Recurrent Neural Network-based Language Modeling for an Automatic Russian Speech Recognition System

Irina Kipyatkova
St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences (SPIIRAS),
St. Petersburg State University of Aerospace Instrumentation (SUAI)
St. Petersburg, Russia
kipyatkova@iias.spb.su

Alexey Karpov
St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences (SPIIRAS),
ITMO University
St. Petersburg, Russia
karpov@iias.spb.su

***Abstract*—In the paper, we describe a research of recurrent neural network language models for N-best list rescoring for automatic continuous Russian speech recognition. We tried recurrent neural networks with different number of units in the hidden layer. We achieved the relative word error rate reduction of 14% with respect to the baseline 3-gram model.**

I. Introduction

Automatic recognition of continuous Russian speech is a very challenging task due to several features of the language. Russian is a morphologically rich inflective language. Word formation is performed by morphemes, which carry grammatical meaning. This results in increasing the vocabulary size and perplexity of *n*-gram language models (LMs). Word order in Russian is not strictly fixed that complicates creation of LMs and decreases their efficiency. The most widely used LMs are *n*-gram LMs. These models are good enough for languages with restricted word order (for example, English) but for the Russian language they are not so efficient.

The most automatic speech recognition (ASR) systems use Hidden Markov Models (HMMs) for acoustic modeling and *n*-grams for language modeling. But at present, the usage of neural networks (NNs) is become very popular. NNs can be used for training both acoustic and language models. At acoustic modeling NNs are often combined with HMMs using hybrid and tandem methods [1], [2], [3]. For language modeling both feedforward and recurrent neural networks (RNNs) are used. In our research we used RNN LM for N-best list rescoring of automatic speech recognition (ASR) system. In Section II we give a survey of using NN for LM creation, in Section III we describe RNN LM, in Section IV we present our baseline and RNN LMs, Section V describes the architecture of ASR system with RNN LM, experiments on using RNN LM for N-best list rescoring for Russian speech recognition are presented in Section VI.

II. Related works

The use of NN for LM training was firstly presented in [4]. In that paper, the comparison of NN LM with *n*-gram LM with Kneser-Ney discounting was made. Models were trained on the corpus of 600M words. NN LM was trained not for the whole vocabulary, but for the most frequent words. An algorithm for NN training using large training corpus was proposed. According to the algorithm, instead of performing several epochs over the whole training data, a different small random subset is used at each epoch. Speech recognition was carried out using the back-off LM, and NN LM was used for the lattice rescoring. WER reduction was 0.5%.

RNN LM was firstly used in [5]; in that paper, it was proposed to merge rare words (the words, occurrence frequency of which is less than a threshold) into a special rare token for training optimization. Experiment on speech recognition was conducted using the baseline 5-gram model with Kneser-Ney discounting. RNN LM was applied for rescoring 100-best list. Perplexity reduction of RNN LM was almost 50% and the WER reduction was 18% comparing to the baseline model.

In [6], a comparison of LMs based on feedforward and recurrent NN is made. The following realizations of NN LMs were used: (1) The LIMSI shortlist feedforward NN LM software, (2) the RWTH clustered feedforward NN LM, (3) the RWTH clustered recurrent Long Short-Term Memory NN LM implementation. For NN LM training in-domain corpus of 27M words was used. For NN LM clustering 200 classes were precomputed based on relative word frequencies. Hidden layer size varied between 300 and 500 nodes, depending on the performance on the development data. NN LMs were interpolated with the *n*-gram model. Experiments on speech recognition showed that LMs based on RNN outperform feedforward NN LMs. On the test set RNN LM showed 0.4% absolute WER reduction comparing to feedforward NN.

