Recurrent Neural Network Language Model Adaptation for Conversational Speech Recognition

Ke Li[1], *Hainan Xu*[1], *Yiming Wang*[1], *Daniel Povey*[1,2], *Sanjeev Khudanpur*[1,2]

[1] Center for Language and Speech Processing,
[2] Human Language Technology Center of Excellence,
Johns Hopkins University, Baltimore, MD, USA

{kli26,hxu31,freewym,khudanpur}@jhu.edu, dpovey@gmail.com

Abstract

We propose two adaptation models for recurrent neural network language models (RNNLMs) to capture topic effects and long-distance triggers for conversational automatic speech recognition (ASR). We use a fast marginal adaptation (FMA) framework to adapt a RNNLM. Our first model is effectively a cache model – the word frequencies are estimated by counting words in a conversation (with utterance-level hold-one-out) from 1st-pass decoded word lattices, and then is interpolated with a background unigram distribution. In the second model, we train a deep neural network (DNN) on conversational transcriptions to predict word frequencies given word frequencies from 1st-pass decoded word lattices. The second model can in principle model trigger and topic effects but is harder to train. Experiments on three conversational corpora show modest WER and perplexity reductions with both adaptation models.

Index Terms: ASR, recurrent neural network language model (RNNLM), neural language model adaptation, fast marginal adaptation (FMA), cache model, deep neural network (DNN), lattice rescoring

1. Introduction

Language models are a vital component of an automatic speech recognition (ASR) system. A simple language model is an n-gram [1]. In recent years, recurrent neural network language models (RNNLMs) have consistently surpassed traditional n-grams in ASR and related tasks [2, 3, 4, 5, 6].

In conversational speech recognition, if a word has been uttered, the same word and topic-related words are likely to appear again. For example, if “Korea” appears in a conversation, the same word and topic-related word “Seoul” is more likely to appear again. Though RNNLMs can in principle implicitly model these phenomena, we believe that in practice they probably do not model them very well, so there may be value in combining RNNLMs with explicit models that capture the same kinds of effects as cache models [7, 8] and trigger models [9, 10].

In this work, we propose two adaptation models for RNNLMs to better model these phenomena for conversational speech recognition. We adopt a fast marginal adaptation (FMA) framework [11] to adapt a RNNLM, i.e., multiplying the probabilities from the RNNLM by a factor specific to each word, and renormalizing. These factors are related to a conversation-specific estimate of word frequency based on 1st-pass decoded word lattices. The first adaptation model is a conversational cache model estimated by counting words in a conversation (holding out the current utterance) from 1st-pass decoded word lattices. And then it is interpolated with the background unigram distribution estimated from the training text corpus. In the second model, we train a deep neural network (DNN) to predict word frequencies given word frequencies (it is trained with different subsets of training conversations as input and output, so it does not simply learn the identity mapping); in test time we give it word frequencies obtained from word lattices from 1st-pass decoding. This DNN in principle can model topic effects as well as cache-like effects, but it requires training, and overfitting can be an issue on small datasets. Both adaptation models incorporate both past and future context information from the 1st-pass decoded word lattices. The adapted RNNLMs are used for 2nd-pass rescoring.

The rest of this paper is organized as follows. Section 2 introduces related prior work. Section 3 describes the FMA framework, the two adaptation models, and the adaptation and rescoring pipeline. The experimental setup is briefly explained in Section 4. Section 5 shows experiments, results, and related analysis. The conclusion and future work are presented in Section 6.

2. Prior Work

In this section, we briefly introduce prior work on LM adaptation. We first introduce LM adaptation to recent history contexts. For n-gram models, adding a cache component is a common approach and has shown success in early ASR research [7, 8]. Jelinek et al. adopted this approach to adapt a trigram LM [12] and obtained reductions on both word error rates (WERs) and perplexity (PPL). Kneser et al. [11] proposed a dynamic marginal adaptation framework for domain and online adaptations. For neural network language model adaptation, Grave et al. [13, 14] adapted RNNLMs to recent history by a neural cache scheme – storing past hidden activations as memory and accessing them through dot product with the current hidden activation.

