Interspeech 2018
2-6 September 2018, Hyderabad, India

Investigation on LSTM Recurrent N-gram Language Models for Speech Recognition

Zoltán Tüske[1,2], *Ralf Schlüter*[1], *Hermann Ney*[1]

[1] Human Language Technology and Pattern Recognition,
Computer Science Department, RWTH Aachen University, 52056 Aachen, Germany
[2] IBM Research, Thomas J. Watson Research Center, P.O. Box 704, Yorktown Heights, NY 10598

{tuske,schlueter,ney}@cs.rwth-aachen.de

Abstract

Recurrent neural networks (NN) with long short-term memory (LSTM) are the current state of the art to model long term dependencies. However, recent studies indicate that NN language models (LM) need only limited length of history to achieve excellent performance. In this paper, we extend the previous investigation on LSTM network based n-gram modeling to the domain of automatic speech recognition (ASR). First, applying recent optimization techniques and up to 6-layer LSTM networks, we improve LM perplexities by nearly 50% relative compared to classic count models on three different domains. Then, we demonstrate by experimental results that perplexities improve significantly only up to 40-grams when limiting the LM history. Nevertheless, the ASR performance saturates already around 20-grams despite across sentence modeling. Analysis indicates that the performance gain of LSTM NNLM over count models results only partially from the longer context and cross sentence modeling capabilities. Using equal context, we show that deep 4-gram LSTM can significantly outperform large interpolated count models by performing the backing off and smoothing significantly better. This observation also underlines the decreasing importance to combine state-of-the-art deep NNLM with count based model.

Index Terms: speech recognition, language-modeling, LSTM, n-gram

1. Introduction

In statistical speech recognition, the language model (LM) estimates the prior probability for strings of words, for sentences or utterances. Models usually factorize the sentence probability by chain rule using conditional dependence on the previous words (Eq. 1). These history conditioned word probabilities are estimated mostly by maximum likelihood criteria, and often approximated by assuming conditional dependence only on the previous $n-1$ words (n-gram). Traditionally, the probabilities are estimated directly, resulting in the well known count based models [1, 2, 3]; however, practical estimation is restricted to $n \in \{4, 5\}$ words even on billion word corpora. Exponential, or max-entropy models can be seen as a generalization of the count models. They generate the posteriors from a common, properly chosen feature space [4, 5]. Deep neural networks (NN) learn such features extremely well, and have become the state-of-the-art approach, leading also to considerable ASR performance gains over classic count models [6, 7]. Recurrent, nowadays long short-term memory (LSTM), NNLMs are broadly used and their structures fit naturally to sequences with variable length [8, 9, 10]. Thus, they might be capable of exploiting extreme long range dependencies, and there is no need for n-gram approximation in Eq 1.

$$p(w_1^I) = \prod_{i=1}^{I} p(w_i|w_1^{i-1}) \approx \prod_{i=1}^{I} p(w_i|w_{i-n+1}^{i-1}) \qquad (1)$$

However, there is a returning need to determine the effective length of history the actual state-of-the-art models exploit, due to the continuous progress in natural language processing [11, 12]. Performance of shallow sigmoid feed-forward (FF) network usually saturated at 4 to 10-grams [10, 12, 13]. But FF LM is indeed able to take advantage of up to 20-grams when using deep structure and rectified linear units (ReLU [14]), as has been shown in [15]. Applying the state-of-the-art LSTM approach of [16], in [17] an investigation was carried out to discover the effective memory of such recurrent LM. The authors concluded that powerful models using only 13-grams should be able to match the state of the art.

In this paper, the study of [17] is extended. Focusing also on word error rates (WER), we validate n-gram LSTM language models on three different ASR tasks (narrow-band telephone speech, broadband Skype calls, broadcast news). Besides re-optimizing our previous best LSTM models, we attempt to estimate the necessary language model history to achieve high recognition performance. Further experiments are carried out to measure, if interpolation of count and NNLM is still necessary for the best results, and if spanning context across neighboring utterances using the best scoring hypothesis is beneficial. It is generally believed, that the improvement of NNLMs results from long-span modeling capabilities. Thus, we also design experiments to quantify how much gain is related to modeling longer dependencies.