Three approaches for exploiting succeeding word information in RNN LMs were proposed in [7]. The first approach was forward-backward model that combines RNN LMs exploiting preceding and succeeding words. In this case both forward and backward RNN LMs were created, then interpolation of the models was carried out. The second approach was the extension of a Maximum Entropy RNN LM that incorporates succeeding word information. The third approach combined LMs using two-pass alternating rescoring. In this case, an N-best list was rescored using the conventional

ISBN 978-952-68397-0-7 (paperback), ISBN 978-952-68397-1-4 (PDF)

RNN LM, and a part of the N-best list ($\alpha \cdot N, \alpha \in (0,1)$) was selected. Then the obtained N-best list was rescored using RNN LM with succeeding word information, and a new N-best list was created. These steps were repeated until the best hypothesis was obtained. The models were trained on a corpus of 37M words with 195K vocabulary. After combination of the three approaches, the WER was reduced from 16.83% to 14.44%.

In [8], the strategies for NN LM training on large data sets are presented: (1) reducing 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 [9], RNN LM was applied in the first pass decoding for Bing voice search task. In the paper 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. Using the RNN LM allowed to reduce the WER from 25.3% to 23.2%. RNN was also applied for lattice rescoring. The best results were obtained, when the lattice was created using the RNN LM interpolated with the baseline *n*-gram model in the first pass, and then rescored with the same model using interpolation weight of 0.3. In this case WER was equal to 22.7%.

A novel RNN LM dealing with multiple time-scale contexts is presented in [10]. Several lengths of contexts were considered in one LM. Experiments on recognition of large vocabulary spontaneous speech showed improvements over RNN LM in term of perplexity and word error rate.

RNN LM for Russian was firstly used in [11]. 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. Obtained LM was used for rescoring 500-best list that allowed to achieve WER relative improvement of 7.4%.

For acoustic modeling in Russian ASR, deep NN is presented in [12], [13].

III. Neural Network for Language Modeling

For language modeling both feedforward and recurrent NNs can be used. Architecture of feedforward NN LM is presents on Fig. 1 [14]. In feedforward NN, the input layer is a history of *n*-1 preceding words. Each word is associated with a vector, with length of *V* (vocabulary capacity). Only one value of the vector, which corresponds to the index of the given word, is equal to 1 and all other values are 0. Each word is mapped to its continuous space representation using linear projections. The layer formed by the concatenating the continuous word vectors is called the projection layer. The size of this layer is determined by the number of features used to represent each word. The second layer is a hidden layer. The output layer has the number of units equal to vocabulary size of the model. The main drawback of feedforward NNs is that they use preceding context of a fixed length for word prediction.

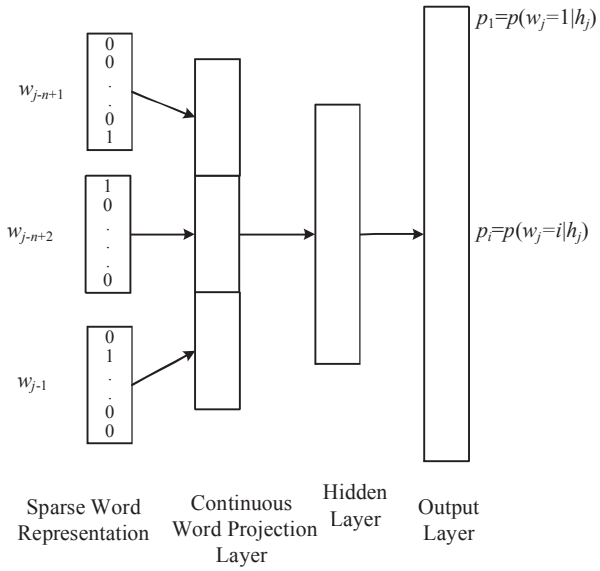

Fig. 1. Feedforward neural network architecture

Recurrent neural networks (RNN) for the first time were proposed in [15]. In RNN, the hidden layer represents all preceding history, thereby the length of the context is not restricted.

We used the same architecture of RNN LM as in [5]; it is presented on Fig. 2. RNN consists of an input layer *x*, 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*-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*-1). Size of the hidden layer is chosen empirically and usually it consists of 30-500 units [5].

