# SEQUENCE MEMOIZER BASED LANGUAGE MODEL FOR RUSSIAN SPEECH RECOGNITION

*Daria Vazhenina, Konstantin Markov*

The University of Aizu, Japan

**ABSTRACT**

In this paper, we propose a novel language model for Russian large vocabulary speech recognition based on sequence memoizer modeling technique. Sequence memoizer is a long span text dependency model and was initially proposed for character language modeling. Here, we use it to build word level language model (LM) in ASR. We compare its performance with recurrent neural network (RNN) LM, which also models long span word dependencies. A number of experiments were carried out using various amounts of train data and different text data arrangements. According to our experimental results, the sequence memoizer LM outperforms recurrent neural network and standard 3-gram LMs in terms of perplexity, while RNN LM achieves better word error rate. The lowest word error rate is achieved by combining all three language models together using linear interpolation.

***Index Terms***— sequence memoizer, advanced language modeling, inflective languages

## 1. INTRODUCTION

Although the underlying speech technology is mostly language-independent, differences between languages with respect to their structure and grammar have substantial effect on the automatic speech recognition (ASR) systems performance. Research in the ASR area has been traditionally focused on several main languages, such as English, French, Spanish, Chinese or Japanese, and some other languages, especially eastern European languages, have received much less attention.

The Russian language belongs to the Slavic branch of the Indo-European group of languages, which are characterized by complex mechanism of word-formation and flexible word order. Word relations within a sentence are marked by inflections and grammatical categories such as gender, number, person, case, etc. [1]. Sentence structure is not restricted by hard grammatical rules as in the English, German or Arabic languages. These two factors greatly reduce the predictive power of the conventional n-gram language models (LMs).

Nevertheless, in current Russian large vocabulary continuous speech recognition (LVCSR) systems conventional n-grams are usually used [2-6]. An improved bi-gram model was proposed in [7] where the counts of some existing n-grams are increased after syntactic analysis of the training data. Long-distance dependencies between words are identified and added as new bi-gram counts for building 2-gram and 3-gram LMs. This allowed to reduce the word error rate of a speech recognition system with dictionary of 204K words from 27.5% to 26.9%.

In conventional n-gram language models, prediction of the next word is usually conditioned just on a few preceding words, which is clearly insufficient to capture semantics. Recently, recurrent neural network (RNN) LM was proposed for better predicting sequential data using longer context dependency [8].

RNN LM allows effective processing of arbitrary length word sequences, which overcomes the main n-gram drawback - dependency on only few consecutive words. In [9], performance of this model was compared with many other state-of-the-art language models such as structured LM, random forest LM and several types of neural network LMs for the English language. It significantly outperforms all of them both in terms of perplexity and WER. In [10], RNN LM was implemented in Russian LVCSR system. Using 40M words training corpus, standalone RNN LM showed better performance than factored language model and baseline 3-gram LM. The best relative WER reduction of 7.4% was achieved using interpolation of all 3 models.

The Sequence memoizer (SM), proposed in [11], is a hierarchical Bayesian model that is able to capture long range dependences and power-law characteristics. The next word in this model is conditionally dependent on all previous words in a given sequence. Here, models are built over sequence of symbols, using ‘ ‘ (space) as word boundary and ‘.’ (dot) as sentence-end symbol. Performance of the SM language model was evaluated by perplexity using APNews dataset, which consists of 14M words and has vocabulary size of about 18K words. It showed improvement over standard 4-gram, hierarchical Pitman-Yor 4-gram and conventional neural network LMs. To our knowledge, SM hasn’t yet been used as language model in a speech recognition task.

This paper describes our implementation of the sequence memoizer for Russian LVCSR with vocabulary of 100K words. We investigated the influence of different training corpora sizes and text data arrangement on the

language model performance. It is compared with the RNN LM, which also allows to model unbounded-depth sequences. Both language modeling techniques are implemented using n-best re-scoring. While SM LM achieved the lowest perplexity, best, in terms of WER, was the interpolation of the conventional 3-gram LM with both the SM and RNN LMs.