For RNNLM adaptation to target domains, Chen et al. [15] explored multi-genre adaptation task using topic representations as an additional input feature. Gangireddy et al. [16] investigated domain adaptation (genre and show levels) by scaling forward-propagated hidden activations and fine-tuning the parameters of the whole RNNLM in a broadcast transcription task. Mikolov et al. [17] utilized learned topic vectors as extra input features to capture local context for RNNLM adaptation. Ma et al. [18] explored several domain adaptation approaches including fine-tuning for DNN and LSTM based LMs. Singh et al. [19] adapted RNNLM under the FMA framework [11] and conducted only one pass decoding with the adapted RNNLM.

3. Methods

3.1. Fast Marginal Adaptation for RNNLM

To generate an adapted version of RNNLM $p_{\text{rnnlm}}^{\text{adapt}}(w|h)$ which leverages the conversational context, we apply a FMA framework [11]. We first train a baseline RNNLM $p_{\text{rnnlm}}(w|h)$ on the background corpus and then apply the FMA framework yielding:

$$p_{\text{rnnlm}}^{\text{adapt}}(w|h) = \frac{1}{Z(h)} \cdot \left(\frac{p^{\text{adapt}}(w)}{p^{\text{bg}}(w)} \right)^{\alpha} \cdot p_{\text{rnnlm}}(w|h) \tag{1}$$

where $p^{\text{adapt}}(w)$ is the probability distribution from the adaptation model (conversational cache and DNN models), $p^{\text{bg}}(w)$ is the background unigram distribution estimated from the background corpus, and $Z(h)$ is a normalization constant. The hyper-parameter α is for controlling the scaling factor of the adapted RNNLM. The range of α is $[0, 1]$.

Now we introduce our two modifications based on Equation (1). We want to choose the adaptive models as close as possible to the locally estimated unigram distribution while constraining them to respect the background estimates. Considering this, we make the first modification: we use the linear interpolation of the conversational adaptation model with the background unigram model as the numerator of the scaling factor:

$$\beta \cdot p^{\text{adapt}}(w) + (1 - \beta) \cdot p^{\text{bg}}(w) \tag{2}$$

which yielding the following adapted RNNLM:

$$\begin{aligned} p_{\text{rnnlm}}^{\text{adapt}}(w|h) =& \frac{1}{Z(h)} \cdot \left(\beta \cdot \frac{p^{\text{adapt}}(w)}{p^{\text{bg}}(w)} + (1 - \beta) \right)^{\alpha} \cdot \\ & p_{\text{rnnlm}}(w|h) \end{aligned} \tag{3}$$

The second modification is to use an empirical location-based weighting method for estimating the conversational cache models on the word lattices from the 1st-pass decoding. Our intuition is that, for decoding an utterance, utterances closer to it can be more important than those far away. Therefore, we put more weights on utterances within a window centered at the current utterance being decoded in the 2nd-pass rescoring.

3.2. Conversational Cache Model

Cache models exploit the unigram distribution of a recent history context (a fixed number of words or a document) to improve an original LM. After a word is spoken in a conversation, there is more chances that it is spoken again. For example, the frequency of the word "pollution" is 0.6% in a environmental related conversation, compared to 0.008% in the whole Switchboard training corpus. Thus, cache models that stores recent history can adapt LMs to local context.

Our conversational cache models are unigram cache models, i.e., self-trigger models, estimated on conversations from 1st-pass decoded word lattices, and thus can adapt RNNLMs to conversational context (topic effects and long-distance self-triggers). Considering both past and future contexts are useful, our conversational cache models, which are estimated by counting words (holding out the current utterance) in conversations, contain both contexts for the current utterance to be rescored in 2nd-pass rescoring.

