INTERSPEECH 2017
August 20–24, 2017, Stockholm, Sweden

Approximated and domain-adapted LSTM language models for first-pass decoding in speech recognition

Mittul Singh[1,2], *Youssef Oualil*[1], *Dietrich Klakow*[1,2]

[1]Spoken Language Systems (LSV)
[2]Saarbrücken Graduate School of Computer Science, Saarland Informatics Campus
Saarland University, Saarbrücken, Germany
firstname.lastname@lsv.uni-saarland.de

Abstract

Traditionally, short-range Language Models (LMs) like the conventional *n*-gram models have been used for language model adaptation. Recent work has improved performance for such tasks using adapted long-span models like Recurrent Neural Network LMs (RNNLMs). With the first pass performed using a large background *n*-gram LM, the adapted RNNLMs are mostly used to rescore lattices or *N*-best lists, as a second step in the decoding process. Ideally, these adapted RNNLMs should be applied for first-pass decoding. Thus, we introduce two ways of applying adapted long-short-term-memory (LSTM) based RNNLMs for first-pass decoding. Using available techniques to convert LSTMs to approximated versions for first-pass decoding, we compare approximated LSTMs adapted in a Fast Marginal Adaptation framework (FMA) and an approximated version of architecture-based-adaptation of LSTM. On a conversational speech recognition task, these differently approximated and adapted LSTMs combined with a trigram LM outperform other adapted and unadapted LMs. Here, the architecture-adapted LSTM combination obtains a 35.9 % word error rate (WER) and is outperformed by FMA-based LSTM combination obtaining the overall lowest WER of 34.4 %.
Index Terms: first-pass, adaptation, approximation, LSTM

1. Introduction

Language model adaptation has been well studied using the conventional *n*-gram models [1, 2, 3] that have a short-range context. Recently, Deena et al. [4] have improved upon the short-range models by using adapted RNN-based Language Models (RNNLMs) [5]. They use these LMs to rescore lattices or *N*-best lists as a second step in the decoding process with the first pass performed on a large background *n*-gram LM. In such a setup, the second-pass LM cannot score correct hypotheses missed in the first pass, as these hypotheses cannot be recovered at this second stage [6].

To overcome this loss of hypotheses at the second pass, adapted long-span models need to be applied in the first pass. For this purpose, we introduce two such ways of applying adapted long-short-term-memory (LSTM) based RNNLMs [7]. In the first method, we use existing techniques [6, 8] for approximating LSTM to *n*-gram LMs and use this approximated LM in the Fast Marginal Adaptation (FMA) framework to create an adapted FMA-based LSTM LM. In contrast to this *extrinsic* adaptation of an approximated LSTM LM, we design an *intrinsic* approach where a model-based adaptation is applied to the LSTM network architecture and then approximated by an *n*-gram LM for first pass-decoding. To the best of our knowledge, we are the first to apply adapted long-span models for first-pass decoding in a domain-adaptation-based speech recognition task.

We begin by discussing our work in the context of prior work in Section 2. Section 3 presents different adaptation techniques for LSTM-based language models. Then, Section 4 briefly describes the *n*-gram scoring-based approximation techniques for converting the LSTMs to *n*-gram back-off LM. With Section 5 describing the experimental data, we then consider the benefits of different approximation methods and evaluate different adaptation techniques on the conversational Metalogue speech recognition task [9], in Sections 6 and 7. Finally, we conclude in Section 8.

2. Prior work

2.1. Approximating LSTM language models

In prior work, Adel et al. [8] compared different techniques — the variational approximation method [6], probability-based conversion [8] and iterative conversion [10] — for approximating RNNLMs using *n*-gram language models in a speech recognition task. Comparison of these techniques showed that the iterative conversion method performed best. However, smaller bigram-based approximations of RNNLMs outperformed the trigram-based approximation using this iterative method, making this method unfavourable for capturing long-range information in long-span models. Hence, we apply the variational approximation method and probability conversion method, which have shown improvements with larger context sizes.