Input, hidden, and output layers are as follows [5]:

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

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

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

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

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

$g(z)$ is softmax function:

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

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

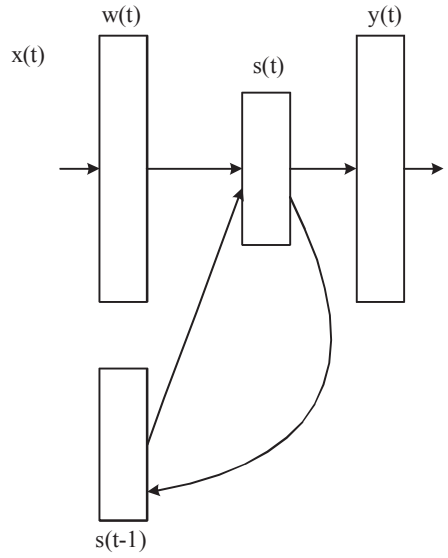

Fig. 2. Recurrent neural network architecture

IV. Creation of Language Models for Russian ASR

A. Creation of the baseline language models

For the language model creation, we used a Russian text corpus of a number of on-line newspapers [16]. The size of the corpus after text normalization and deletion of doubling or short (<5 words) sentences is over 350M words, and it has above 1M unique word-forms. As a baseline model we used 3-gram LM created using the SRI Language Modeling Toolkit (SRILM) [17]. We created 3-gram LMs with different vocabulary sizes using Kneser-Ney discounting method, obtained results are presented in Table I. The experimental setup was the same as described in Section VI.

TABLE I. Summary of the results on very large vocabulary Russian speech recognition using 3-gram LMs

Vocabulary size, K words	# *n*-grams, M	Perplexity	OOV rate, %	*n*-gram hit, %	WER, %
110	94.4	516	1.9	56.4	26.85
150	99.5	553	1.1	56.2	26.54
219	104.1	597	0.6	56.0	26.78
303	106.6	630	0.5	56.0	27.34

The best speech recognition results were obtained with 150K vocabulary [18]. So, the same 150K vocabulary was chosen for the creation of RNN LMs as well.

B. Creation of recurrent neural network language models

For creation of RNN LM we used Recurrent Neural Network Language Modeling Toolkit (RNNLM toolkit) [19]. In order to speedup training the factorization of the output layer was performed [8]. 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. We chose the number of classes equal to 100. We created three models with different number of units in the hidden layer: 100, 300, and 500 [20]. Perplexities of the obtained models computed on the text corpus of 33M words are presented in Table II.

Then we made linear interpolation of the models with the baseline 3-gram model with different interpolation coefficients. Perplexities of the obtained models are presented in Table III.

V. Russian Speech Recognition System with RNN LM

Architecture of the Russian ASR system with developed RNN LMs is presented on Fig. 3. The system works in 2 modes [16]: 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 acoustic model's training manually segmented corpus of Russian speech is used; the LMs are created based on a text corpus. Thus, the following stages of the training process can be distinguished:

- training of the acoustic models of speech units
- preliminary processing of the text material for creation of the LMs;
- creation of transcriptions for words from the collected text corpus;
- creation of the *n*-gram LMs;
- creation of the RNN LM.

Training of acoustic models of speech units is carried out with use of Russian speech corpus. Speech databases with records of large number of speakers are needed to provide speaker-independent speech recognition. Recording is performed in soundproofing room. The phrases to be pronounced are sequentially shown to a speaker. Each phrase is recorded in separate sound wav file. Then semiautomatic labeling of acoustic signal on phrases, words, and phonemes is carried out. HMMs are used for acoustic modeling, and each phoneme (speech sound) is modeled by one continuous HMM.

TABLE II. Perplexities of RNN LMs

Number units in hidden layer	Perplexity
100	981
300	997
500	766

TABLE III. PERPLEXITIES OF RNN LMS INTERPOLATED WITH 3-GRAM LM

Language model	Interpolation coefficients		
	0.4	0.5	0.6
RNN with 100 hidden units + 3-gram LM	457	465	482
RNN with 300 hidden units + 3-gram LM	457	467	484
RNN with 500 hidden units + 3-gram LM	394	392	396

A phoneme model has three states: the first state describes phoneme's start, the second state presents the middle part, and the third state is phoneme's end. HMM of a word is obtained by connection of phoneme's models from corresponding phonemic alphabet. Similarly the models of words are connected with each other, generating the models of phrases. The aim of training of the acoustic models based on HMM is to determine such model's parameters that would lead to maximum value of probability of appearance of this sequence by training sequence of observations [21].