## 2. SEQUENCE MEMOIZER

Formulation of the sequence memoizer is based on an unbounded-depth hierarchical Pitman-Yor process. Hierarchical Bayesian language models have succeeded to achieve a comparable performance to the state-of-the-art n-gram LMs smoothed with modified Kneser-Ney (MKN) smoothing. A hierarchical Pitman-Yor Process (HPYP) LM, initially introduced in [12], is a type of Bayesian language model based on the Pitman-Yor (PY) process that has been shown to improve the perplexity over the MKN smoothed n-gram LM.

In the HPYP LM, given context $\boldsymbol{u}$ consisting of a sequence of $n$ previous words, let $G_{\boldsymbol{u}}(w)$ be a distribution over word $w$ having Pitman-Yor process as a prior:

$$G_{\boldsymbol{u}} \sim PY(d_{\boldsymbol{u}}; \theta_{\boldsymbol{u}}; G_{\pi(\boldsymbol{u})}) \quad (1)$$

where $d_{\boldsymbol{u}} \epsilon [0,1)$ is a discount parameter, $\theta_{\boldsymbol{u}}$ is a strength parameter, $\pi(\boldsymbol{u})$ is a context of $\boldsymbol{u}$ consisting of $(n-1)$ previous words. Since base distribution $G_{\pi(\boldsymbol{u})}$ is unknown either, its prior is recursively placed over it in (1) with parameters $(d_{\pi(\boldsymbol{u})}; \theta_{\pi(\boldsymbol{u})}; G_{\pi(\pi(\boldsymbol{u}))})$. This recursion is repeated until we get $G_{\emptyset}$, that is a distribution of the current word given an empty context $\emptyset$. The prior for this distribution is given following form

$$G_{\emptyset} \sim PY(d_{\emptyset}; \theta_{\emptyset}; G_0) \quad (2)$$

where the base distribution $G_0$ is assumed to be uniform over the vocabulary.

Sequence memoizer is essentially an implementation of such unbounded-depth HPYP LM, where $n \to \infty$ [13]. In this case, strength parameter $\theta_{\boldsymbol{u}}$ is equal to 0. Then, predictive distribution of a word given its previous context $\boldsymbol{u}$ takes form

$$G_{\boldsymbol{u}}(w) = \frac{c_{\boldsymbol{u}}(w) - d_{\boldsymbol{u}} t_{\boldsymbol{u}}(w) + d_{\boldsymbol{u}} t_{\boldsymbol{u}} G_{\pi(\boldsymbol{u})}(w)}{c_{\boldsymbol{u}}} \quad (3)$$

where $c_{\boldsymbol{u}}(w)$ is a count of draws with the context being $\boldsymbol{u}$ of word $w$; $c_{\boldsymbol{u}}$ is a count of context being $\boldsymbol{u}$; $t_{\boldsymbol{u}}(w)$ is a count of draws with the context of word $w$ being $\boldsymbol{u}$ and recursion, using $\pi(\boldsymbol{u})$ shorter suffix of the context, $G_{\pi(\boldsymbol{u})}$ was applied; $t_{\boldsymbol{u}}$ is a count of draws with the context being $\boldsymbol{u}$ and recursion $G_{\pi(\boldsymbol{u})}$ was applied. If context $\boldsymbol{u}$ doesn't appear in the context tree, then the longest suffix of $\boldsymbol{u}$ is used, $\pi(\boldsymbol{u})$ or $\pi(\pi(\boldsymbol{u}))$ and so on.

When building model over very long sequences, large number of recursion of form (1) might be required, which rises the computational cost a lot. To reduce the size of the model all non-branching, non-leaf nodes are integrated out leaving a finite number of nodes in a compact context tree.