2.2. First-pass decoding using RNNLMs

The above mentioned approximation methods can be considered as an off-line way of using RNNLMs for decoding, as these methods are first used to approximate RNNLMs to produce an off-line copy of *n*-gram-based language models, which is later used in the first pass.

On the other hand, the on-line methods for approximating RNNLMs directly apply RNNLMs and dynamically perform approximation during decoding [11, 12]. These on-line methods employ a cache-based mechanism, storing RNNLM states in caches and pruning the number of caches as they perform decoding. A more detailed comparison can be found in Huang et al. [12]. In our work, we only experiment with off-line first-pass decoding methods.

The work was supported by the Cluster of Excellence for Multimodal Computing and Interaction, the German Research Foundation (DFG) as part of SFB1102 and the EU FP7 Metalogue project (grant agreement number: 611073).

http://dx.doi.org/10.21437/Interspeech.2017-147

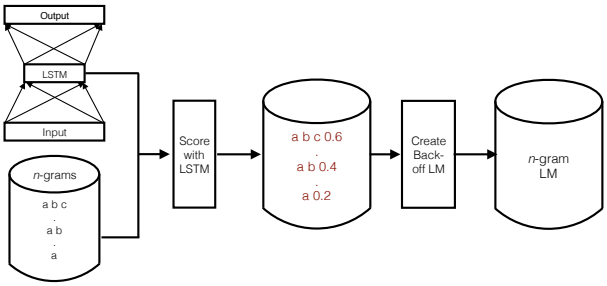

Figure 1: *The figure shows the steps followed to convert LSTM LM to an n-gram LM*

2.3. Intrinsic and extrinsic adaptation with LSTMs

Prior work [13, 14, 4] describes feature-based and model-based *intrinsic* adaptation of neural network LMs. The former utilized auxiliary features in a single training pass, whereas, the latter performed a two-pass training. These models were applied in a multi-domain adaptation setting where the auxiliary features are either available or estimated readily. In contrast, we concentrate on small single-domain adaptation corpus where such estimation is difficult and hence, only the model-based scheme is applicable for adapting LSTMs. To decode with this adapted LSTM, we approximate this model to an *n*-gram LM.

Previously, only *intrinsic* adaptation of LSTMs have been studied. Here, we also explore applying *extrinsic* technique to approximated LSTMs, further described in Section 3.1.

3. Language model adaptation for LSTMs

In this section, we describe the application details of two techniques for language model adaptation of LSTMs.

3.1. Fast Marginal Adaptation for LSTMs

To create an adapted version of LSTM ($P_{adapt}^{LSTM}(w|h)$), we incorporate an LSTM in the Fast Marginal Adaptation framework (FMA) [2]. We first approximate the LSTM trained on the background corpus (bg) to an *n*-gram LM using existing techniques (later described in Section 4) and then, apply the approximated LSTM ($P_{bg}^{LSTM}(w|h)$) in the FMA framework yielding,

$$P_{adapt}^{LSTM}(w|h) = \frac{1}{Z(h)} \left(\frac{P_{adapt}(w)}{P_{bg}(w)} \right)^{\beta} P_{bg}^{LSTM}(w|h),$$

where, $P_{adapt}(w)$ and $P_{bg}(w)$ are unigrams trained on adaptation and background corpora respectively. Here, β controls the scaling ($\frac{P_{adapt}(w)}{P_{bg}(w)}$) of approximated LSTM's probability on the adaptation data and $Z(h)$ is the normalization constant.

3.2. Output adapted LSTM

FMA-based adaptation forms an *extrinsic* way of adapting LSTM, whereas, directly adapting LSTM is performing architecture-based adaptation, an *intrinsic* adaptation technique described in Deena et al [4]. For simplicity, we construct a three-layered LSTM neural network using this *intrinsic* method, unlike the prior work [4] that used a four-layered network.

Similarly to Deena et al [4], we train this neural network on the background corpus and perform adaptation by re-train the weights between the hidden layer and the output layer on the adaptation corpus.