The block of preliminary text material processing carries out the following operations. At first, texts are divided into sentences, which must begin from an uppercase letter or a digit before which inverted commas may be situated. A sentence ends by the point, exclamation, question mark or dots. It takes into account that initials and/or a surname can be placed within the sentence. Formally, it is similar to a boundary between two sentences, therefore, if the point is after a single uppercase letter, the point is not considered as the end of the sentence. Sentences containing direct and indirect speech are divided into separate sentences. These sentences can be of the following types: (1) direct speech is placed after indirect speech; (2) direct speech is before indirect speech; (3) indirect speech is within direct speech. In the first case, a formal sign for distinguishing direct and indirect speech is presence of the colon mark followed by inverted commas. In the second case, the division is made if the comma follows the inverted commas and followed by the dash. In the third case, the initial sentence is divided into three sentences: (1) from inverted commas to the corresponding comma; (2) between the first comma with dash to the second comma with dash; (3) from comma with dash to the end of the sentence.

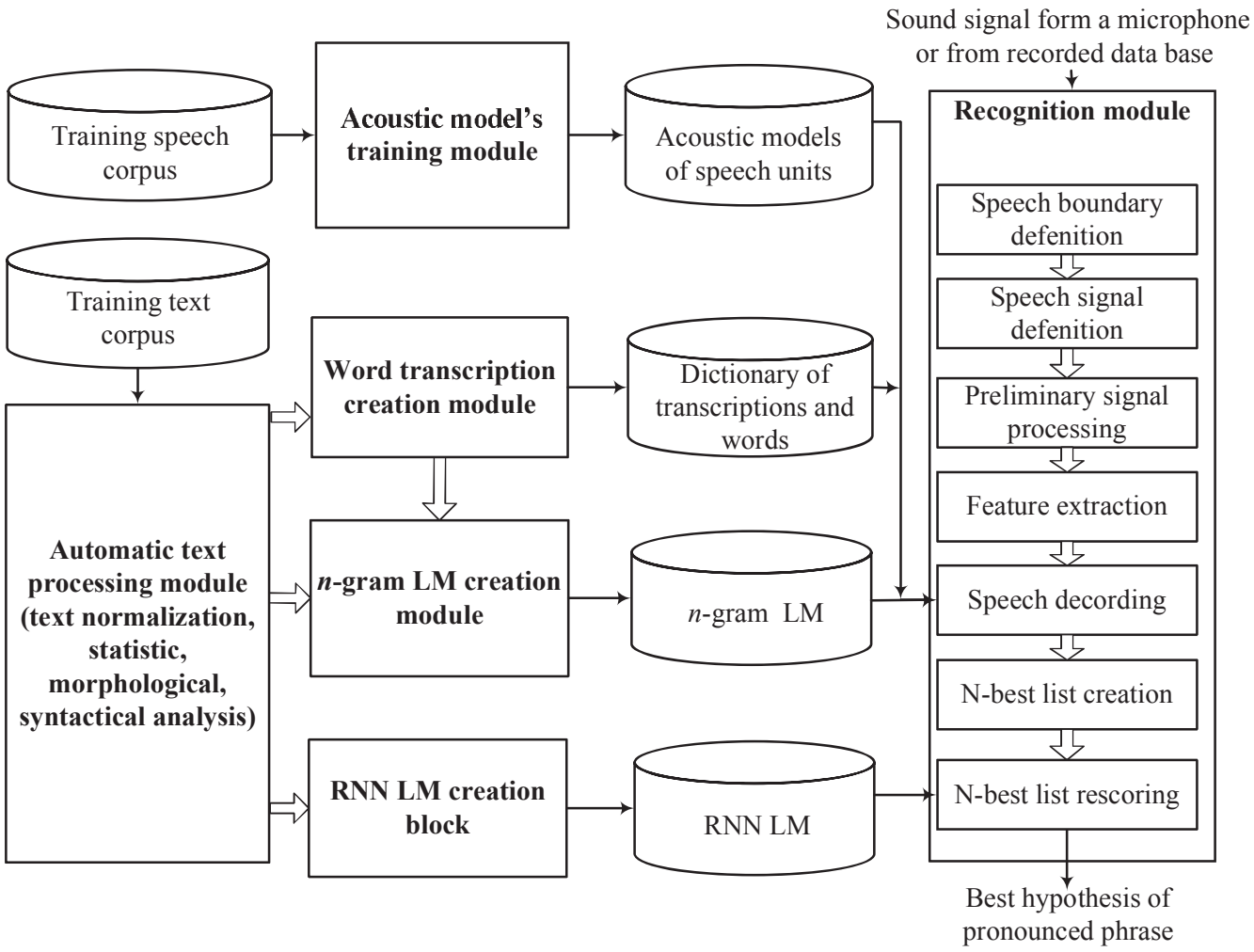

Fig. 3. Architecture of Russian ASR system with RNN LMs

Then, a text written in any brackets is deleted, and sentences consisting of less than six words are also deleted. Then punctuation marks are deleted, symbols "№" and "#" are replaced by the word "number". All numbers and digits are combined in a single class that is denoted by the symbol "№" in the resulting text. A group of digits, which can be divided by point, comma, space or dash sign is denoted as a single number. Also the symbol "№" denotes Roman numbers that are a combination of Latin letters I, V, X, L, C, D, M, which can be divided by space or dash. Internet links and E-mails are distinguished in single classes and denoted by the symbols "<>" and "<@>", respectively. Uppercase letters are replaced by lowercase letters, if a word begins from an uppercase letter. If a whole word is written by the uppercase letters, then such change is made, when the word exists in a vocabulary only. Also at this stage of training the vocabulary of words occurred in the training corpus is created.

The word transcription creation block generates transcriptions for the words from the vocabulary obtained by the block of preliminary text processing. Transcriptions are generated by application of transcribing rules to the list of words [22], [23].