2. Experimental setup

2.1. Speech corpora

We evaluated LSTM n-grams on three different ASR tasks:

Models for English narrow-band telephone conversation are based on the standard 300 hours of Switchboard corpus (SWB-300). The lexicon size was limited to 30k. For language modeling we also used the Fisher corpora, resulting in 24 million running words. Our cross validation (CV) set was defined only on the Switchboard part, randomly selecting around 10% of the recordings. The details of the speaker adaptive and discriminatively (MPE-SA) trained acoustic model (AM) are described in [18]. Perplexities (PPL) and recognition results are reported on the complete Hub5 2000 (Hub5'00) test set.

The English broadcast news and conversation speech recognizer was build within the Quaero project [19, 20]. The speech corpus consisted of 250 hours of data. For language model training, text data was collected from 9 different sources (e.g. English Gigaword, web blogs) and consisted of about 3 billion words. Following the experimental setups in [21], a subcorpus of 50 million words was also defined using the best match-

10.21437/Interspeech.2018-2476

ing domain (e.g. transcription of acoustic data). The lexicon size was 150k. The ASR experiments were carried out with a speaker independent (SI), hybrid, 6-layer bidirectional LSTM AM trained according to the minimum phone error (MPE) rate criteria [22]. Results are reported on the project evaluation set of 2013, similar to [21, 15].

Previously we also developed speech recognition systems for the IWSLT'2016 evaluation campaign [23, 24]. The task focused on recognizing German Skype conversations. The speaker adaptive and sequence discriminative AMs were trained according to [18]. For this task, we train language models on a corpus of 1-billion words covering 11 different domains [25]. The reported perplexities (PPL) and WERs were measured on the evaluation set. Similar to the Quaero setup, a sub-corpus of 40 million running words were also selected to train NNLMs. Since the vocabulary size was 377k, word-class approximation was used on this task [24], except when doing multi-domain training of the last layer.

For all the above mentioned tasks, 4 or 5-gram Kneser-Ney (KN) smoothed count models were trained on each data source separately. The final count models were obtained by linearly interpolating those models, minimizing perplexity on the CV or development set. Furthermore, previous LSTM LM models were also available, and we used them as additional baselines in our language modeling experiments.

2.2. Implementations

All of our previous LMs were trained using simple stochastic gradient descent. Here we extended the LM training by the most recent techniques: dropout [26], Nesterov momentum accelerated Adam [27, 28, 29]. Dropout is applied to the output of the LSTM, but not to the recurrency. Similar to [30], we also added a projection layer to the recurrent connection, which allowed to increase the number of LSTM cells.

To train n-gram LSTM models, a fixed length of truncated history should be processed for each word position separately. Like in the case of feed-forward networks, the beginning of the sentence should be padded with the sentence boundary symbol. Unfortunately, the recurrent state resulting from generating the output for the previous word position cannot be re-used. On one hand this allows for stronger randomization of the training data, because each target label can be handled independently. On the other hand, this leads to a significant increase in computational cost with long n-grams. E.g. a 6-layer LSTM estimating 41-grams virtually corresponds to using a 240-layer feed-forward network. In contrast, a fully recurrent model's effective depth is not limited (equal to the number of layers times word position), but by re-using previous states, it needs to perform only a single forwarding step at each layer.

Uni-directional forward processing of the history, however, allows to speed up training by estimating conditional probabilities for the next few words as well, using a window of labels. This leads to modeling also few tokens longer n-grams, thus we call this approach jitter n-gram, similar to the approch in [31]. An example is given in Fig. 1. As can be seen, the model estimates in the same time a 4 and 5-gram model. Since the position of window of labels is picked randomly over the word sequence, the same target label can appear at different positions in the window. To train n-gram LSTM, we used e.g. 15 jitter for 40-gram, and only 2 jitter for 3-gram models. The jitter is used only during model training and was switched off when measuring perplexities and rescoring. Our final LM models are up to 6-layer LSTM networks, thus we also applied as much parallelization between layers as possible [32]. We also note that