4. Approximations to LSTM-based LMs

To perform first-pass decoding with above described long-span models, we approximate this model to an *n*-gram LM using a variant of the probability conversion method [8]. Previously, Adel et al [8] collected RNNLM probabilities for every word of the training text and assigned these probabilities to the associated *n*-gram. Then the probabilities of *n*-grams appearing more than once are averaged together. Next, the probabilities are normalized and smoothed to produce a backing-off *n*-gram LM. In contrast, we independently collect the probabilities on different *n*-grams instead of the word. This modification might lose capturing relevant context information but skips the averaging step, hence, speeds up the conversion. An illustration of these steps is shown in Figure 1.

Switching from scoring words to *n*-grams allows us to explore different methods of instantiating the set of *n*-grams. First, where independent *n*-grams are extracted from the data used for training the language models for the speech recognition system.

This set, however, might not cover most *n*-grams present in the test set. Hence, we apply the second method where we sample *large* amounts of text from a Kneser-Ney trigram LM [15] to obtain a good coverage of test set.

The methods described above still lack the knowledge of in-domain data available to a speech recognition system. To overcome this absence, we use the *N*-best list produced by one of our baselines of the system. These *N*-best lists are then converted to *n*-grams, which can then be scored using an LSTM.

5. Data

The speech corpus collected during the Metalogue project aims to develop a dialogue system to monitor, teach and interact with participants, debating a multi-issue bargaining topic to improve their negotiation skills. The speech data includes non-native English speakers, with a mild to strong Greek accent, debating the implementation of new anti-smoking regulations, where the speakers are 4 male and 2 female undergraduates of age between 19 and 25. The data comprises 6 debate sessions in English, with a total duration of 1 hour of speech. We use the transcriptions of 5 such sessions as the adaptation training data (18k tokens) and the 6^{th} session (11k tokens) as our test set.

Unavailability of a large anti-smoking debate text to train good language models makes the Metalogue task difficult. Though not domain-specific information but similar argumentation structures can be found in readily News text and hence, the 1996 English Broadcast News Speech transcriptions (HUB4) [16] forms a good source of background information. This corpus is divided into two parts, training (152M tokens) and validation (23M tokens) sets. The vocabulary size of these sets was restricted to around 80K most frequent words in the transcriptions, with all the out-of-vocabulary words (OOV) being replaced by a predefined unknown word symbol. Any new words in the transcriptions of the Metalogue corpus are also added to this vocabulary, making it a closed vocabulary with zero OOVs in the data.

6. Perplexity experiments

To create approximated and adapted LSTMs, we first train LSTMs using in-house GPU enabled tools on the background training corpus and then process this model using the techniques described in Section 3 and Section 4. In this section, we compare different approximated and adapted models, evalu-

Table 1: *Approximated LSTM-based Trigram model's perplexity on the adaptation test set. This approximated LSTM-based LM is constructed using different sources of n-grams displayed in the first column, followed by perplexity in the second column and size of data in millions (M) of tokens*

n-grams' Source	PPL	Size
Variational Approximation		
Sampled from LSTM	327.5	5M
Probability Conversion		
Sampled from KN3	371.6	5M
Sampled from KN3	302.3	250M
HUB4	292.4	150M
N1000	**238.4**	21M

ating these models on test set *perplexity*.

6.1. *n*-gram sources for approximated LSTMs

The LSTM model are approximated by *n*-gram LMs using variational approximation and probability-conversion methods. We redesigned the latter method to create an approximated LSTM-based LM independently of *n*-grams' sources, whereas the former inherently produces text to create *n*-gram LMs. Constructing an approximated LSTM-based trigram language model using these approximation methods, we compare these methods using different *n*-grams' sources and the corresponding *perplexity* are reported in Table 1.

