia, Bo Li, and Ruoming Pang, "Shallow-fusion end-to-end contextual biasing.," in *Interspeech*, 2019, pp. 1418–1422.

[14] Rongqing Huang, Ossama Abdel-hamid, Xinwei Li, and Gunnar Evermann, "Class LM and Word Mapping for Contextual Biasing in End-to-End ASR," in *Proc. Interspeech 2020*, 2020, pp. 4348–4351.

[15] Anjuli Kannan, Yonghui Wu, Patrick Nguyen, Tara N Sainath, Zhijeng Chen, and Rohit Prabhavalkar, "An analysis of incorporating an external language model into a sequence-to-sequence model," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 1–5828.

[16] Jinyu Li, Rui Zhao, Hu Hu, and Yifan Gong, "Improving rnn transducer modeling for end-to-end speech recognition," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2019, pp. 114–121.

[17] Yanzhang He, Tara N Sainath, Rohit Prabhavalkar, Ian McGraw, Raziel Alvarez, Ding Zhao, David Rybach, Anjuli Kannan, Yonghui Wu, Ruoming Pang, et al., "Streaming end-to-end speech recognition for mobile devices," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6381–6385.

[18] Jinxi Guo, Gautam Tiwari, Jasha Droppo, Maarten Van Segbroeck, Che-Wei Huang, Andreas Stolcke, and Roland Maas, "Efficient Minimum Word Error Rate Training of RNN-Transducer for End-to-End Speech Recognition," in *Proc. Interspeech 2020*, 2020, pp. 2807–2811.

[19] Matt Shannon, "Optimizing expected word error rate via sampling for speech recognition," *CoRR*, vol. abs/1706.02776, 2017.

[20] Mehryar Mohri and Michael Riley, "A weight pushing algorithm for large vocabulary speech recognition," in *EUROSPEECH 2001 - SCANDINAVIA - 7th European Conference on Speech Communication and Technology*, 2001, pp. 1603–1606.

[21] Yang Xiang, Sylvain Gubian, Brian Suomela, and Julia Hoeng, "Generalized simulated annealing for efficient global optimization: the GenSA package for R.," *The R Journal Volume 5/1, June 2013*, 2013.

[22] Linda Liu, Yile Gu, Aditya Gourav, Ankur Gandhe, Shashank Kalmane, Denis Filimonov, Ariya Rastrow, and Ivan Bulyko, "Domain-aware neural language models for speech recognition," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021.

[23] Anirudh Raju, Denis Filimonov, Gautam Tiwari, Guitang Lan, and Ariya Rastrow, "Scalable Multi Corpora Neural Language Models for ASR," in *Proc. Interspeech 2019*, 2019, pp. 3910–3914.

[24] R. Kneser and H. Ney, "Improved backing-off for m-gram language modeling," in *1995 International Conference on Acoustics, Speech, and Signal Processing*, 1995, vol. 1, pp. 181–184 vol.1.