LSTM-Based Language Models for Spontaneous Speech Recognition

Ivan Medennikov[1,2(✉)] and Anna Bulusheva[1]

[1] STC-innovations Ltd, St. Petersburg, Russia
{medennikov,bulusheva}@speechpro.com
[2] ITMO University, St. Petersburg, Russia

Abstract. The language models (LMs) used in speech recognition to predict the next word (given the context) often rely on too short context, which leads to recognition errors. In theory, using recurrent neural networks (RNN) should solve this problem, but in practice the RNNs do not fully utilize the potential of the long context. The RNN-based language models with long short-term memory (LSTM) units take better advantage of the long context and demonstrate good results in terms of perplexity for many datasets. We used LSTM-LMs trained with regularization to rescore the recognition word lattices and obtained much lower WER as compared to the n-gram and conventional RNN-based LMs for the Russian and English languages.

Keywords: Recurrent neural networks · Long shorm-term memory · Language models · Automatic speech recognition

1 Introduction

Many speech recognition errors are due to the fact that the language model used relies on too short word context to predict the next word. For example, the modern n-gram models [1] usually operate with a context of 2–5 words. The feedforward neural network language models [2,3] always rely on a context of a fixed length, but this is not always sufficient for good prediction. In theory, this could be resolved with the help of the RNN-based language model (RNNLM) [4–6] which takes into account all preceding words. They significantly outperform the n-gram models in various ASR tasks [4,5]. But RNNs are very difficult to train because of the vanishing gradient problem; in practice, RNNs do not fully utilize the potential of the long context [7]. To overcome these difficulties, it has been proposed to apply RNNs with LSTM units [8–13]. But, like the RNNLM, they are prone to overfitting. The regularization techniques commonly used for the feedforward neural networks perform rather poorly on RNN and LSTM networks [14,15]. The RNN regularization technique proposed in [16] successfully solves this problem.

In this research, we apply LSTM language models trained with dropout regularization to rescore the recognition hypotheses. We obtained a significant word

A. Ronzhin et al. (Eds.): SPECOM 2016, LNAI 9811, pp. 469–475, 2016.
DOI: 10.1007/978-3-319-43958-7_56

error rate (WER) reduction as compared to the n-gram and conventional RNN language models for Russian and English languages.

The rest of the paper is organized as follows. In Sect. 2, we describe the LSTM and RNN regularization. In Sect. 3, we give the results of experiments on recognition of Russian and English spontaneous speech, and discuss them in Sect. 4.

2 Description of LSTM Units

In order to overcome the vanishing gradient problem for RNNs, Sepp Hochreiter and Jürgen Schmidhuber proposed RNN architecture elements called long short-term memory units [8]. A rather complex structure of LSTM (see Fig. 1) makes it possible to store long-term information effectively.

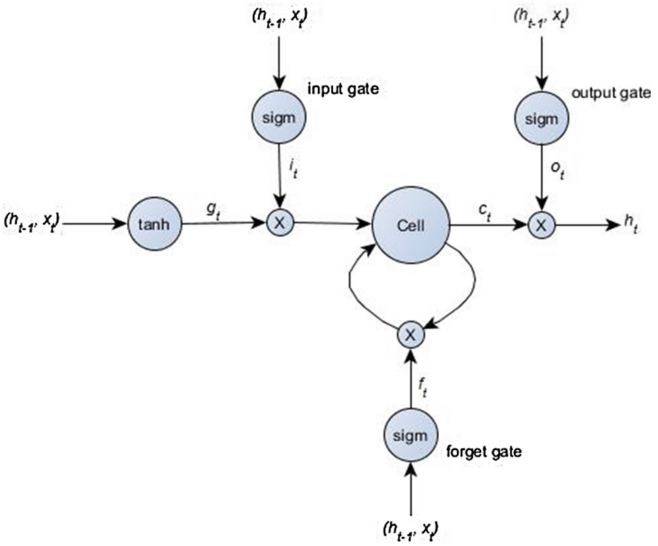

Fig. 1. Structure of LSTM

The long-term memory is implemented with the use of the memory cell vector. LSTM allows to store, change, or delete the information placed in the memory cell. This is controlled by three gates which are presented in every LSTM block. They consist of the sigmoid layer followed by the element-wise multiplication operation. The sigmoid layer outputs take values from zero to one, which indicate what fraction of a component should pass through the gate. For example, zero value means a full forbiddance to pass, while the unit value means the opposite. So, the input gate determines which information from the input is allowed to enter inside the LSTM block, the forget gate determines which

information should be removed from the memory cell. Finally, the output is determined by the cell state and the output gate values.