For a 5 million token-size corpora, using sampled text from the LSTM obtains a better perplexity than sampled text from a Kneser-Ney smoothed trigram model (KN3). However, obtaining the latter text with KN3 is much faster than the former. For instance, due to the constrained size of GPU memory, the text sampler took nearly a week to sample five million tokens from an LSTM, whereas using KN3 to sample similar number of tokens took half an hour as a single-threaded job. So instead of slowly improving perplexity by sampling more text from LSTM, we sampled up to 250 million tokens (time to sample $\sim$ 1 day) from KN3 to improve the approximated LSTM's perplexity, which is reflected by a 7% improvement over the score of 327.5 for five million tokens sampled from the LSTM.

As an alternative sources of *n*-grams, we also used the background corpora (HUB4 with 150 M tokens) and the 1000-best lists (N1000 with 21 M tokens) created using KN3 for first-pass of decoding (Section 7.1). Using these alternate sources, approximated LSTM models further improved the perplexity over the sampled-text versions. As these alternate sources are more easily available, using these sources instead of sampled text to construct approximated LSTMs allows for a faster application of language models to first-pass decoding.

6.2. Approximate LSTMs for language model adaptation

In this section, we compare the following language model adaptation techniques for LSTMs in terms of perplexity. As baselines for this comparison, we chose the Kneser-Ney smoothed trigram model (KN3) and its fast marginal adaptation (FMA3).

LSTM$_{\mathrm{bg}}$ and **LSTM**$_{\mathrm{adapt}}$ represent three-layered LSTMs within 300 hidden units trained on background corpus HUB4 and adaptation Metalogue training corpora respectively. To perform first-pass decoding with these LMs, we construct approximate trigram versions of these LMs (as described in Section 4). To apply a simple baseline, we combine these approximated LMs linearly to perform language model adaptation, forming LSTM$_{\mathrm{bg}}$ + LSTM$_{\mathrm{adapt}}$.

Table 2: *Perplexity results on different adapted LSTM-based trigram models, constructed using various n-gram sources. Here OutAdapt-LSTM refers to LSTM$_{bg}$ with output layer re-trained on adaptation training set*

LM	HUB4	SAMPLED	N1000
n-gram LMs			
KN3	198.1	-	247.3
FMA3	202.6	-	**169.5**
Original LSTMs			
LSTM$_{\mathrm{adapt}}$	433.3	-	-
LSTM$_{\mathrm{bg}}$	181.1	-	-
OutAdapt-LSTM	**166.5**	-	-
Approximated and Adapted LSTM-based Trigram LMs			
OutAdapt-LSTM	258.4	265.9	224.0
LSTM$_{\mathrm{bg}}$ + LSTM$_{\mathrm{adapt}}$	184.2	202.6	**180.1**
FMA-LSTM	263.4	261.6	185.7
Interpolation of various adapted Trigram LMs with KN3			
FMA3 + KN3	184.4	-	166.1
OutAdapt-LSTM + KN3	174.6	178.0	**141.4**
LSTM$_{\mathrm{bg}}$ + LSTM$_{\mathrm{adapt}}$ + KN3	164.4	180.3	148.5
FMA-LSTM + KN3	192.1	191.7	153.0

OutAdapt-LSTM is an LSTM trained on the training set with the LSTM's output layer re-trained on adaptation training set. To perform first-pass decoding with this LM, we construct an approximate version of this LM.

FMA-LSTM is constructed using LSTM$_{\mathrm{bg}}$, which is applied as described in Section 3.1.

Among the baselines and original LSTM-based LMs, OutAdapt-LSTM has the best perplexity value. However, applying an LSTM directly to first-pass decoding is prohibitively expensive and, hence, we approximate the LSTM-based LMs using three different sources of *n*-grams: the background corpus (HUB4), the text sampled using KN3 (SAMPLED) and the 1000-best lists (N1000).

Comparing the approximated adapted LSTMs, LSTM$_{\mathrm{bg}}$ + LSTM$_{\mathrm{adapt}}$ obtains the lowest perplexity among the approximated LSTM-based LMs. However, the adapted *n*-gram LM which scores N1000 *n*-grams, FMA3, obtains a lower perplexity. Only after linear interpolation with KN3 are the approximated LSTM-based LMs able to improve upon the perplexities, with OutAdapt-LSTM + KN3 achieving the lowest perplexity.