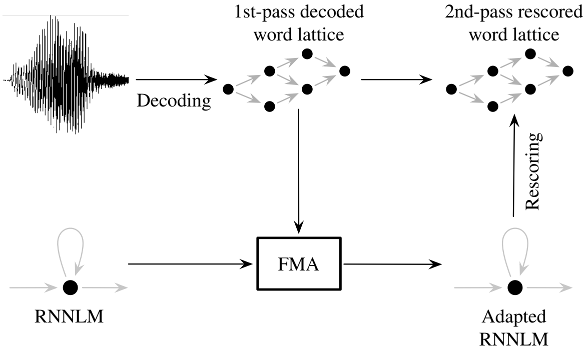

Figure 1: *Adaptation and Rescoring Pipeline*

3.3. Deep Neural Network (DNN) Model

Though a cache model can capture the long-distance self-triggers efficiently, it can not model the co-occurrences or triggers of different words well. As an extension of conversational cache model, DNN adaptation model [20, 21, 22] in principle can model these phenomena within conversations.

In our corpus, each conversation contains the utterances of two sides, corresponding to two speakers, which we call side A and B. We split both sides into two equal-sized disjoint subsets (A1, A2 and B1, B2), and collect unigram counts of each subset. We train a DNN model, which takes unigram counts of one subset as input, and predicts the unigram distribution of the other subset from the same speaker. The DNN is trained to minimize the Kullback-Leibler (KL) divergence of the predicted distribution and the actual distribution.

3.4. Adaptation and Rescoring Pipeline

Figure 1 gives an outline of the overall adaptation and rescoring scheme. We first perform a 1st-pass decoding with a n-gram LM, and generate word lattices from which we extract word-posteriors for the two adaptation models. The adaptation scheme works by combining a baseline RNNLM and the word-posteriors from lattices, into an "adapted-RNNLM". When rescoring a particular lattice $\mathcal{L}$ (corresponding to an utterance) during the 2nd-pass rescoring, we compute a sum of word-posteriors of all other lattices within the same conversation as unigram counts for adaptation models, and adjust the probabilities of the baseline RNNLM.

For the conversational cache model, we use an unnormalized version[1] of Equation (3) to adjust the RNNLM probabilities during rescoring. For the DNN model, we pass the unigram counts computed on 1st-pass decoded word lattices (holding out the current utterance) in a conversation into the DNN and output a unigram distribution. We use the unnormalized version of Equation (1) to adjust the RNNLM probabilities during rescoring. Considering DNN models are trained on training corpora and have a smoothing effect inherently, we do not interpolate the output of DNN with a background unigram distribution as we do for cache models.

[1] For efficiency we do not normalize the output word probabilities of adapted RNNLM for rescoring tasks; the perplexities reported in this paper are all from normalized RNNLMs after adaptation for a fair comparison.

4. Experimental Setup

4.1. Datasets

We conduct experiments on three conversational speech recognition corpora of different languages, namely Switchboard (*SWBD*), Callhome Spanish (*Spanish*) and Callhome Egyptian (*Egyptian*). For SWBD and Spanish, we use additional Fisher data for building language models. For SWBD, we report results on the full HUB5'00 evaluation set ("eval'00 (all)" in all tables) and its "SWBD" subset ("eval'00 (swb)" in all tables). We also report results on the RT03 test set (LDC2007S10) of SWBD corpus. The sizes of these corpora are shown in Table 1.

Table 1: *Sizes of Different Corpora*

Corpus	Acoustic data (hours)	Text (words)
SWBD	2040	24.4 M
Spanish	167	1.74 M
Egyptian	14	0.5 M

4.2. Setups

We use the open source ASR toolkit Kaldi [23] to build all our ASR systems. For acoustic models, we use the lattice-free MMI systems described in [24], with explicit pronunciation and silence probability modeling [25]. Backstitch optimization method [26] is used during acoustic model training on SWBD. We use Kaldi-RNNLM [27] to train TDNN-LSTM [28, 29] based RNNLMs as baselines, and adapt them using cache and DNN models respectively. We use a pruned rescoring method proposed in [30] to perform a 2nd-pass lattice rescoring with the adapted RNNLMs on the 1st-pass decoded lattices.