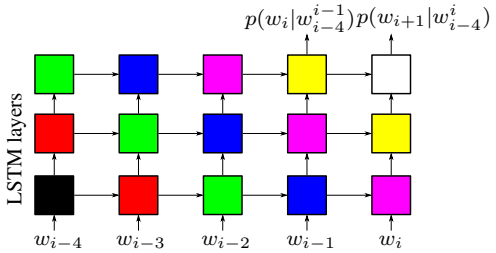

Figure 1: *Jitter n-gram: example of a 3-layer, 5-gram, recurrent NNLM with a jitter of 2. Layers are unfolded along the word sequence, and blocks with the same color are evaluated in parallel.*

Table 1: *Optimization of single-layer LSTM LM on Switchboard, PPLs are reported on the CV set.*

number of nodes in:			dropout & NAdam	PPL	#param.
word embed.	recurr. proj.	LSTM cell			
1000	-	1000	-	52.2 [18]	68M
500		500		49.7	32M
			+	47.2	
128	256	2048	-	52.1	15M
			+	45.0	

using such a deep LSTM network, word-class approximation resulted only about 10% relative speed-up on Quaero.

The final NNLMs for Quaero and IWSLT were trained using billion word corpora. Since the billion word corpus does not fit to the target domain perfectly, fine-tuning on the smaller corpus was always performed in the last step. To exploit data from various sources, we developed a log-linear interpolation methods for NNLM in [15]. To squeeze the maximum performance out of the models, this multi-domain (MD) approach was also used. We trained domain dependent output layers on the last hidden layer output of the best LSTM NNLM, without updating the hidden layers. The interpolation parameters were optimized on the development set.

NNLMs were trained on the concatenation of neighboring sentences, and minimized the perplexity directly by stochastic gradient descent. The learning rate scheduling and early stopping was controlled by the improvement of the objective function on the cross-validation, or development set. LSTMs without the n-gram constraints performed 150-300 backpropagation through time steps before update. Batch size for jitter models was set to 64, whereas classic LSTM processed only 4 sequences in parallel to achieve the best results. We also noticed during our experiments that n-gram LSTM LMs are excellent initialization to train unconstrained ones. The best, ∞-order LSTM LMs include this technique (Sec. 3.2). When the NNLMs were evaluated, estimating the conditional probabilities across utterances or sentences was optional. To measure the ASR performance of our LSTM LMs, the lattice output of our speech recognizer was processed by `rwthlm` [33].

3. Experimental results

3.1. LSTM LM optimization

In the first set of experiments we re-optimized our previous LSTM LM on the Switchboard task, (row 1 in Table 1), without n-gram constraints. As can be seen, significant, over 10% relative gain in perplexity can be achieved by restructuring the network, increasing the size of LSTM layers up to 2048, and restricting the number of parameters by a projection layer. However, introducing the projection layer was only beneficial with

Table 2: *Overall improvements of our LSTM LMs on various tasks, compared to count LM (KN) and previous LSTM baselines. +3B/+1B indicates training on billions of words, and +MD denotes multi-domain interpolation approach. NNLM results are without interpolation with count model.*

Task / Model		PPL		#param
SWB-300		CV	Hub5'00	
KN4		75.9	74.6	5M
		74.5	73.8	21M
NN	prev. baseline [18]	52.2	52.3	68M
	1-layer	45.0	46.3	15M
	3-layer	39.6	42.1	25M
	6-layer	38.4	40.8	38M
Quaero-EN		dev	eval	
KN4 (3B)		134.3	132.1	559M
NN	prev. baseline [21]	100.5	106.1	160M
	1-layer	89.6	92.6	53M
	6-layer	73.1	76.2	76M
	+3B	69.3	71.9	
	+MD	68.9	71.6	
IWSLT'16		dev	eval	
KN5 (1B)		277.5	264.3	189M
NN	prev. baseline [24]	215.2	209.3	198M
	4-layer	127.5	127.1	163M
	+1B	124.0	123.4	
	+MD	116.3	117.0	

Table 3: *Effect of n-gram initialization of classic LSTMs. Models had 6-layers and were trained on 50M-Quaero data.*

n-gram context	batch size	PPL (dev)	speed [kword/s]
40	64	75.4	5.4
∞	4	74.3	3.8
	16	76.0	9.1
$40 \rightarrow \infty$	4	73.6	3.8
	16	73.1	9.1

advanced optimization methods: NAdam and 10% dropout rate. The reduced embedding and the low-dimensional projection reduced the number of parameters drastically. We used the same model settings for each task, and as can be seen in Table 2, significant improvement was achieved in perplexities (PPL). With the help of advanced optimization, the PPLs plateaued only after 4-6 LSTM layers. The best stand-alone NNLMs showed about 50% PPL improvement over the count model, even if it was trained on billions of words. Multi-domain interpolation technique proved to be efficient on the German task, where none of the available text resources truly matched the word distribution of the development and evaluation sets.