Moreover, for the different source of *n*-grams, 1000-best list based corpora shows the best results across the board because of the extra decoding-pass information already contained in these lists.

7. Speech Recognition Experiments

We compare the different approximated and adapted LSTM models on the Metalogue speech recognition task [9]. In this section, we describe the relevant system details and speech recognition experiments using these LSTM models.

7.1. Metalogue speech recognition system

The Metalogue corpus mainly contains many spontaneous speech phenomena such as repetitions, hesitations, etc., and is not used for training purposes in the speech recognizer. Therefore, we train a multi-style speech recognition system on a collection of corpus, including the 1996 English Broadcast News Speech Corpus, Voxforge[1], LibriSpeech [17] and WSJ0 [18, 19], which amounts to a total training duration of $\approx$ 1200

[1] http://www.voxforge.org

Table 3: *WERs on Metalogue speech recognition task for adapted and approximated LSTM-based trigram LMs for n-grams from HUB4 and 1000-best lists (N1000)*

LM	HUB4	N1000
Baselines		
$\text{LSTM}_{\text{adapt}}$	50.0	45.9
LSTM_{bg}	55.7	41.0
FMA3	36.5	**36.1**
Approximated and Adapted LSTM Trigram LMs		
OutAdapt-LSTM	42.9	39.0
LSTM_{bg} + $\text{LSTM}_{\text{adapt}}$	39.2	37.8
FMA-LSTM	38.8	**37.3**
Interpolation of adapted Trigram models with KN3		
FMA3 + KN3	35.6	35.0
OutAdapt-LSTM + KN3	36.0	36.3
LSTM_{bg} + $\text{LSTM}_{\text{adapt}}$ + KN3	36.2	**34.7**
FMA-LSTM + KN3	35.2	34.8

hours of training data for the acoustic model. The acoustic model was trained using the standard GMM/HMM model combined with Linear Discriminant Analysis (LDA), Maximum Likelihood Linear Transform (MLLT) estimation and Speaker Adaptive Training (SAT). All these models were trained and applied using the Kaldi toolkit [20]. Using other acoustic model training approaches such as deep neural networks is planned as a natural next step to improve our current ASR system.

7.2. Speech recognition experiments with LSTMs

The aforementioned adapted and approximated LSTM-based trigram models are applied in the above described Metalogue speech recognition system (Section 7.1). Table 3 reports *word error rate* (WER) results of these experiments using HUB4 and N1000 as*n*-gram sources.

For these experiments, we use approximated LSTMs trained on background (bg) and adaptation (adapt) training sets as the simple baselines. As a more competitive baseline, we also apply the fast-marginal-adaptation-based trigram language model to the metalogue speech recognition task.

Among the LSTM-based models, FMA-based LSTM (FMA-LSTM) obtains the lowest WERs across different sources of *n*-grams, in contrast to LSTM_{bg} + $\text{LSTM}_{\text{adapt}}$, which showed a better perplexity value. Until these approximated LSTM LMs are interpolated with a KN3, these LMs are all outperformed by the FMA3 model. This is quite similar to results obtained during our perplexity experiments.

As KN3 is interpolated with approximated LSTM models and FMA3, we observe a higher rate of improvement for LSTM models than FMA3 model. Moreover, the KN3-interpolated version of LSTMs obtain a lower word error rate than KN3-interpolated FMA3 model. We attribute this observation to KN3's ability to model short-context information which is complementary to LSTM's capabilities to leverage longer-context information than the short-context FMA3 model and hence, the interpolation with KN3 benefits the LSTM models more.

As shown in Table 3, we also observe the impact of using different *n*-gram sources. 1000-best list based *n*-grams, which are rich in first-pass decoding information, obtain lower word error rates in comparison to *n*-grams from HUB4.