5. Experiments

5.1. Evaluation

We evaluate our two adaptation models by rescoring and language modeling tasks on the three corpora. Table 2 shows the WERs by adapted RNNLMs and the baseline, which is an RNNLM rescored system without adaptation, on the three corpora. We can see that both adaptation models improve WERs consistently on all the datasets. For the two relatively large corpora (SWBD and Spanish), the conversational cache and DNN models obtain comparable improvements. While for the small Egyptian corpus, cache models outperform the DNN models. It is expected since the small amount of training text is insufficient to train a well generalized DNN model.

Table 2: *WERs from Cache and DNN adapted RNNLMs*

Corpus	Test set	Baseline	Cache	DNN
SWBD	eval'00 (all)	10.6	10.3	**10.2**
	eval'00 (swb)	7.1	**6.8**	**6.8**
	rt03	10.0	**9.7**	9.8
Spanish	dev	24.9	**24.6**	**24.6**
	test	21.5	**21.3**	**21.3**
Egyptian	dev	44.8	**43.8**	44.5
	test	46.4	**45.2**	46.1

Table 3: *Perplexities from Cache and DNN adapted RNNLMs*

Corpus	Test set	Baseline	Cache	DNN
SWBD	eval'00 (all)	60.9	**54.5**	56.0
	rt03	51.6	**47.2**	47.7
Spanish	dev	74.7	**65.6**	71.2
	test	71.7	**63.5**	69.4
Egyptian	dev	**48.5**	48.8	50.4
	test	**47.0**	47.2	48.5

Table 3 shows PPLs of adapted and baseline RNNLMs on the three corpora. The adaptation with conversational cache models obtains better PPLs on SWBD and Spanish, compared with DNN models. While for Egyptian, both models give similar PPLs.

5.2. Analysis of FMA and Modifications

In this section, we compare the standard FMA and its modifications using conversational cache and DNN models on the three corpora. The hyper-parameter α for WER and PPL results in Tables 4 and 5 are separately tuned on dev datasets. We use smoothing weight $\beta = 0.5$ for interpolating a cache model and the background unigram distribution. $\beta = 1$ means no smoothing. "Weight" means the weighting method described in section 3.1. We use a wighting window with size 8 centered at the current decoding utterance. The weight ratio for utterances inside and outside the window is 6. "1spk" means estimating cache models from (or applying DNN models on) utterances of one speaker instead of two.

WERs and PPLs in Table 4 show that standard FMA gives better WERs and PPLs (except for PPLs on Egyptian corpus) compared with baseline. FMA with smoothing makes both WERs and PPLs better. The weighting method further improves WERs and PPLs under the two speaker mode. And for the one speaker mode in the last row of Table 4, the WERs and PPLs are comparable to those under the two speaker mode.

Table 5 presents WERs and PPLs from DNN adpated RNNLMs. Compared with cache models, the standard FMA with DNN models gives larger performance improvements on SWBD and Spanish. With the weighting trick, adaptation with DNN models obtains better WERs and PPLs on most datasets, compared with no weighting. Similar to cache models, DNN adapted RNNLMs under one or two speakers mode yield comparable results. This indicates that both models can be applied to real scenario when only one speaker's utterances exist.

5.3. Comparison of Word-Poseriors and 1-best Hypotheses

In all the experiments above, the two adaptation models are based on word-posteriors of 1st-pass decoded lattices. An alternative approach is to use the 1-best hypotheses of 1st-pass decoding.

