Language Models with RNNs for Rescoring Hypotheses of Russian ASR

Irina Kipyatkova[1,2(✉)] and Alexey Karpov[1,3]

[1] St. Petersburg Institute for Informatics and Automation
of the Russian Academy of Sciences (SPIIRAS), St. Petersburg, Russia
{kipyatkova, karpov}@iias.spb.su
[2] St. Petersburg State University of Aerospace Instrumentation (SUAI),
St. Petersburg, Russia
[3] ITMO University, St. Petersburg, Russia

Abstract. In this paper, we describe a research of recurrent neural networks (RNNs) for language modeling in large vocabulary continuous speech recognition for Russian. We experimented with recurrent neural networks with different number of units in the hidden layer. RNN-based and 3-gram language models (LMs) were trained using the text corpus of 350M words. Obtained RNN-based language models were used for N-best list rescoring for automatic continuous Russian speech recognition. We tested also a linear interpolation of RNN LMs with the baseline 3-gram LM and achieved 14 % relative reduction of the word error rate (WER) with respect to the baseline 3-gram model.

Keywords: Recurrent neural networks · Language model · Automatic speech recognition · Russian speech

1 Introduction

For automatic speech recognition (ASR) a language model (LM) is needed. The most widely used model is *n*-gram model which estimates posterior probability of the word consequence in a text. Commonly 3-gram model is employed. The usage of *n*-gram LMs with longer context can lead to the data sparseness problem. LMs based on recurrent neural networks (RNN) estimate probabilities based on all previous history that is their advantage over *n*-gram models.

In our research we used RNN LM for N-best list rescoring of automatic speech recognition (ASR) system. In Sect. 2 we give a survey of using NNs for LM creation, in Sect. 3 we describe RNN LM, in Sect. 4 we present our baseline LM, Sect. 5 gives a description of our RNN LMs, experiments on using RNN LM for N-best list rescoring for Russian speech recognition are presented in Sect. 6.

2 Related Work

The use of NN for LM training was firstly presented in [1]. RNN for language modeling was firstly used in [2]. In [3], a comparison of LMs based on feed-forward and recurrent NN was made. On the test set RNN LM showed 0.4 % absolute word error rate (WER) reduction comparing to feed-forward NN.

L. Cheng et al. (Eds.): ISNN 2016, LNCS 9719, pp. 418–425, 2016.
DOI: 10.1007/978-3-319-40663-3_48

In [4], the strategies for NN LM training on large data sets are presented: (1) reduction of training epochs; (2) reduction of number of training tokens; (3) reduction of vocabulary size; (4) reduction of size of the hidden layer; (5) parallelization. It was shown that when data are sorted by their relevance the fast convergence during training and the better overall performance are observed. A maximum entropy model trained as a part of NN LM that leads to significant reduction of computational complexity was proposed. 10 % relative reduction was obtained comparing to the baseline 4-gram model.

In [5] it was proposed to call RNN LM to compute LM score only if newly hypothesized word has a reasonable score. Also cache based RNN inference was proposed in order to reduce runtime. Three approaches for exploiting succeeding word information in RNN LMs were proposed in [6]. In order to speed up training noise contrastive estimation training was investigated in [7] for RNNLMs. Noise contrastive estimation does not require normalization at the output layer and thereby allows speeding up training. A novel RNN LM dealing with multiple time-scale contexts was presented in [8]. Several lengths of contexts were considered in one LM. In [9], paraphrastic RNN LMs, which use multiple automatically generated paraphrase variants, were investigated. In [10] Long Short-Term Memory (LSTM) NN architecture was explored for modeling English and French languages. Investigation of the jointly trained maximum entropy and RNN LMs for Code-Switching speech is presented in [11]. It was proposed to integrate part-of-speech and language identifier information in RNN LM. In [12] the discriminative method for RNN LM was proposed. As a discriminative criterion the log-likelihood ratio of the ASR hypotheses and references was used.