3.2. Effect of initialization with n-gram

Training classic LSTM ("∞-gram") we noted that the best results were achieved if the batch size was limited to 4 sequences, which slowed down the training significantly 3. However, n-gram LSTMs were very robust against larger batch size. To combine the benefit of both infinitely long history and faster training, we experimented with n-gram initialization of classic LSTM LMs. As can be seen in Table 3, besides being able to train classic LSTMs in this way faster, slight PPL improvements were also observed.

3.3. Comparison of n-gram and classic recurrent LMs

In these experiments we limited the word-history a recurrent LSTM could access to perform the probability estimation. The

Table 4: *Effect of limited n-gram context on WER, measured on Hub5'00. NNLM is interpolated with count LM, λ denotes the count LM weight.*

LM		n-gram	λ	PPL	WER	
					ML-SI	MPE-SA
KN		4	1.00	74.6	15.7	14.1
+NN	[18]	∞	0.18	50.3	13.9	12.6
	optimized	4	0.17	57.9	13.9	12.4
		10	0.11	45.7	13.0	11.8
		20	0.12	42.0	12.8	11.6
		40	0.11	39.4	12.7	11.4
		∞	0.09	39.8	13.0	11.9
			0.00	40.8	13.1	12.0

PPL and WER comparison of the limited and unlimited models can be seen in Tables 4 and 5. As can be seen in Table 4, using weaker maximum likelihood (ML) AM, the large PPL differences related to longer contexts still translate to WER improvements. Increasing model context (also spanning over utterances), PPL reached a minimum around 40-grams. Reasonably good WER can already be achieved with 10-20 grams. Powerful, speaker adaptive AM did not reduce the relative gap between the different LMs. For reference, the best system achieved 11.4% and 12.0% WER on the complete Hub5e'01 and RT03s test sets. On broadcast news and German Skype calls, similar trend can be observed: the effective NNLM length is about 20-40 grams (Table 5). As a side note, confusion network based decoding ([34]) of rescored lattices improved the Quaero evaluation results further to 7.2% WER. Overall, rescoring with optimized LSTM LMs led to large, 14-19% relative, WER improvement over count models. Experimenting with short n-grams on Quaero and SWB-300, we also observed that a 4-gram LSTM NN already accounts for 50% of the WER, and 30% of the PPL improvement. This clearly shows that only part of the improvement is related to long-span modeling capability. We noted that the interpolation weight (λ) of the count model with long-span deep LSTMs is small, only around 0.1. Thus, we question if count LMs are still really necessary to obtain optimal WER ($\lambda = 0.0$). Table 4, and 5 also show that no or only slight degradation can be measured. This indicates that count models are less complementary to recent NNLMs than observed previously, e.g in [8, 21, 35].

3.4. Effect of retaining LSTM states across sentences

When the LM estimates conditional probability from long context, then the question naturally arises: is it beneficial to span language model context over sentences, and utterances? To answer this question, we initialized each rescoring step by the LSTM LM state of the previous utterance using the single-best path. The experimental results are shown in Table 6. We observed growing and significant improvement in perplexities with increasing context and modeling across sentences. This indicates that even events far in the LM history can trigger the probability estimation of the actual word. Importantly, this PPL improvement also translates to WER improvement.

3.5. Perplexity analysis

In Table 7 we analyzed the perplexities of the count n-gram LMs, and the LSTM NNLMs with variable history. We calculated order-wise perplexities by partitioning local perplexities according to the n-gram hit from the count LM as described in [35]. The following observations can be made. Similar to [21, 35], NN with long-span and across sentence modeling achieves roughly four times lower perplexities than a count

Table 5: *Effect of limiting LSTM LM history, measured on the evaluation set of Quaero and IWSLT. NNLMs are interpolated with count LM, λ denotes the count LM weight.*