The LSTM is described by the equations

$$\begin{aligned}
&\mathrm{LSTM} : h_{t-1}, c_{t-1}, x_t \mapsto h_t, c_t, \\
&\begin{pmatrix} i_t \\ f_t \\ o_t \\ g_t \end{pmatrix} = \begin{pmatrix} \mathrm{sigm} \\ \mathrm{sigm} \\ \mathrm{sigm} \\ \tanh \end{pmatrix} T_{2n,4n} \begin{pmatrix} x_t \\ h_{t-1} \end{pmatrix}, \\
&c_t = f_t \odot c_{t-1} + i_t \odot g_t, \\
&h_t = o_t \odot \tanh(c_t).
\end{aligned} \tag{1}$$

Here $x_t, h_t, c_t, i_t, f_t, o_t, g_t \in \mathbb{R}^n$ denote the input vector, output vector, memory cell state and the activations of input gate, forget gate, output gate and input modulation gate at time t; $T_{2n,4n} : \mathbb{R}^{2n} \to \mathbb{R}^{4n}$ is a linear transform with a bias; $\odot$ symbol denotes element-wise multiplication; logistic (sigm) and hyperbolic tangent (tanh) activation functions are applied element-wise.

A more detailed description and an algorithm to train the LSTM can be found in [9].

2.1 Regularization with Dropout

The standard regularization techniques that exist for feedforward neural networks [14,15] perform rather poorly on RNN and LSTM networks, which commonly leads to model overfitting. The use of dropout regularization for RNN and LSTM networks is proposed in [16]. The key idea of this technique consists of applying the dropout to non-recurrent connections only. The formulas below describe this method in more detail:

$$\begin{aligned}
&\mathrm{LSTM} : h_{t-1}, c_{t-1}, x_t \mapsto h_t, c_t, \\
&\begin{pmatrix} i_t \\ f_t \\ o_t \\ g_t \end{pmatrix} = \begin{pmatrix} \mathrm{sigm} \\ \mathrm{sigm} \\ \mathrm{sigm} \\ \tanh \end{pmatrix} T_{2n,4n} \begin{pmatrix} \mathrm{D}(x_t) \\ h_{t-1} \end{pmatrix}, \\
&c_t = f_t \odot c_{t-1} + i_t \odot g_t, \\
&h_t = o_t \odot \tanh(c_t),
\end{aligned} \tag{2}$$

where D stands for the dropout operator which sets a random subset of its argument to zero. Its detailed description can be found in [16].

3 Experiments

3.1 Experiments on English Spontaneous Speech

For our experiments, we chose a training set consisted of transcriptions from the *Switchboard-1 Release 2* and *Fisher English* corpora. The 3-gram LM for the first

recognition pass was trained with modified Kneser-Ney (MKN) smoothing [1] on the transcriptions of the *Switchboard* corpus with 30 K words vocabulary. The baseline 4-gram model was built following the `swbd(s5c)` recipe. The vocabulary was around 30 k words and the model was produced by interpolation of two 4-gram models built with MKN smoothing (*Switchboard* and *Fisher*).

In order to train neural language models, we mixed all sentences and partitioned them into two parts: cross-validation(CV) (20 K sentences) and TRAIN (the remaining ones, about 2.5 M sentences). All words which were not found in the vocabulary of the 4-gram model were replaced by the ⟨UNK⟩ token. As the TEST set we chose the transcriptions of the `HUB5 2000` evaluation set. We trained RNNLM [4] and LSTM-LM on the TRAIN part, and then evaluated the perplexity on the CV and TEST parts.

For the training of RNNLM we used Tomas Mikolov's utility `rnnlm-0.4b` from the http://www.rnnlm.org site. The RNNLM topology was the following: 256 neurons in the hidden layer, 200 direct connections, 4 direct order. In order to speed up the training process, we factorized the output layer into 200 classes.

To train LSTM-LM we used the `TensorFlow` toolkit [17]. We trained two neural networks in the "medium" and "large" configurations given in [16]. LSTMs had two layers and were unrolled for 35 steps. We initialized the hidden states with zeros. Then we used the final hidden states of the current minibatch as the initial hidden state of the subsequent minibatch (successive minibatches sequentially traverse the training set). The size of each minibatch was 100. The "medium" LSTM-LM had 650 units per layer. We applied 50 % dropout to the non-recurrent connections. The "large" LSTM-LM had 1500 units per layer. We applied 65 % dropout to the non-recurrent connections. In addition, for the "large" model forget gate biases were initialized with value of 1.0.