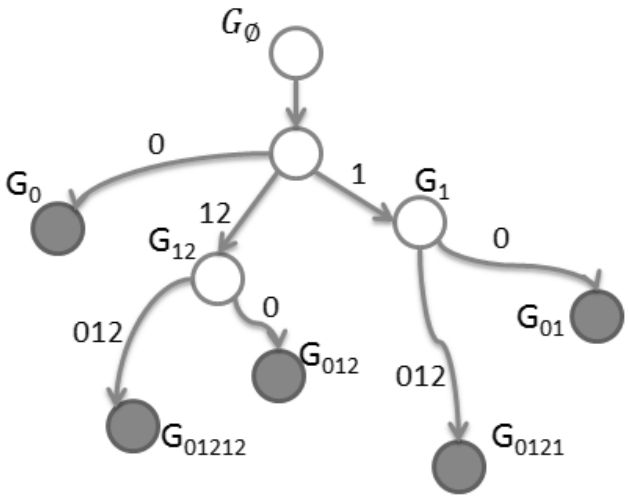


**Figure 1 Sequence memoizer compact context tree**

Figure 1 shows the graphical model instantiated by the sequence of integers *01212*. Note that in this SM compact context tree, nodes that are not branching nodes and are not associated with observed data are already integrated out. For instance, in our example $G_\emptyset \to G_{12}$ path in non-compact tree will take form $G_\emptyset \to G_1 \to G_{12}$. In this case, parameters in form (1) are changed to $(d_\emptyset; d_1;\ 0; G_1)$.

Inference in the SM model is performed by recursive application of the Chinese restaurant process in the same way as for the HPYP LM. In [14], a detailed inference scheme of the model discount parameter $d_{\boldsymbol{u}}$ and word arrangement variables $c_{\boldsymbol{u}}(w)$, $c_{\boldsymbol{u}}$, $t_{\boldsymbol{u}}(w)$, $t_{\boldsymbol{u}}$. is described.

To calculate perplexity for this model, predictive distribution of the form (3) is used as probability of a word given context $P(w|\boldsymbol{u})$.

## 3. EXPERIMENTS

### 3.1. Databases and feature extraction

Our text corpus contains 41M words with vocabulary size of about 100K words. This corpus was assembled from recent news articles published by freely available Internet sites of several on-line Russian newspapers for the years 2006-2011. We split our corpus into 40M words train set and a test set consisting of 1M words. For experiments with different corpus sizes, we separated 10M, 20M and 30M words from the full train set and used them as smaller train sets.

**Table 1. Perplexities obtained using test sets with various average sentence length and train set of 10M words**

| Average length | Model name | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | SM-1 | SM-2 | SM-3 | SM-4 | SM-5 | SM-6 | SM-fs | SM-bs | RNN-fs | 3-gram |
| 9 | 495 | 3273 | 1307 | 5073 | 710 | 857 | 779 | 572 | 1237 | 733 |
| 13 | 420 | 2208 | 880 | 3306 | 540 | 642 | 640 | 484 | 938 | 571 |
| 17 | 409 | 1944 | 793 | 2883 | 492 | 583 | 617 | 472 | 794 | 518 |
| 21 | 398 | 1755 | 730 | 2584 | 458 | 540 | 595 | 458 | 717 | 483 |
| | | | | | | | | | | |
| all | 414 | 2134 | 855 | 3191 | 526 | 625 | 644 | 479 | 902 | 550 |

In our ASR experiments, we used the SPIIRAS [16] and GlobalPhone [3] Russian speech databases. Speech data are collected in clean acoustic conditions. In total, there are 28671 utterances pronounced by 165 speakers (86 male and 79 female) with duration of about 38 hours. Speech test data consist of 10% of the GlobalPhone recordings pronounced by 5 male and 5 female speakers not used for acoustic model (AM) training.