	LM			n-gr.	λ	PPL	WER
Quaero-EN	KN (3B)			4	1.00	132.1	9.6
	+NN	[21]	50M	∞	0.28	92.0	8.6
		optimized		3	0.56	121.5	9.2
				4	0.37	109.9	8.9
				20	0.19	77.2	7.9
				40	0.18	73.8	8.0
				∞	0.17	72.5	7.9
			3B	4	0.33	108.1	8.8
				+MD	0.26	103.3	8.8
				20	0.16	74.8	7.9
				40	0.15	71.8	7.8
				∞	0.13	70.1	7.7
					0.00	71.6	7.8
IWSLT'16	KN (1B)			5	1.00	264.3	20.4
	+NN	[24]	40M	∞	0.32	182.8	19.0
		optimized		20	0.17	133.0	17.9
				40	0.15	124.9	17.7
				∞	0.15	121.0	17.6
			1B		0.09	114.4	17.5
					0.00	117.0	17.5

Table 6: *Effect of across-utterance language modeling. NNLMs are interpolated with count model.*

		SWB			Quaero		IWSLT	
		Hub5'00			eval		eval	
X-utt.	n-gr.:	4	10	40	20	∞	20_{40M}	∞_{1B}
	PPL	60.4	51.2	51.2	79.8	78.8	142.3	136.8
$\times$		57.9	45.7	39.4	74.8	70.1	133.0	114.4
	WER	12.5	12.0	12.1	7.9	7.9	18.4	17.9
$\times$		12.4	11.8	11.4	7.9	7.7	17.7	17.5

Table 7: *Measuring count model based order-wise perplexities on the Hub5'00 of Switchboard and Quaero evaluation sets. Percentage in parentheses indicates the rel. frequency of the given order on the corresponding set. NNLM results are w/o interpolation with count model.*

Model			n-gr.	X-utt.	PPL by KN order 1	2	3	4	Tot. PPL
SWB		KN	4	n	26408 (3%)	253 (32%)	47.4 (34%)	14.9 (28%)	74.6
		NN			19021	195	39.8	15.1	62.7
			10		7627	168	34.4	13.0	53.6
				y	6760	130	31.4	12.4	47.0
			40		5159	108	27.6	11.2	40.5
			∞	n	6478	151	35.9	13.9	53.3
				y	4866	110	27.6	11.3	40.8
Quaero	50M	KN	4	n	44736 (3%)	515 (32%)	59.8 (34%)	11.2 (28%)	163.2
		NN	3		22386	367	54.7	17.8	144.0
			4		16380	312	50.6	12.7	120.4
			∞	y	9251	215	38.7	10.3	87.7
	3B	KN	4	n	86050 (1%)	1143 (22%)	160.9 (35%)	25.9 (40%)	132.1
		NN			46389	750	122.0	25.9	107.8
			∞		25809	594	98.2	20.8	85.8
				y	21308	476	86.1	19.2	75.2

model for lower order. This is due to the fact that NNLM does not necessarily ignore words from the history and backs off only to a single, lower order context in case of an unseen n-gram. The longer the history the larger the gain we observed for lower orders, but the improvement at higher orders was limited and saturated at 10-20 grams. Transfer of LSTM states across utterances decreases the word confusion for low order and result in large perplexity gain. The NNs automatically summarized the previous utterances into a single vector similar to the ideas behind trigger, cache LM, or bag of words [36, 37]. Most importantly and in contrast to [21], our optimized LSTM is able to significantly outperform count models even for higher orders, and even if the models were trained on billions of words. Limiting the history of the NNLM to the same length as the count LM (4-gram), we also performed a more fair comparison: at higher order the two model perform equally. These observations are an evidence that 4-5 grams can be too short even if they were observed frequently enough to estimate the conditional probability robustly. It is also interesting to see that on average even a 3-gram NNLM could outperform a 4-gram count model (obviously not at order 4 level). When the count model backs off to bigram, a 3-gram NNLM can still use both words in the history, e.g. by backing off to skip n-gram estimations [38].

Based on the partitioned and over all perplexities we could also quantify and localize the improvement of LSTMs over classic count model. About 15% relative improvement in PPL is related to the better modeling capacity when we compare models at same history (e.g. 132.1→107.8 in Quaero). Another 30% improvement is clearly the result of longer span modeling (107.8 → 75.2). It is also worth to note, that NN gains a lot from the increasing context (up to 40-grams) where the confusion is inherently high: at word position where the short n-gram of the neighboring words has not been seen before.