Table 1. Experiment results on English spontaneous speech

Language model	PPL			WER, %	
	Train	CV	Test	SW	Full
4-gram	66.366	62.946	87.039	11.7	17.1
RNNLM	57.982	78.578	76.123	10.8	16.1
LSTM-LM (medium)	51.104	58.964	56.822	10.4	15.4
LSTM-LM (large)	46.033	54.821	52.892	10.1	15.2

The speech recognition experiments performed on the HUB5 2000 evaluation set were carried out with the use of the `Kaldi` speech recognition toolkit [18]. As the baseline we chose the DNN-HMM acoustic model trained with the state-level Minimum Bayes Risk (sMBR) sequence-discriminative criterion using the `nnet1` setup in the Kaldi `swbd (s5c)` recipe [19]. The recognition was performed with the trigram LM, and then the word lattices were rescored with the use of the 4-gram model. Next, we carried out the rescoring of the 100-best list with the use of neural network language models; in doing so, we calculated the language score by the formula

$$lm_{\text{rescore}} = \lambda lm_{rnn} + (1 - \lambda) lm_{4\text{gr}}, \quad (3)$$

where lm_{rnn} were calculated with the use of RNNLM, LSTM-LM (medium), LSTM-LM (large). Experiment results in terms of perplexity and WER are shown in Table 1. Note that valid PPL of the baseline 4-gram model is low due to the presence of valid texts in the training data for this LM.

3.2 Experiments on Russian Spontaneous Speech

The experiments on Russian spontaneous speech were performed in a similar way. We chose the DNN-HMM acoustic-model trained on 390 h of Russian spontaneous speech (telephone conversations). The model was trained on 120-dimensional speaker-dependent bottleneck features [20,21] extracted from the DNN trained in a speaker adaptive manner using i-vectors. We used 50-dimensional i-vectors constructed with our toolset [22]. The acoustic model training was carried out with the use of the sMBR sequence-discriminative criterion. The experiments were conducted on the same test dataset as in [20] which contained about one hour of Russian telephone conversations.

We chose two datasets to train language models. The first one consisted of the transcriptions of the AM training dataset. The second one contained a large amount (about 200 M words) of texts from Internet forum discussions, books and subtitles from the http://www.opensubtitles.org site. The baseline 3-gram language model with a vocabulary of 214 K words was built in the SRILM toolkit [23]. It was obtained by interpolation of 3-gram LMs trained on the first and second datasets using Modified Kneser-Ney smoothing. The size of this model was reduced to 4.16 M bigrams and 2.49 M trigrams with the use of pruning.

RNNLM and LSTM-LM were trained only on mixed sentences from the training dataset 1 divided into CV (15 K sentences), TEST (4 K sentences) and TRAIN (the remaining ones, 243 K sentences) parts. In order to speed up the training we utilized the vocabulary of 45 K most frequent words; all other words were replaced by the ⟨UNK⟩ token. During RNNLM training we used 256 neurons in the hidden layer and 200 classes in the output layer. LSTM-LMs were trained in the "medium" and "large" configurations described in the paper [16].

The experiments were carried out with the use of the `Kaldi` toolkit. The word lattices generated during the recognition phase with the 3-gram LM were used to extract the 100-best hypotheses list. The list was then rescored with the neural network language model. Since the 3-gram and the neural network language models contained different vocabularies we had to use unigram weights from the 3-gram LM for words that were absent in the 45 K vocabulary of RNN-based model. The results in terms of perplexity and WER are presented in Table 2. It should be noted that the 3-gram LM was trained using both dataset 1 and dataset 2 with a full vocabulary of 214 K words, while RNN-based models were trained using dataset 1 only with a reduced vocabulary of 45 K words. So, the 3-gram LM perplexity results can not be directly compared with the results of RNN-based models and reported only for reference.

Table 2. Experiment results on Russian spontaneous speech

Language model	PPL			WER, %
	Train	CV	Test	
3-gram	134.26	134.921	228.015	19.5
RNNLM	134.136	164.147	186.757	18.8
LSTM-LM (medium)	110.689	127.358	148.627	17.9
LSTM-LM (large)	105.918	124.618	146.812	17.8

4 Discussion and Conclusion

In this study we used LSTM-LM with regularization to rescore the n-best lists produced by English and Russian spontaneous speech recognition systems. This technique takes into account a longer context to predict the next word as compared with the n-gram and even with the RNN-based models. The LSTM-LMs do not suffer from the vanishing gradient problem while training.

Our experiments demonstrate that LSTM-LM gives much better results than the n-gram model and RNNLM. As compared with the n-gram model, we obtained relative WER reduction by 11.1–13.7 % for the English language and by 8.4 % for the Russian language. As compared with RNNLM, relative WER reduction is 5.6–6.5 % for the English language and 5.3 % for the Russian language.

We plan to apply LSTM-LM to other ASR tasks and study other promising language model architectures, such as character-aware neural language models [13] and end-to-end memory networks [24] in the future.

Acknowledgements. The work was financially supported by the Ministry of Education and Science of the Russian Federation. Contract 14.579.21.0121, ID RFMEFI57915X0121.