Also, among each of the subcategories, FMA-based language models generally perform best for different sources of *n*-grams. Previously, Kneser et al. [2] showed that FMA generally outperforms simple linear interpolation of language models trained on background and adaptation training sets. In most cases, we observe a similar trend. When comparing approximated OutAdapt-LSTM to FMA-LSTM word error rates, the former performs worse whereas in terms of perplexity the non-approximated version performs much better than the FMA-LSTM version. We suspect this degradation in performance is due to losses during the approximation process.

Table 4: *LSTM-based 5-gram LMs interpolated with kneserney trigram (KN3) on Metalogue speech recognition task with 1000-best list as the n-grams source*

LM	N1000
OutAdapt-LSTM + KN3	35.9
LSTM_{bg} + $\text{LSTM}_{\text{adapt}}$ + KN3	35.0
FMA-LSTM + KN3	**34.4**

We note that using sampled text from KN3 as a source we observed similar language model trends. However, the sampled text as source was outperformed by *n*-grams from HUB4 and N1000 and, hence, are not reported in Table 3.

In all the above experiments, we chose *n*-grams up to a size three to be scored by LSTMs. As LSTMs are long-span models, using a short ranged *n*-grams can be a hindrance in the models performing well. To alleviate this issue, we score *n*-grams up to five built on 1000-best list with adapted LSTMs to be used in decoding. We only consider 1000-best list *n*-grams as these *n*-grams obtain the best results in our previous experiments and report the results in the Table 4. As shown in this table, the larger context mostly helps improve the speech recognition performance, with the FMA-based *5g*-LSTM interpolated with KN3 (FMA-LSTM+KN3) performing the best on the speech recognition task.

8. Conclusion

In this paper, we presented Fast Marginal Adaptation based LSTMs and adaptation-based-architecture LSTMs that used approximation techniques to apply LSTMs in first-pass decoding.

We applied the variational approximation technique and the probability conversion method and found that the latter achieved a lower perplexity faster than the former. Also, using acoustically rich *n*-grams from *N*-best list for approximations achieved the lowest perplexity values on the adaptation test set.

On a conversational speech recognition task, we compare these approximated and adapted LSTMs to other adapted and unadapted language models for first-pass decoding. As part of our future work, we further analyse the impact of these adapted first-pass decoding techniques in combination with second-pass rescoring techniques. Nevertheless, in our first-pass decoding experiments FMA-based LSTM outperformed the other LMs generally, obtaining the lowest WER of 34.4 %.

9. Acknowledgements

We would like to thank Rose Hoberman and the anonymous reviewers for their comments which helped improve this paper. Thanks also go to Andrea Fischer, Arunav Mishra and David Howcraft for proofreading.

10. References

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

[2] R. Kneser, J. Peters, and D. Klakow, "Language model adapta-

tion using dynamic marginals," in *Fifth European Conference on Speech Communication and Technology, EUROSPEECH 1997, Rhodes, Greece, September 22-25, 1997*, 1997.

[3] D. Klakow, "Language model adaptation for tiny adaptation corpora," in *INTERSPEECH 2006 - ICSLP, Ninth International Conference on Spoken Language Processing, Pittsburgh, PA, USA, September 17-21, 2006*, 2006.

[4] S. Deena, M. Hasan, M. Doulaty, O. Saz, and T. Hain, "Combining feature and model-based adaptation of rnnlms for multi-genre broadcast speech recognition," in *Interspeech 2016, 17th Annual Conference of the International Speech Communication Association, San Francisco, CA, USA, September 8-12, 2016*, 2016, pp. 2343–2347.

[5] T. Mikolov, M. Karafiát, L. Burget, J. Cernocký, and S. Khudanpur, "Recurrent neural network based language model," in *INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication Association, Makuhari, Chiba, Japan, September 26-30, 2010*, 2010, pp. 1045–1048.

[6] A. Deoras, T. Mikolov, and K. Church, "A fast re-scoring strategy to capture long-distance dependencies," in *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, 27-31 July 2011, John McIntyre Conference Centre, Edinburgh, UK, A meeting of SIGDAT, a Special Interest Group of the ACL*, 2011, pp. 1116–1127.