4. Conclusions

In this paper we experimented with a truncated version of a state-of-the-art recurrent LSTM NNLM. By limiting its history, we were able to determine the effective dependencies considered by such model. As has been observed, perplexities plateau at around 40-grams, and for ASR purpose 20-gram language models should be satisfactory. We also showed that highly optimized long-span NNLMs can take advantage of across sentence or utterance modeling and decrease WER significantly. Additional experiments revealed the high performing NNLM might make the interpolation with count models unnecessary in the near future. Detailed analysis indicated that roughly 2/3 of the improvement achieved by current best NNLMs is related to the long-span modeling, and mostly concentrates on previously unseen short n-grams. Knowing the effective context, the computationally expensive recurrent processing of n-grams is presumably not necessary to arrive to the best results. We plan to explore the alternatives to find fast and efficient NN language models with limited history.

5. Acknowledgements

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 694537). The work reflects only the authors' views and the European Research Council Executive Agency is not responsible for any use that may be made of the information it contains. The GPU cluster used for the experiments was partially funded by Deutsche Forschungsgemeinschaft (DFG) Grant INST 222/1168-1.

6. References

[1] F. Jelinek and R. L. Mercer, "Interpolated estimation of Markov source parameters from sparse data," in *Proc. of the Workshop on Pattern Recognition in Practice*, 1980, pp. 381–397.

[2] S. M. Katz, "Estimation of probabilities from sparse data for the language model component of a speech recognizer," *IEEE Trans. on Acoustics, Speech and Signal Processing*, vol. 35, no. 3, pp. 400–401, Mar. 1987.

[3] R. Kneser and H. Ney, "Improved backing-off for m-gram language modeling," in *Proc. of IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, May 1995, pp. 181–184.

[4] R. Rosenfeld, "A maximum entropy approach to adaptive statistical language modelling," *Computer Speech & Language*, vol. 10, no. 3, pp. 187 – 228, 1996.

[5] S. F. Chen and S. M. Chu, "Enhanced word classing for model M," in *Proc. of Interspeech*, 2010, pp. 1037–1040.

[6] Y. Bengio, R. Ducharme, and P. Vincent, "A neural probabilistic language model," in *Advances in Neural Information Processing Systems (NIPS)*, vol. 13, 2000, pp. 932–938.

[7] H. Schwenk, "Continuous space language models," *Computer Speech & Language*, vol. 21, no. 3, pp. 492–518, 2007.

[8] T. Mikolov, M. Karafiat, L. Burget, J. Cernocky, and S. Khudanpur, "Recurrent neural network based language model," in *Proc. of Interspeech*, 2010, pp. 1045–1048.

[9] M. Sundermeyer, R. Schlüter, and H. Ney, "LSTM neural networks for language modeling," in *Proc. of Interspeech*, Portland, OR, USA, Sep. 2012, pp. 194–197.

[10] E. Arisoy, T. N. Sainath, B. Kingsbury, and B. Ramabhadran, "Deep neural network language models," in *Proc. of NAACL-HLT Workshop*, 2012, pp. 20–28.

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

[12] L. Hai Son, A. Allauzen, and F. Yvon, "Measuring the influence of long range dependencies with neural network language models," in *Proc. of NAACL-HLT Workshop*, 2012, pp. 1–10.

[13] M. Sundermeyer, I. Oparin, J.-L. Gauvain, B. Freiberg, R. Schlüter, and H. Ney, "Comparison of feedforward and recurrent neural network language models," in *Proc. of IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, 2013, pp. 8430–8434.

[14] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. of Int. Conf. on Machine Learning*, 2010, pp. 807–814.

[15] Z. Tüske, K. Irie, R. Schlüter, and H. Ney, "Investigation on log-linear interpolation of multi-domain neural network language model," in *Proc. of IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Shanghai, China, 2016, pp. 6005–6009.

[16] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[17] C. Chelba, M. Norouzi, and S. Bengio, "N-gram language modeling using recurrent neural network estimation," *CoRR*, 2017. [Online]. Available: http://arxiv.org/abs/1703.10724

[18] Z. Tüske, W. Michel, R. Schlüter, and H. Ney, "Parallel neural network features for improved tandem acoustic modeling," in *Proc. of Interspeech*, Aug. 2017, pp. 1651–1655.