References

1. Kneser, R., Ney, H.: Improved backing-off for M-gram language modeling. In: 1995 International Conference on Acoustics, Speech and Signal Processing (ICASSP), vol. 1, pp. 181–184 (1995)
2. Bengio, Y., Ducharme, R., Vincent, P.: A neural probabilistic language model. Adv. Neural Inf. Process. Syst. **13**, 932–938 (2001)
3. Schwenk, H.: Continuous space language models. Comput. Speech Lang. **21**, 492–518 (2007)
4. Mikolov, T., Karafiat, M., Burget, L., Cernocky, J., Khudanpur, S.: Recurrent neural network based language model. In: 11th Annual Conference of the International Speech Communication Association (Interspeech), pp. 1045–1048. Makuhari (2010)
5. Kipyatkova, I., Karpov, A.: A comparison of RNN LM and FLM for Russian speech recognition. In: Ronzhin, A., Potapova, R., Fakotakis, N. (eds.) SPECOM 2015. LNCS, vol. 9319, pp. 42–50. Springer, Heidelberg (2015)

6. Mikolov, T.: Statistical language models based on neural networks. Ph.D. thesis, Brno University Technology (2012)
7. Bengio, Y., Simard, P., Frasconi, P.: Learning long-term dependencies with gradient descent is difficult. IEEE Trans. Neural Netw. **5**(2), 157–166 (1994)
8. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
9. Graves, A., Schmidhuber, J.: Framewise phoneme classification with bidirectional LSTM and other neural network architectures. Neural Netw. **18**(5–6), 602–610 (2005)
10. Józefowicz, R., Vinyals, O., Schuster, M., Shazeer, N., Wu, Y.: Exploring the limits of language modeling (2016). arXiv preprint arXiv:1602.02410
11. Sundermeyer, M., Schlüter, R., Ney, H.: LSTM neural networks for language modeling. In: 13th Annual Conference of the International Speech Communication Association (INTERSPEECH), Portland, pp. 194–197 (2012)
12. Soutner, D., Müller, L.: Application of LSTM neural networks in language modelling. In: Habernal, I. (ed.) TSD 2013. LNCS, vol. 8082, pp. 105–112. Springer, Heidelberg (2013)
13. Kim, Y., Jernite, Y., Sontag, D., Rush, A.: Character-aware neural language models (2015). arXiv preprint arXiv:1508.06615
14. Bayer, J., Osendorfer, C., Chen, N., Urban, S., van der Smagt, P.: On fast dropout and its applicability to recurrent networks (2013). arXiv preprint arXiv:1311.0701
15. Graves, A.: Generating sequences with recurrent neural networks (2013). arXiv preprint arXiv:1308.0850
16. Zaremba, W., Sutskever, I., Vinyals, O.: Recurrent neural network regularization (2014). arXiv preprint arXiv:1409.2329
17. Abadi, M., et al.: TensorFlow: large-scale machine learning on heterogeneous systems (2015). http://tensorflow.org/
18. Povey, D., et al.: The Kaldi speech recognition toolkit. In: IEEE workshop on Automatic Speech Recognition and Understanding (ASRU), Big Island, pp. 1–4 (2011)
19. Vesely, K., Ghoshal, A., Burget, L., Povey, D.: Sequence-discriminative training of deep neural networks. In: 14th Annual Conference of the International Speech Communication Association (Interspeech), Lyon, pp. 2345–2349 (2013)
20. Prudnikov, A., Medennikov, I., Mendelev, V., Korenevsky, M., Khokhlov, Y.: Improving acoustic models for Russian spontaneous speech recognition. In: Ronzhin, A., Potapova, R., Fakotakis, N. (eds.) SPECOM 2015. LNCS, vol. 9319, pp. 234–242. Springer, Heidelberg (2015)
21. Medennikov, I.P.: Speaker-dependent features for spontaneous speech recognition. Sci. Tech. J. Inf. Technol. Mech. Opt. **16**(1), 195–197 (2016). doi:10.17586/2226-1494-2016-16-1-195-197
22. Kozlov, A., Kudashev, O., Matveev, Y., Pekhovsky, T., Simonchik, K., Shulipa, A.: SVID speaker recognition system for NIST SRE 2012. In: Železný, M., Habernal, I., Ronzhin, A. (eds.) SPECOM 2013. LNCS, vol. 8113, pp. 278–285. Springer, Heidelberg (2013)
23. Stolcke, A.: SRILM – an extensible language modeling toolkit. In: Seventh International Conference on Spoken Language Processing, vol. 3, pp. 901–904 (2002)
24. Sukhbaatar, S., Szlam, A., Weston, J., Fergus, R.: End-to-end memory networks (2015). arXiv preprint arXiv:1503.08895