RNN LM for Russian was firstly used in [13]. RNN LM was trained on the text corpus containing 40M words with vocabulary size of about 100K words. An interpolation of the obtained model with the baseline 3-gram and factored LMs was carried out. The resulted LM was used for rescoring 500-best list that demonstrated 7.4 % relative improvement of WER.

Despite of the increasing popularity of usage NNs for language modeling there are only a few studies on NN-based LMs for Russian. We made a research of implementation RNNs for Russian LM creation.

3 Artificial Neural Networks for Language Modeling

We used the same structure of RNN LM as in [2]; it is presented in Fig. 1. RNN consists of an input layer x, a hidden (or context) layer s, and an output layer y. The input to the network in time t is vector $x(t)$. The vector $x(t)$ is a concatenation of vector $w(t)$, which is a current word in time t, and vector $s(t\text{-}1)$, which is output of the hidden layer obtained on the previous step. Size of $w(t)$ is equal to vocabulary size. The output layer $y(t)$ has the same size as $w(t)$ and it represents probability distribution of the next word given the previous word $w(t)$ and the context vector $s(t\text{-}1)$. The size of the hidden layer is chosen empirically and usually it consists of 30–500 units [2].

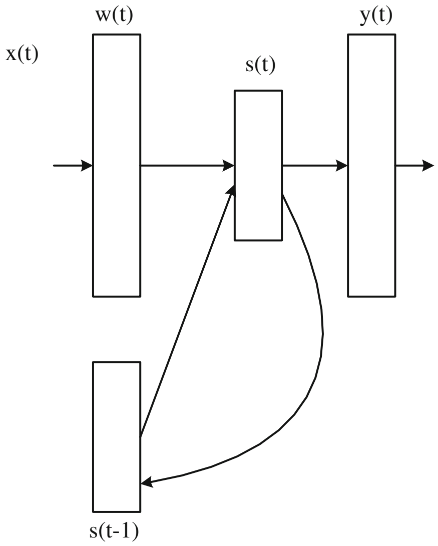

Fig. 1. General structure of the recurrent neural network.

Input, hidden, and output layers are as follows [2]:

$$x(t) = w(t) + s(t-1)$$

$$s_j(t) = f\left(\sum_i x_i(t)u_{ji}\right)$$

$$y_k(t) = g\left(\sum_j s_j(t)u_{kj}\right),$$

where *f*(*z*) is sigmoid activation function:

$$f(z) = \frac{1}{1+e^{-z}}$$

g(*z*) is softmax function:

$$g(z_m) = \frac{e^{Z_m}}{\sum_k e^{z_k}}$$

NN training is carried out in several epochs. Usually, for training the back propagation algorithm with the stochastic gradient descent is used.

In order to speed up training in [14] it was suggested to perform factorization of the output layer. Words were mapped to classes according to their frequencies. At first, probability distribution over classes was computed. Then, probability distribution for the words that belong to a specific class was computed. In this case, word probability is computed as follows:

$$P(w_i|h_i) = (P(c_i)|s(t))P(w_i|c_i, s(t)),$$

where c_i is a class of the given word, h_i is a history of the previous word.

4 Training Textual Corpus and Baseline Language Model

For the language model creation, we collected and automatically processed a Russian text corpus of a number of on-line newspapers. The procedure of preliminary text processing and normalization is described in [15]. At first, texts were divided into sentences. Then, a text written in any brackets was deleted, and sentences consisting of less than six words were also deleted. Uppercase letters were replaced by lowercase letters, if a word began from an uppercase letter. If a whole word was written by the uppercase letters, then such change was made, when the word existed in a vocabulary only. The size of the corpus after text normalization is over 350M words, and it has above 1M unique word-forms.

For the statistical text analysis, we used the SRI Language Modeling Toolkit (SRILM) [16]. During LMs creation we used the Kneser-Ney discounting method, and did not apply any *n*-gram cutoff. We created various 3-gram LMs with different vocabulary sizes, and the best speech recognition results were obtained with 150K vocabulary [17]. The perplexity measure of the baseline model was 553. So this vocabulary was chosen for further experiments with N-best list rescoring.