[19] M. Nußbaum-Thom, S. Wiesler, M. Sundermeyer, C. Plahl, S. Hahn, R. Schlüter, and H. Ney, "The RWTH 2009 Quaero ASR evaluation system for English and German," in *Proc. of Interspeech*, Makuhari, Japan, 2010, pp. 1517–1520.

[20] "http://www.quaero.org."

[21] M. Sundermeyer, H. Ney, and R. Schlüter, "From feedforward to recurrent LSTM neural networks for language modeling," *IEEE/ACM Trans. on Audio, Speech, and Language Processing*, vol. 23, no. 3, pp. 517–529, Mar. 2015.

[22] D. Povey and P. Woodland, "Minimum phone error and I-smoothing for improved discriminative training," in *Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, 2002, pp. I.105–108.

[23] M. Cettolo, J. Niehues, S. Stüker, L. Bentivogli, R. Cattoni, and M. Federico, "The IWSLT 2016 evaluation campaign," in *Proc. of Int. Workshop on Spoken Language Translation*, 2016.

[24] W. Michel, Z. Tüske, M. A. B. Shaik, R. Schlüter, and H. Ney, "The RWTH Aachen LVCSR system for IWSLT-2016 German Skype conversation recognition task," in *Proc. of Int. Workshop on Spoken Language Translation*, 2016.

[25] M. A. B. Shaik, Z. Tüske, S. Wiesler, M. Nussbaum-Thom, S. Peitz, R. Schlüter, and H. Ney, "The RWTH Aachen German and English LVCSR systems for IWSLT-2013," in *Proc. of Int. Workshop on Spoken Language Translation*, 2013, pp. 120–127.

[26] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *CoRR*, 2012. [Online]. Available: http://arxiv.org/abs/1207.0580

[27] Y. Nesterov, "A method of solving a convex programming problem with convergence rate $O(1/k^2)$," in *Soviet Mathematics Doklady*, vol. 27, no. 2, 1983, pp. 372–376.

[28] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. of Int. Conf. on Learning Representations*, 2015.

[29] T. Dozat, "Incorporating Nesterov momentum into Adam," in *ICLR Workshop*, 2016.

[30] R. Jozefowicz, O. Vinyals, M. Schuster, N. Shazeer, and Y. Wu, "Exploring the limits of language modeling," *CoRR*, 2016. [Online]. Available: http://arxiv.org/abs/1602.02410

[31] A. r. Mohamed, F. Seide, D. Yu, J. Droppo, A. Stoicke, G. Zweig, and G. Penn, "Deep bi-directional recurrent networks over spectral windows," in *Proc. of IEEE Workshop on Automatic Speech Recognition and Understanding*, 2015, pp. 78–83.

[32] J. Appleyard, T. Kociský, and P. Blunsom, "Optimizing performance of recurrent neural networks on GPUs," *CoRR*, 2016. [Online]. Available: http://arxiv.org/abs/1604.01946

[33] M. Sundermeyer, R. Schlüter, and H. Ney, "`rwthlm` – The RWTH Aachen University neural network language modeling toolkit," in *Proc. of Interspeech*, 2014, pp. 2093–2097.

[34] L. Mangu, E. Brill, and A. Stolcke, "Finding consensus among words: Lattice-based word error minimization," in *Eurospeech*, 1999, pp. 495–498.

[35] I. Oparin, M. Sundermeyer, H. Ney, and J.-L. Gauvain, "Performance analysis of neural networks in combination with n-gram language models," in *Proc. of IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, 2012, pp. 5005–5008.

[36] R. Lau, R. Rosenfeld, and S. Roukos, "Trigger-based language models: a maximum entropy approach," in *Proc. of IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, vol. 2, 1993, pp. 45–48.

[37] K. Irie, R. Schlüter, and H. Ney, "Bag-of-words input for long history representation in neural network-based language models for speech recognition," in *Proc. of Interspeech*, 2015, pp. 2371–2375.

[38] H. Ney, U. Essen, and R. Kneser, "On structuring probabilistic dependencies in stochastic language modelling," *Computer Speech & Language*, vol. 8, pp. 1–38, 1994.