The block of *n*-gram model creation performs statistic analysis of text corpus and builds a stochastic *n*-gram language model.

Block of RNN LM creation performs computation of neural network and builds a RNN LM with specified number of units in hidden layer.

In the speech recognition mode, an input speech signal is transformed into the sequence of feature vectors (Mel-Frequency Cepstral Coefficients with the 1st and 2nd order derivatives are used), then search of the most probable hypotheses is performed with the help of preliminary trained acoustic and *n*-gram language models, and N-best list of hypotheses is created. Then RNN LM is applied for rescoring obtained N-best list of hypotheses and for selection of the best recognition hypothesis for pronounced phrase.

VI. Experiments

A. Experimental Setup

For training the speech recognition system we used our own corpus of spoken Russian speech Euronounce-SPIIRAS, created in 2008-2009 in the framework of the Euro-Nounce project [24, 25]. The speech data were collected in clean acoustic conditions, with 16kHz sampling rate, 16-bit audio quality. A signal-to-noise ratio (SNR) at least 35-40 dB was provided. 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 hours.

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

For speech recognition we used Julius ver. 4.2 decoder [26]. The WER obtained with the baseline 3-gram language model was 26.54%. We produced several N-best lists with different number of hypotheses and made their rescoring using RNN LMs.

B. Experiments on rescoring N-best lists using RNN LM

Evaluation of performance of the ASR system was carried by word error rate (WER) [27]:

$$W = \frac{S + I + D}{N} \cdot 100\%$$

where S is a number of substitution errors, I is a number of insertion errors, D is a number of deletion errors, N is a total number of words is the recognizing phrase.

We made rescoring of several N-best lists using RNN LMs and RNN LMs interpolated with baseline models with different interpolation coefficients. Obtained results are summarized in Table IV. Interpolation coefficient equal to 1 means that only RNN LM was used.

From the table we can see that in the most cases rescoring decreased WER comparing to the recognition with the baseline model except the case of using RNN LMs with 100 hidden layers for 20 and 50-best list rescoring. The lowest WER=22.87 was archived using RNN LM with 500 hidden units interpolated with 3-gram model with interpolation coefficient equal to 0.5.