We compare WERs using word-posteriors with those using 1-best hypotheses on the three corpora in Table 6. In general, for both cache and DNN models on most test sets, performances using lattice posteriors are better than (or on par with) those

Table 4: *WERs and PPLs of Cache based FMA and Modifications*

Method	WER							Perplexity					
	SWBD			Spanish		Egyptian		SWBD		Spanish		Egyptian	
	eval'00(all)	eval'00(swb)	rt03	dev	test	dev	test	eval'00(all)	rt03	dev	test	dev	test
Baseline	10.6	7.1	10.0	24.9	21.5	44.8	46.4	60.9	51.6	74.4	71.7	**48.5**	**47.0**
FMA $\beta = 1$	10.5	7.1	9.9	24.8	21.4	44.2	45.7	62.0	52.4	73.3	71.0	50.9	49.2
FMA $\beta = 0.5$	10.3	6.9	9.8	24.7	21.4	44.0	45.3	55.8	48.0	68.1	65.8	49.0	47.4
FMA+Weight $\beta = 0.5$	**10.2**	6.9	**9.7**	**24.6**	**21.3**	**43.8**	**45.2**	**54.5**	**47.2**	66.5	**64.3**	48.8	47.2
FMA+Weight (1spk) $\beta = 0.5$	10.3	**6.8**	**9.7**	24.7	**21.3**	44.0	45.5	56.5	48.8	**66.3**	**64.3**	49.4	47.6

Table 5: *WERs and PPLs of DNN based FMA and Modifications* ($\beta = 1$)

Method	WER							Perplexity					
	SWBD			Spanish		Egyptian		SWBD		Spanish		Egyptian	
	eval'00(all)	eval'00(swb)	rt03	dev	test	dev	test	eval'00(all)	rt03	dev	test	dev	test
Baseline	10.6	7.1	10.0	24.9	21.5	44.8	46.4	60.9	51.6	74.4	71.7	**48.5**	**47.0**
FMA	10.3	**6.7**	9.9	**24.6**	**21.3**	**44.5**	46.2	57.2	48.4	**71.2**	**69.4**	50.5	48.5
FMA+Weight	**10.2**	6.8	**9.8**	**24.6**	**21.3**	**44.5**	**46.1**	**56.0**	**47.7**	71.3	69.5	50.4	48.5
FMA+Weight (1spk)	10.3	6.9	**9.8**	24.7	**21.3**	**44.5**	46.2	56.5	48.2	71.9	69.9	50.6	48.7

Table 6: *WERs from Cache and DNN adapted RNNLMs based on Word-posteriors vs. 1-best Hypotheses*

Corpus	Test set	Baseline	Cache		DNN	
			Posts	1-best	Posts	1-best
SWBD	eval'00(all)	10.6	**10.2**	10.3	**10.2**	10.3
	eval'00(swb)	7.1	**6.8**	7.0	**6.8**	6.9
	rt03	10.0	**9.7**	**9.7**	9.8	9.8
Spanish	dev	24.9	**24.6**	**24.6**	**24.6**	**24.6**
	test	21.5	21.3	**21.2**	21.3	21.3
Egyptian	dev	44.8	**43.8**	**43.8**	44.5	44.5
	test	46.4	**45.2**	**45.2**	46.1	46.2

using 1-best hypotheses. For Spanish test set, adaptation using 1-best hypotheses for estimating the cache model gives a better WER than using word-posteriors. The adaptation methods for cache and DNN models are the FMA with the two modification methods and the FMA with the weighting method, respectively. Both are under two speakers mode.

5.4. Correlation between PPLs and WERs

Although perplexity is a good measure for language modeling performance, a common observation is that there is not a strong correlation between PPLs and WERs [31] in speech recognition tasks. In particular, small improvement in PPLs does not necessarily translate to improvement in WERs, and vice versa.