[7] M. Sundermeyer, R. Schlüter, and H. Ney, "LSTM neural networks for language modeling," in *INTERSPEECH 2012, 13th Annual Conference of the International Speech Communication Association, Portland, Oregon, USA, September 9-13, 2012*, 2012, pp. 194–197.

[8] H. Adel, K. Kirchhoff, N. T. Vu, D. Telaar, and T. Schultz, "Comparing approaches to convert recurrent neural networks into backoff language models for efficient decoding," in *INTERSPEECH 2014, 15th Annual Conference of the International Speech Communication Association, Singapore, September 14-18, 2014*, 2014, pp. 651–655.

[9] J. van Helvert, V. Petukhova, C. A. Stevens, H. de Weerd, D. Börner, P. van Rosmalen, J. Alexandersson, and N. Taatgen, "Observing, coaching and reflecting: Metalogue - A multimodal tutoring system with metacognitive abilities," *EAI Endorsed Transactions on Future Intelligent Educational Environments*, vol. 2, no. 6, p. e6, 2016.

[10] E. Arisoy, S. F. Chen, B. Ramabhadran, and A. Sethy, "Converting neural network language models into back-off language models for efficient decoding in automatic speech recognition," *IEEE/ACM Trans. Audio, Speech & Language Processing*, vol. 22, no. 1, pp. 184–192, 2014.

[11] G. Lecorvé and P. Motlícek, "Conversion of recurrent neural network language models to weighted finite state transducers for automatic speech recognition," in *INTERSPEECH 2012, 13th Annual Conference of the International Speech Communication Association, Portland, Oregon, USA, September 9-13, 2012*, 2012, pp. 1668–1671.

[12] Z. Huang, G. Zweig, and B. Dumoulin, "Cache based recurrent neural network language model inference for first pass speech recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2014, Florence, Italy, May 4-9, 2014*, 2014, pp. 6354–6358.

[13] J. Park, X. Liu, M. J. F. Gales, and P. C. Woodland, "Improved neural network based language modelling and adaptation," in *INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication Association, Makuhari, Chiba, Japan, September 26-30, 2010*, 2010, pp. 1041–1044.

[14] S. R. Gangireddy, P. Swietojanski, P. Bell, and S. Renals, "Unsupervised adaptation of recurrent neural network language models," in *Interspeech 2016, 17th Annual Conference of the International Speech Communication Association, San Francisco, CA, USA, September 8-12, 2016*, 2016, pp. 2333–2337.

[15] R. Kneser and H. Ney, "Improved backing-off for m-gram language modeling," in *1995 International Conference on Acoustics, Speech, and Signal Processing, ICASSP '95, Detroit, Michigan, USA, May 08-12, 1995*, 1995, pp. 181–184.

[16] D. Graff, Z. Wu, R. MacIntyre, and M. Liberman, "The 1996 broadcast news speech and language-model corpus," in *Proceedings of the DARPA Workshop on Spoken Language technology*, 1997, pp. 11–14.

[17] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2015, South Brisbane, Queensland, Australia, April 19-24, 2015*, 2015, pp. 5206–5210.

[18] D. S. Pallett, J. G. Fiscus, W. M. Fisher, J. S. Garofolo, B. A. Lund, and M. A. Przybocki, "1993 benchmark tests for the ARPA spoken language program," in *Human Language Technology, Proceedings of a Workshop held at Plainsboro, New Jerey, USA, March 8-11, 1994*, 1994.

[19] F. Kubala, "Design of the 1994 CSR benchmark tests," in *Proceedings of Spoken Language System Technology Workshop*, 1995, pp. 41–46.

[20] D. Povey, A. Ghoshal, G. Boulianne, N. Goel, M. Hannemann, Y. Qian, P. Schwarz, and G. Stemmer, "The Kaldi speech recognition toolkit," in *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*, 2011.