Fig. 4 shows the 10-best list of ASR for the Russian phrase: *"Чистота воздуха зависит и от ветра"* (*"Purity of the air also depends on the wind"*). After rescoring of this 10-best list using RNN LM with 500 hidden units interpolated with the baseline 3-gram LM, the hypothesis #4 was selected as the best one. So, after N-best list rescoring we obtained the correct hypothesis for this utterance.

V. Conclusion

Statistical *n*-gram LMs do not have efficiency for Russian ASR because of almost free word order in Russian. RNN LMs are able to store arbitrary long history of a given word that is their advantage over *n*-gram LMs. We have tried RNNs with

TABLE IV. WER obtained after rescoring N-best lists with RNN LMs (%)

Number units in hidden layer	Interpolation coefficient	10-best list	20-best list	50-best list
100	1.0	26.33	26.65	26.72
	0.6	25.13	25.06	24.98
	0.5	25.13	24.89	24.91
	0.4	25.06	24.72	24.72
300	1.0	25.41	25.30	25.49
	0.6	24.68	24.53	24.51
	0.5	24.59	24.04	24.18
	0.4	24.53	23.97	24.10
500	1.0	24.51	23.67	23.97
	0.6	23.76	23.07	22.96
	0.5	23.65	23.00	22.87
	0.4	23.82	23.26	23.24

#1 <s> чистота воздуха зависит и ответ </s>
#2 <s> чистота воздуха зависит ли ответы </s>
#3 <s> чистота воздуха зависит як ветра </s>
#4 <s> чистота воздуха зависит и от ветра </s>
#5 <s> чистота воздуха зависит як ветры </s>
#6 <s> чистота воздуха зависит и ответы </s>
#7 <s> чистота воздуха зависит не от ветра </s>
#8 <s> чистота воздуха зависели от ветра </s>
#9 <s> чистота воздуха зависит нет ветра </s>
#10 <s> чистота воздуха зависит я кветта </s>

Fig. 4. An example of N-best list of recognition hypotheses

various number of units in hidden layer, also we made the linear interpolation of the RNN LM with the baseline 3-gram LM. We achieved the relative WER reduction of 14% 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] G. Hinton et al, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups", *IEEE Signal Processing Magazine*, vol. 29, no. 6, 2012, pp. 82–97.

[2] D. P. W. Ellis, R. Singh, and S. Sivadas, "Tandem Acoustic Modeling in Large-Vocabulary Recognition", *in Proc. ICASSP*, 2001, pp. 517-520.

[3] F. Seide, G. Li, and D. Yu. "Conversational speech transcription using context-dependent deep neural networks", *in Proc. INTERSPEECH-2011*, 2011, pp. 437- 440.

[4] H. Schwenk, J.-L. Gauvain, "Training Neural Network Language Models On Very Large Corpora", *in Proc. Conference on Empirical Methods on Natural Language Processing*, 2005, pp. 201–208.

[5] T. Mikolov, M. Karafiat, L. Burget, J. Cernocky, S. Khudanpur, "Recurrent neural network based language model". *in Proc. INTERSPEECH'2010*, 2010, pp. 1045-1048.

[6] M. Sundermeyer, I. Oparin, J.-L, Gauvain, B. Freiberg, R. Schluter, H. Ney, "Comparison of Feedforward and Recurrent Neural Network Language Models", *in Proc. ICASSP'2013*, 2013, pp. 8430-8434.

[7] Y. Shi, M. Larson, P. Wiggers, C.M. Jonker, "Exploiting the Succeeding Words in Recurrent Neural Network", in Proc. *INTERSPEECH'2013*, 2013, pp. 632- 636.

[8] T. Mikolov, A. Deoras, D. Povey, L. Burget, J. Černocký, "Strategies for Training Large Scale Neural Network Language Models", *in Proc. ASRU'2011*, 2011, pp. 196-201.

[9] Z. Huang, G. Zweig, B. Dumoulin, "Cache based recurrent neural network language model inference for first pass speech recognition", *In Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 6404-6408.