5 Creation of Language Models Based on Recurrent Neural Networks

For creation of RNN LM we used Recurrent Neural Network Language Modeling Toolkit (RNNLM toolkit) [18]. We made factorization of the output layer of RNN and created LMs with the number of classes equal to 100 and 500. We created models with different number of units in the hidden layer: 100, 300, and 500 [19, 20].

Then we have made a linear interpolation of the RNN LMs with the baseline 3-gram model. In this case, the probability score was computed as follows:

$$P_{IRNN}(w_i|h_i) = \lambda P_{RNN}(w_i|h_i) + (1-\lambda)P_{BL}(w_i|h_i)$$

where $P_{RNN}(w_i|h_i)$ is a probability computed by the RNN LM; $P_{BL}(w_i|h_i)$ is a probability computed by the baseline 3-gram model; λ is an interpolation coefficient.

LMs are evaluated by perplexity which is computed on held-out text date. Perplexity can be considered to be a measure of on average how many different equally most probable words can follow any given word. Lower perplexities represent better LMs [21]. Perplexities of the obtained models computed on the text corpus of 33M words are presented in Table 1. The interpolation coefficient of 1.0 means only

Table 1. Perplexities of RNN LMs interpolated with 3-gram LM.

Language model	Number of classes	Interpolation coefficients			
		0.4	0.5	0.6	1.0
RNN with 100 hidden units + 3-gram LM	100	457	465	482	981
	500	471	482	500	1074
RNN with 300 hidden units + 3-gram LM	100	457	467	484	997
	500	432	436	446	843
RNN with 500 hidden units + 3-gram LM	100	394	392	396	766
	500	417	419	428	870

RNN LM was used. In the table, we can see RNN LMs have smaller perplexities than the 3-gram LM.

6 Experiments

Architecture of the Russian ASR system with developed RNN LMs is presented on Fig. 2. The system works in 2 modes [15]: training and recognition. In the training mode, acoustic models of speech units, LMs, and phonemic vocabulary of word-forms that will be used by recognizer are created.

For training the speech recognition system we used our own corpus of spoken Russian speech Euronounce-SPIIRAS [22]. The database consists of 16,350 utterances pronounced by 50 native Russian speakers (25 male and 25 female). Each speaker pronounced more than 300 phonetically-balanced and meaningful phrases. Total duration of speech data is about 21 h. For acoustic modeling, we applied continuous density Hidden Markov Models (HMMs).

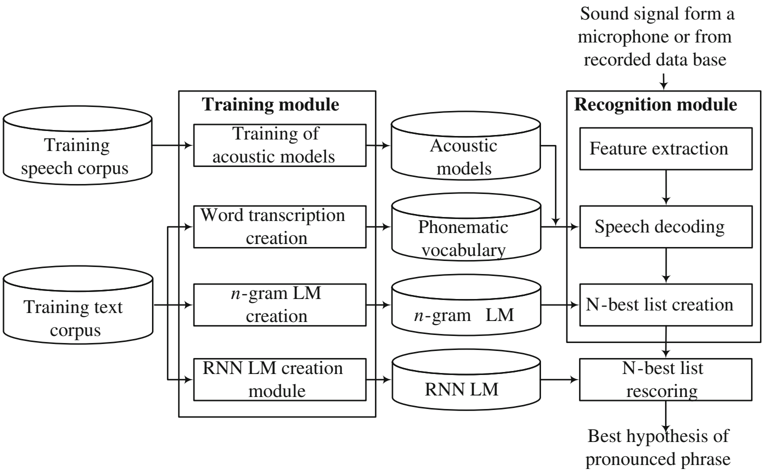

Fig. 2. Architecture of Russian ASR system with RNN LMs.