We have observed this phenomenon in Egyptian corpus, and one explanation is the following: the direct effect of our adaptation method is to boost the probability of a word that is observed to be more frequent in conversations compared to its background unigram probability. This guarantees that, if such word appears in the current lattice, it has a very high likelihood to appear in the final decoded result. If a wrong word is boosted, little negative effect would take place if the word is not in the lattice. However, in the PPL computation, since adapted probabilities are renormalized, a strong boost of a wrong word can have a much larger negative effect on probabilities of correct words, thus the PPLs as well.

6. Conclusion and Future Work

In this work, we propose conversational cache and DNN adaptation models for RNNLMs to capture topic effects and long-distance triggers for conversational speech recognition. Experiments on SWBD and Spanish corpora show consistent WER and PPL improvements by both models. We observe 3.9% relative WER reduction and 10.5% PPL reduction on the full eval2000 dataset of SWBD, and obtain 5.6% relative WER reduction on the subset of eval2000. For Egyptian corpus, both adaptation models obtain WER improvements. In general, compared with DNN models, conversational cache models yield comparable improvement on SWBD and Spanish corpora while perform better on Egyptian corpus.

In the future, we plan to explore higher order cache and trigger models. We also would like to further investigate into DNNs to better model long-distance trigger effects.

7. Acknowledgements

This work was partially supported by DARPA LORELEI award number HR0011-15-2-0024,NSF Grant No CRI-1513128 and IARPA MATERIAL award number FA8650-17-C-9115. The authors thank Tongfei Chen for helpful discussions and drawing the figure.

8. References

[1] J. T. Goodman, "A bit of progress in language modeling," *Computer Speech & Language*, vol. 15, no. 4, pp. 403–434, 2001.

[2] T. Mikolov, M. Karafiát, L. Burget, J. Černocký, and S. Khudanpur, "Recurrent neural network based language model," in

Eleventh Annual Conference of the International Speech Communication Association, 2010.

[3] T. Mikolov, S. Kombrink, A. Deoras, L. Burget, and J. Cernocky, "Rnnlm-recurrent neural network language modeling toolkit," in *Proc. of the 2011 ASRU Workshop*, 2011, pp. 196–201.

[4] M. Sundermeyer, R. Schlüter, and H. Ney, "Lstm neural networks for language modeling," in *Thirteenth Annual Conference of the International Speech Communication Association*, 2012.

[5] T. He, Y. Zhang, J. Droppo, and K. Yu, "On training bi-directional neural network language model with noise contrastive estimation," in *Chinese Spoken Language Processing (ISCSLP), 2016 10th International Symposium on*. IEEE, 2016, pp. 1–5.

[6] X. Chen, A. Ragni, X. Liu, and M. J. Gales, "Investigating bidirectional recurrent neural network language models for speech recognition," *Proc. ICSA INTERSPEECH*, 2017.

[7] R. Kuhn and R. De Mori, "A cache-based natural language model for speech recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 12, no. 6, pp. 570–583, 1990.

[8] J. Kupiec, "Probabilistic models of short and long distance word dependencies in running text," in *Proceedings of the workshop on Speech and Natural Language*. Association for Computational Linguistics, 1989, pp. 290–295.

[9] R. Lau, R. Rosenfeld, and S. Roukos, "Trigger-based language models: A maximum entropy approach," in *Acoustics, Speech, and Signal Processing, 1993. ICASSP-93., 1993 IEEE International Conference on*, vol. 2. IEEE, 1993, pp. 45–48.

[10] N. Singh-Miller and M. Collins, "Trigger-based language modeling using a loss-sensitive perceptron algorithm," in *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, vol. 4. IEEE, 2007, pp. IV–25.

[11] R. Kneser, J. Peters, and D. Klakow, "Language model adaptation using dynamic marginals," in *Fifth European Conference on Speech Communication and Technology*, 1997.

[12] F. Jelinek, B. Merialdo, S. Roukos, and M. Strauss, "A dynamic language model for speech recognition," in *Speech and Natural Language: Proceedings of a Workshop Held at Pacific Grove, California, February 19-22, 1991*, 1991.