The speech signal was coded with energy and 12 MFCCs and their first and second order derivatives. The AM consists of 5342 tied states with 16 mixture GMMs as output models. Our speech decoder (Julius ver. 4.2 [17]) produces 500-best hypothesis list, which we use for re-scoring by the SM and RNN LMs.

The SM LMs were built using the java version of the Sequence memoizer toolkit [18] and the RNN LMs were implemented using the RNNLM toolkit (v.0.3b) [15].

### 3.2. Experimental results

When modeling long span word dependencies across sentence boundaries, sequence modeling would strongly depend on the sentence order in the training data. In many cases text corpus consists of unconnected by meaning sentences, because after data pre-processing some sentences are eliminated. Thus, we can assume that our initial data are shuffled. To find out how performance of the model depends on train data order, we built models using shuffled and sorted data. Here, we used random shuffling and sorting by sentence length in increasing and decreasing order.

Our sequence memoizer model is built using word as atomic unit, unlike previous attempts built using symbols. In this case, vocabulary size of the model increases significantly from 128 to 100K. Because of the high computational complexity we weren't able to make as many sampling iterations as probably be necessary for more efficient parameter estimation. Changing sampling number up to 1000 didn't show any influence on the model performance, while the computation time increased a lot. Thus, we used one sampling iteration for building our SM models.

For RNN LM evaluation we used optimal parameters identified in [10], 150 hidden nodes and 1000 classes. We used train data sorted in increasing order of sentence length in all experiments with RNN, since the performance didn't vary significantly depending on train data order.

*3.2.1. Test sequence length experiment*

In this experiment, we used small train set of 10M words. From the rest of text data, we selected test sets of 1M words each so that average sentence length in these sets is different. In total, all test sets contain 2.8M words.

Our baseline 3-gram was trained using 10M word train set as it is. RNN LM was trained using same set sorted in increasing order of sentence length (RNN-fs). In order to investigate effect of sentence order on the SM LM performance, we randomly shuffled the training set 6 times (SM-1 – SM-6), as well as sorted it in increasing order (SM-fs) and decreasing order (SM-bs). Perplexity, obtained using all test sets is summarized in Table 1. Performance of SM varies in very wide range depending on train data order. Shuffled models SM-1, SM-5 and sorted in decreasing order model SM-bs outperform both 3-gram and RNN LMs over all test sets. For all models, perplexity improves as the average sentence length increases.

*3.2.2. Performance with increasing size of the train corpus*

In [19] it was reported, that with lots of training data, improvements provided by many advanced modeling techniques almost disappear.

To investigate influence of increasing amount of train data, we used 4 train sets of 10, 20, 30 and 40 millions of words and test set as described in Section 3.1. We chose both models built using sorted data and two SM LMs built using shuffled data, which showed better performance in the previous experiment; SM-1 and SM-5. In the same manner, we built SM LMs using 20M and 30M train sets. Full size models were trained using sorted 40M train data. RNN LMs

**Table 2. Perplexities of models built using various sizes of train sets**

| Train set size | SM-1 | SM-5 | SM-fs | SM-bs | RNN-fs | 3-gram | Relative improvement, % |
|---|---|---|---|---|---|---|---|
| 10M | **407** | 480 | 611 | 468 | 780 | 504 | 19 |
| 20M | 236 | 267 | 328 | **170** | 422 | 327 | 48 |
| 30M | 155 | 160 | 243 | **115** | 323 | 282 | 59 |
| 40M | - | - | 205 | **117** | 320 | 257 | 54 |

**Table 3 WERs of standalone and interpolated models built using 40M train set**

| Model | SM-fs | SM-bs | RNN-fs | 3-gram |
|---|---|---|---|---|
| SM/RNN/3-gram | 38.8 | 38.7 | 33.9 | 34.5 |
| SM + RNN | 33.7 | 33.7 | - | - |
| SM/RNN + 3-gram | 34.5 | 34.4 | 32.7 | - |
| SM + RNN + 3-gram | 32.7 | **32.6** | - | - |

were built using each train set separately with same parameters identified in Section 3.2.