To test the ASR system we used a speech corpus that contains 500 phrases pronounced by 5 different speakers (each speaker said the same 100 phrases). The phrases were taken from the materials of an on-line newspaper that were not used in the training data.

For automatic speech recognition, we applied the open-source Julius engine ver. 4.2 [23]. At speech decoding stage, the baseline 3-gram language models were used, and N-best list of hypotheses was created. Then RNN LM was applied for rescoring obtained N-best list of hypotheses and for selection of the best recognition hypothesis for pronounced phrase.

The WER obtained with the baseline 3-gram LM was 26.54 %. We produced a 50-best list and made its rescoring using RNN LMs as well as RNN LMs interpolated (+) with the baseline model using various interpolation coefficients. Obtained results are summarized in Table 2.

Table 2. WER obtained after rescoring N-best lists with RNN LMs (%).

Language model	Number of classes	Interpolation coefficients			
		0.4	0.5	0.6	1.0
RNN with 100 hidden units + 3-gram LM	100	24.72	24.91	24.98	26.72
	500	24.78	24.83	24.83	27.45
RNN with 300 hidden units + 3-gram LM	100	24.10	24.18	24.51	25.49
	500	23.88	23.84	24.25	25.24
RNN with 500 hidden units + 3-gram LM	100	23.24	22.87	22.96	23.97
	500	23.91	23.60	23.73	24.12

In the table we can see that in the most cases the rescoring decreased the WER in comparison with the baseline model excepting the case of using RNN LMs with 100 hidden units without interpolation with the baseline model. Application of RNNs with 100 classes gave better results than RNNs with 500 classes. The lowest WER = 22.87 % was achieved using RNN LM with 500 hidden units and 100 classes interpolated with 3-gram model using the interpolation coefficient of 0.5.

Our results are consistent with those obtained in [13]. But we used training set of 350 million words that is 10 times larger set than in [13]. WER obtained in [13] with help of RNN was equal to 32.9 %. Our results are better and support the hypothesis that RNN-based LMs improve speech recognition accuracy.

7 Conclusion

In the paper, we have described the implementation of RNN LMs for rescoring N-best hypotheses lists of the ASR system. The advantage of RNN LMs over *n*-gram LMs is that they are able to store arbitrary long history of a given word. We have tried RNNs with various number of units in the hidden layer, also we tested the linear interpolation of the RNN LM with the baseline 3-gram LM. And we achieved 14 % relative reduction of WER using RNN LM with respect to the baseline model.

Acknowledgments. This research is partially supported by the Council for Grants of the President of Russia (projects No. MK-5209.2015.8 and MD-3035.2015.8), by the Russian Foundation for Basic Research (projects No. 15-07-04415 and 15-07-04322), and by the Government of the Russian Federation (grant No. 074-U01).