[10] T. Morioka, T. Iwata, T. Hori, T. Kobayashi, "Multiscale recurrent neural network based language model", *in Proc INTERSPEECH-2015*, 2015, pp. 2366- 2370.

[11] D. Vazhenina, K. Markov, "Evaluation of advanced language modelling techniques for Russian LVCSR". *Springer International Publishing Switzerland. M. Zelezny et al. (Eds.): SPECOM 2013, LNAI 8113*, 2013, pp. 124-131.

[12] N. Tomashenko, Y. Khokhlov, "Speaker adaptation of context dependent deep neural networks based on MAP-adaptation and GMM-derived feature processing", *in Proc. INTERSPEECH-2014*, 2014, pp. 2997-3001.

[13] M.Yu. Zulkarneev, S.A. Penalov, "System of speech recognition for Russian language, using deep neural networks and finite state transducers", *Neurocomputers: development, application*, vol. 10, 2013, pp. 40-46 (in Rus).

[14] A. Gandhe, F. Metze, I. Lane, "Neural Network Language Models for Low Resource Languages", *in Proc. INTERSPEECH-2014*, 2014, pp. 2615- 2619.

[15] J.L. Elman, "Finding Structure in Time", *Cognitive Science*, vol. 14, 1990, pp. 179-211.

[16] A. Karpov, K. Markov, I. Kipyatkova, D. Vazhenina, A. Ronzhin, "Large vocabulary Russian speech recognition using syntactico-statistical language modeling", *Speech Communication*, vol. 56, 2014, pp. 213-228.

[17] A. Stolcke, J. Zheng, W. Wang, V. Abrash, "SRILM at Sixteen: Update and Outlook", *in Proc. IEEE Automatic Speech Recognition and Understanding Workshop ASRU'2011*, 2011.

[18] I. Kipyatkova, A. Karpov, "Lexicon Size and Language Model Order Optimization for Russian LVCSR", *Springer International Publishing Switzerland. M. Zelezny et al. (Eds.): SPECOM 2013, LNAI 8113*, 2013, pp. 219-226.

[19] T. Mikolov, S. Kombrink, A. Deoras, L. Burget, J. Černocký, "RNNLM - Recurrent Neural Network Language Modeling Toolkit", *in ASRU 2011 Demo Session*, 2011.

[20] I. Kipyatkova, A. Karpov. A Comparison of RNNLM and FLM for Russian speech recognition. Speech and Computer. Springer, LNAI, Vol. 9319, Proc. SPECOM-2015, 2015, pp. 42-50.

[21] I. Kipyatkova, A. Karpov, V. Verkhodanova, M. Zelezny, "Modeling of Pronunciation, Language and Nonverbal Units at Conversational Russian Speech Recognition", *International Journal of Computer Science and Applications*, vol. 10, n 1, 2013, pp. 11-30.

[22] I. Kipyatkova, A. Karpov, V. Verkhodanova, M. Zelezny, "Analysis of Long-distance Word Dependencies and Pronunciation Variability at Conversational Russian Speech Recognition", *in Proc. Federated Conference on Computer Science and Information Systems FedCSIS-2012*, 2012, pp. 719-725.

[23] L. Rabiner, B.-H. Juang, *Fundamentals of Speech Recognition,* Prentice Hall, 1993, 507 p.

[24] O. Jokisch, A. Wagner, R. Sabo, R. Jaeckel, N. Cylwik, M. Rusko, A. Ronzhin, R. Hoffmann, "Multilingual speech data collection for the assessment of pronunciation and prosody in a language learning system", *in Proc. SPECOM'2009*, 2009, pp. 515–520.

[25] A. Karpov, I. Kipyatkova, A. Ronzhin, "Very Large Vocabulary ASR for Spoken Russian with Syntactic and Morphemic Analysis", *in Proc. Interspeech'2011*, 2011, pp. 3161–3164.

[26] A. Lee, T. Kawahara, "Recent Development of Open-Source Speech Recognition Engine Julius", *in Proc. Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC 2009)*, 2009, pp.131–137.

[27] S. Young et al, *The HTK Book (for HTK Version 3.4)*, Cambridge, UK, 2009.