Perplexities obtained using various model sizes are summarized in Table 2. In the last column, relative improvement in perplexity obtained by SM LM with the lowest perplexity over baseline 3-gram is presented. We can observe that relative improvement doesn't vanish with increasing size of train data; it keeps on the same level for 20M, 30M and 40M words train sets. The lowest perplexities were obtained by SM-bs model, built with data sorted in decreasing order of sentence length, using train sets of 20M, 30M and 40M words.

*3.2.3. Speech recognition evaluation of interpolated models*

Next, we evaluated speech recognition performance using models trained with 40M data set, based on perplexity evaluation results in the previous experiment. In Table 3, speech recognition performance is presented for SM, RNN and 3-gram LMs as well as for their linear interpolations. Although SM outperform both 3-gram and RNN in terms of perplexity, its standalone speech recognition performance is worse than RNN and 3-gram ones. Nevertheless, WER relative improvement of 5.3% was achieved using linear interpolation of all 3 models.

**Table 4. Examples of data generated by models trained on 40M train set.**

| Model | Generated data |
|---|---|
| SM | Потушить огонь пытаются тридцать пожарных расчетов. (Team of thirty firemen is trying to extinguish fire.)<br>В этом году на проведение операции выделено сто восемьдесят тысяч рублей. (One hundred eighty thousand rubles were assigned to conduct the campaign this year.) |
| RNN | Не повторять что в Женеве не смогли держать прошлое да и намерен наконец свою работу сочинений мотоциклы которые он может случится в настоящее время… (Do not repeat that in Geneva could not keep the past and intends to finally his work essay motorcycles which he can happen now…) |
| 3-gram | Поддержка региона России угрозы новой он уровень сложности идти вместе. (Support region Russian threat of a new level of complexity it goes together.)<br>Впрочем другие уволил совместные своего предшественника у нас экономические санкции. (However others fired joint his predecessor we have economic sanctions.) |

*3.2.4. Random sentence generation from the SM, RNN and 3-gram LMs*

For testing model ability to generate valid sentences, we used SM-bs, RNN-fs and 3-gram models trained using 40M train set. Table 4 demonstrates example data generated from each LM with their approximate translation, because it isn't possible to make unambiguous translation of grammatically incorrect, meaningless sentences. It is easy to see that examples generated by SM LM are grammatically correct with appropriate choice of words. Note that RNN LM generated very long sentences, failing in splitting word sequences into sentences of appropriate length.

*3.2.5. Training time comparison for SM and RNN LMs.*

Finally, we compared SM and RNN LMs in terms of training time using different size of text data sets. Here, we used train data sorted in descending order to train SM. Figure 2 shows that SM training time increases almost linearly, which is optimistic result for further experiments with more data.

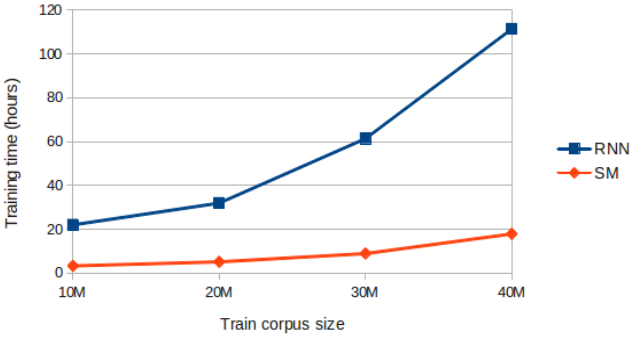


**Figure 2. Training time of SM and RNN LMs built using various train sets**

## 4. CONCLUSION

As far as we know, this is the first attempt to apply sequence memoizer language model for speech recognition task. Similar to [11], we observed reduction in perplexity using sequence memoizer language model. Nevertheless, it didn't result in reduction of WER applying standalone SM LM. Experiments with interpolation with other models show negligible improvement, when SM scores are also included. Also, our experiment with data generation shows that SM is able to capture dependencies within sentence and produce grammatically correct and meaningful sentences. More work needs to be done to determine whether the SM model can be successfully applied to the ASR task.