[13] E. Grave, A. Joulin, and N. Usunier, "Improving neural language models with a continuous cache," *arXiv preprint arXiv:1612.04426*, 2016.

[14] E. Grave, M. M. Cisse, and A. Joulin, "Unbounded cache model for online language modeling with open vocabulary," in *Advances in Neural Information Processing Systems*, 2017, pp. 6044–6054.

[15] X. Chen, T. Tan, X. Liu, P. Lanchantin, M. Wan, M. J. Gales, and P. C. Woodland, "Recurrent neural network language model adaptation for multi-genre broadcast speech recognition," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[16] S. R. Gangireddy, P. Swietojanski, P. Bell, and S. Renals, "Unsupervised adaptation of recurrent neural network language models." in *Interspeech*, 2016, pp. 2333–2337.

[17] T. Mikolov and G. Zweig, "Context dependent recurrent neural network language model." *SLT*, vol. 12, pp. 234–239, 2012.

[18] M. Ma, M. Nirschl, F. Biadsy, and S. Kumar, "Approaches for neural-network language model adaptation," *Proc. Interspeech 2017*, pp. 259–263, 2017.

[19] M. Singh, Y. Oualil, and D. Klakow, "Approximated and domain-adapted lstm language models for first-pass decoding in speech recognition," *Proc. Interspeech 2017*, pp. 2720–2724, 2017.

[20] F. Seide, G. Li, and D. Yu, "Conversational speech transcription using context-dependent deep neural networks," in *Twelfth Annual Conference of the International Speech Communication Association*, 2011.

[21] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[22] D. Yu and L. Deng, *AUTOMATIC SPEECH RECOGNITION.* Springer.

[23] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The kaldi speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding*, no. EPFL-CONF-192584. IEEE Signal Processing Society, 2011.

[24] D. Povey, V. Peddinti, D. Galvez, P. Ghahremani, V. Manohar, X. Na, Y. Wang, and S. Khudanpur, "Purely sequence-trained neural networks for asr based on lattice-free mmi." in *INTERSPEECH*, 2016, pp. 2751–2755.

[25] G. Chen, H. Xu, M. Wu, D. Povey, and S. Khudanpur, "Pronunciation and silence probability modeling for asr," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[26] Y. Wang, V. Peddinti, H. Xu, X. Zhang, D. Povey, and S. Khudanpur, "Backstitch: Counteracting finite-sample bias via negative steps," *Proc. Interspeech 2017*, pp. 1631–1635, 2017.

[27] H. Xu, K. Li, Y. Wang, J. Wang, S. Kang, X. Chen, D. Povey, and S. Khudanpur, "Neural network language modeling with letter-based features and importance sampling," in *Acoustics, Speech and Signal Processing (ICASSP), 2018 IEEE International Conference on*. IEEE, 2018.

[28] V. Peddinti, Y. Wang, D. Povey, and S. Khudanpur, "Low latency acoustic modeling using temporal convolution and lstms," *IEEE Signal Processing Letters*, vol. 25, no. 3, pp. 373–377, 2018.

[29] D. Povey, H. Hadian, P. Ghahremani, K. Li, and S. Khudanpur, "A time-restricted self-attention layer for asr," in *Acoustics, Speech and Signal Processing (ICASSP), 2018 IEEE International Conference on*. IEEE, 2018.

[30] H. Xu, T. Chen, D. Gao, Y. Wang, K. Li, N. Goel, Y. Carmiel, D. Povey, and S. Khudanpur, "A pruned rnnlm lattice-rescoring algorithm for aumatic speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2018 IEEE International Conference on*. IEEE, 2018.

[31] D. Klakow and J. Peters, "Testing the correlation of word error rate and perplexity," *Speech Communication*, vol. 38, no. 1-2, pp. 19–28, 2002.