References

1. Schwenk, H., Gauvain, J.-L.: Training neural network language models on very large corpora. In: Proceedings of the Conference on Empirical Methods on Natural Language Processing. Association for Computational Linguistics, Vancouver, B.C., Canada, pp. 201–208 (2005)
2. Mikolov, T., Karafiat, M., Burget, L., Cernocky, J., Khudanpur, S.: Recurrent neural network based language model. In: Proceedings of INTERSPEECH 2010, vol. 2, pp. 1045–1048. Makuhari, Chiba, Japan (2010)
3. Sundermeyer, M., Oparin, I., Gauvain, J.-L., Freiberg, B., Schluter, R., Ney, H.: Comparison of feedforward and recurrent neural network language models. In: Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP), Vancouver, B.C., Canada, pp. 8430–8434 (2013)
4. Mikolov, T., Deoras, A., Povey, D., Burget L., Černocký, J.: Strategies for training large scale neural network language models. In: Proceedings of ASRU 2011, Hawaii, pp. 196–201 (2011)
5. Huang, Z., Zweig, G., Dumoulin, B.: Cache based recurrent neural network language model inference for first pass speech recognition. In: Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP) 2014, Florence, Italy, pp. 6404–6408 (2014)
6. Shi, Y., Larson, M., Wiggers, P., Jonker, C.M.: Exploiting the succeeding words in recurrent neural network. In: Proceedings of INTERSPEECH 2013, Lyon, France, pp. 632–636 (2013)
7. Chen, X., Liu, X., Gales, M.J.F., Woodland, P.C.: Recurrent neural network language model training with noise contrastive estimation for speech recognition. In: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brisbane, Australia, pp. 5411–5415 (2015)
8. Morioka, T., Iwata, T., Hori, T., Kobayashi, T.: Multiscale recurrent neural network based language model. In: Proceedings of INTERSPEECH 2015, Dresden, Germany, pp. 2366–2370 (2015)
9. Liu, X., Chen, X., Gales, M.J.F., Woodland, P.C.: Paraphrastic recurrent neural network language models. In: Proceedings IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brisbane, Australia, pp. 5406–5410 (2015)
10. Sundermeyer, M., Schlüter, R., Ney, H.: LSTM neural networks for language modeling. In: Proceedings of INTERSPEECH 2012, pp. 194–197 (2012)
11. Vu, N.T., Schultz, T.: Exploration of the impact of maximum entropy in recurrent neural network language models for code-switching speech. In: Proceedings of 1st Workshop on Computational Approaches to Code Switching, Doha, Qatar, pp. 34–41 (2014)
12. Tachioka, Y., Watanabe, S.: Discriminative method for recurrent neural network language models. In: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brisbane, Australia, pp. 5386–5390 (2015)

13. Vazhenina, D., Markov, K.: Evaluation of advanced language modeling techniques for Russian LVCSR. In: Železný, M., Habernal, I., Ronzhin, A. (eds.) SPECOM 2013. LNCS, vol. 8113, pp. 124–131. Springer, Heidelberg (2013)
14. Mikolov, T., Kombrink, S., Burget, L., Černocký, J.H., Khudanpur, S.: Extensions of recurrent neural network language model. In: Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 5528–5531 (2011)
15. Karpov, A., Markov, K., Kipyatkova, I., Vazhenina, D., Ronzhin, A.: Large vocabulary Russian speech recognition using syntactico-statistical language modeling. Speech Commun. **56**, 213–228 (2014)
16. Stolcke, A., Zheng, J., Wang, W., Abrash, V.: SRILM at sixteen: update and outlook. In: Proceedings of IEEE Automatic Speech Recognition and Understanding Workshop ASRU 2011, Waikoloa, Hawaii, USA (2011)
17. Kipyatkova, I., Karpov, A.: Lexicon size and language model order optimization for Russian LVCSR. In: Železný, M., Habernal, I., Ronzhin, A. (eds.) SPECOM 2013. LNCS, vol. 8113, pp. 219–226. Springer, Heidelberg (2013)
18. Mikolov, T., Kombrink, S., Deoras, A., Burget, L., Černocký, J.: RNNLM-recurrent neural network language modeling toolkit. In: ASRU-2011, Demo Session (2011)
19. 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)
20. Kipyatkova, I., Karpov, A.: Recurrent neural network-based language modeling for an automatic Russian speech recognition system. In: Proceedings of International Conference AINL-ISMW FRUCT, St. Petersburg, Russia, pp. 33–38 (2015)
21. Moore, G.L.: Adaptive Statistical Class-Based Language Modelling. Ph.D. thesis, Cambridge University (2001)
22. Jokisch, O., Wagner, A., Sabo, R., Jaeckel, R., Cylwik, N., Rusko, M., Ronzhin, A., Hoffmann, R.: Multilingual speech data collection for the assessment of pronunciation and prosody in a language learning system. In: Proceedings of SPECOM 2009, St. Petersburg, Russia, pp. 515–520 (2009)
23. Lee, A., Kawahara, T.: Recent development of open-source speech recognition engine julius. In: Proceedings of Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC) 2009, Sapporo, Japan, pp. 131–137 (2009)