## 5. REFERENCES

[1] P. Cubberley, "Russian: a linguistic introduction," Cambridge University Press, 2002.

[2] E.W. Whittaker, P.C. Woodland, "Comparison of language modelling techniques for Russian and English," *In: Proc. ICSLP*, 1998.

[3] S. Stuker, T. Schultz, "A grapheme based speech recognition system for Russian," *In: Proc. SPECOM*, St.Peterburg, Russia, pp. 297–303, Sep. 2004.

[4] D. Vazhenina, K. Markov, "Phoneme set selection for Russian speech recognition," *In: Proc. IEEE NLP-KE*, Tokushima, Japan, pp. 475–478, Nov. 2011.

[5] L. Lamel, S. Courcinous, J.L. Gauvain, Y. Josse and V.B. Le, "Transcription of Russian conversational speech," *In: Proc. SLTU*, Cape Town, South Africa, pp. 156-161, May 2012.

[6] Y. Titov, K. Kilgour, S. Stüker and A. Waibel, "The 2011 KIT QUAERO Speech-to-Text System for Russian," *In: Proc. SPECOM*, Kasan, Russia, Sep. 2011.

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

[8] T. Mikolov, M. Karafiat, L. Burget, J. Cernocky, S. Khudanpur, "Recurrent neural network based language model," *In: Proc. INTERSPEECH*, Makuhari, Japan, pp. 1045–1048, Sep. 2010.

[9] T. Mikolov, A. Deoras, S Kombrink, L. Burget and J. Cernocký, "Empirical Evaluation and Combination of Advanced Language Modeling Techniques," *In: Proc. INTERSPEECH,* Florence, Italy, pp. 605-608, Aug. 2011.

[10] D. Vazhenina, K. Markov, "Evaluation of advanced language modelling techniques for Russian LVCSR", *In: Proc. SPECOM*, Pilzen, Czech Republic, pp. 124-130, Sep. 2013.

[11] F. Wood, J. Gasthaus, C. Archambeau, L. James, and Y.W. Teh, "The Sequence Memoizer," Communications of the ACM, vol. 54, no. 2, pp. 91-98, 2011.

[12] Y. W. Teh, "A hierarchical Bayesian language model based on Pitman-Yor processes," *In: Proc. Annual Meeting of the ACL*, Sydney, Australia, pp. 985 – 992, Jul. 2006.

[13] F. Wood, C. Archambeau, J. Gasthaus, L.F. James, and Y. Teh, "A Stochastic Memoizer for Sequence Data," *In: Proc. ICML*, pp. 1129–1136, 2009.

[14] Y.W. Teh. "A Bayesian Interpretation of Interpolated Kneser-Ney," Technical Report TRA2/06, School of Computing, NUS, 2006.

[15] T. Mikolov, S. Kombrink, L. Burget, J. Cernocky, S. Khudanpur, "Extensions of recurrent neural network language models," *In: Proc. ICASSP*, Prague, Czech Republic, pp. 5528–5531, May 2011.

[16] 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*, St.Petersburg, Russia, pp. 515–520, June 2009.

[17] A. Lee, T. Kawahara, "Recent development of open-source speech recognition engine Julius," *In: Proc. APSIPA ASC*, Sapporo, Japan, pp. 131–137, Oct. 2009.

[18] Sequence memoizer, http://www.robots.ox.ac.uk/~fwood/code/sequencememoizer/.

[19] J. T. Goodman, "A bit of progress in language modeling, extended version," Technical report MSR-TR-2001-72, 2001.