Evaluation of Advanced Language Modeling Techniques for Russian LVCSR

Daria Vazhenina and Konstantin Markov

Human Interface Laboratory, The University of Aizu, Japan
{d8132102,markov}@u-aizu.ac.jp

Abstract. The Russian language is characterized by very flexible word order, which limits the ability of the standard *n*-grams to capture important regularities in the data. Moreover, it is highly inflectional language with rich morphology, which leads to high out-of-vocabulary (OOV) word rates. In this paper, we present comparison of two advanced language modeling techniques: factored language model (FLM) and recurrent neural network (RNN) language model, applied for Russian large vocabulary speech recognition. Evaluation experiments showed that the FLM, built using training corpus of 10M words was better and reduced the perplexity and word error rate (WER) by 20% and 4.0% respectively. Further WER reduction by 7.4% was achieved when the training data were increased to 40M words and 3-gram, FLM and RNN language models were combined together by linear interpolation.

Keywords: language modeling, Russian language, factored language models, recurrent neural network, inflectional 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).

Regardless, in current Russian large vocabulary continuous speech recognition (LVCSR) systems conventional *n*-grams are usually used [2,3,4]. An improved bi-gram was proposed in [5] where the counts of some of the existing *n*-grams are increased after syntactic analysis of the training data. Long-distance dependencies between words are

M. Železný et al. (Eds.): SPECOM 2013, LNAI 8113, pp. 124–131, 2013.

identified and added as new bi-gram counts. This allowed to reduce the word error rate of a speech recognition system with dictionary of 208K words from 58.4% to 56.1%.

Recently, for the Arabic, which is also highly inflectional language, it was proposed to incorporate word features, called factors, into the language model [6]. This factored language model (FLM) implements a back-off procedure by excluding factors one by one or even several factors at a time without taking into account factor's distance from the predicted word. This improves the robustness of the probability estimates for rarely observed word *n*-grams. Using this model, relative WER reduction of 3.4% was achieved for Arabic LVCSR system with 70K vocabulary size [7]. For the Turkish language, it was reported that the FLM reduced the WER by 1.7% relative for a 200K words ASR system [8].

For the Czech language, which is also morphologically rich, implementation of neural network (NN) based language models was presented in [9]. Using such 4-gram LM for the Czech lecture recognition task, WER relative improvement of 15% was obtained. To be able to use larger context, LM based on recurrent neural network (RNN) was proposed in [10]. RNN LMs allow effective processing of arbitrary length sequences, which overcomes the main *n*-gram drawback - dependency on only few consecutive words.

This paper describes our implementation of the FLM and RNN LM for Russian LVCSR with vocabulary of 100K words. We investigated the influence of different factors for the FLM and different size of the RNN hidden and output layers on the language model performance. Both language modeling techniques are implemented using n-best re-scoring. Best, in terms of WER, was the interpolation of the conventional 3-gram LM with both the FLM and RNN LM.

2 Factored Language Models

In the factored language model (FLM), it is proposed to include word features, called factors, in the standard *n*-gram language model. Factors of a given word can be any grammatical information about the word, such as its lemma, stem, root, ending, part-of-speech, etc. In the FLM, word sequence $W = \{w_1, w_2 ... w_t\}$ is represented by a sequence of K factors for each word w_i, $f_i^{1:K} = \{f_i^1, f_i^2 ... f_i^K\}$. A probabilistic language model is estimated over the factor vectors. Using *n*-gram-like formula, the general model takes the following form $P(f_t^{1:K} | f_{t-1}^{1:K}, f_{t-2}^{1:K} ... f_{t-n+1}^{1:K})$, which can be simplified to $P(f_t | f_1, f_2 ... f_m)$, where $f_t = w_t$ and $\{f_i\}, i = 1...m, m \leq K * (n-1)$ is any combination of factors. If an *n*-gram of word or factor is not sufficiently observed, generalized back-off procedure is used. As shown on Fig.1(b), during the back-off any factor can be dropped at each step in any order. This flexible back-off procedure is the main advantage of the FLM.

In order to obtain a good FLM performance, we need to tune its parameters: the combination of conditioning factors (factor set) and the back-off tree. In [6], two ways were proposed to optimize these parameters: manually choose the factor set and fix the back-off tree based on linguistic knowledge [8]; automatically determine optimal parameters using genetic algorithm (GA) [6,7].

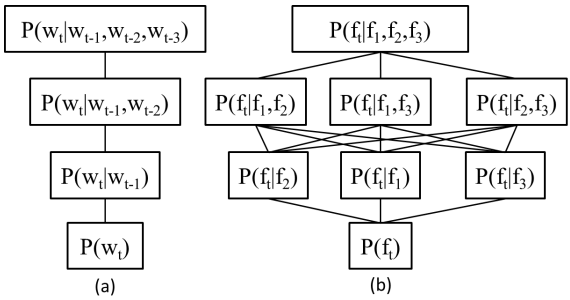

Fig. 1. *N*-gram and FLM back-off trees. (a) Standard *n*-gram back-off path has strict order of dropping words, (b) In the FLM case, any factor may be dropped at each step resulting in many possible paths.

In [6], it was demonstrated that the factor set and back-off tree optimized using GA can perform better than hand-selected ones. In addition, relative WER reductions presented in [6] and [7] were higher than those reported in [8].

The genetic algorithm for FLM optimization seeks the optimal factor set and back-off tree based on minimizing the model perplexity over some test set. This procedure produces many FLMs and those with the lowest perplexity are further evaluated on speech test data.

3 Recurrent Neural Network

The main advantage of the RNN LM over conventional *n*-gram and feed-forward NN LM is the ability to store arbitrary long history of given word [10].

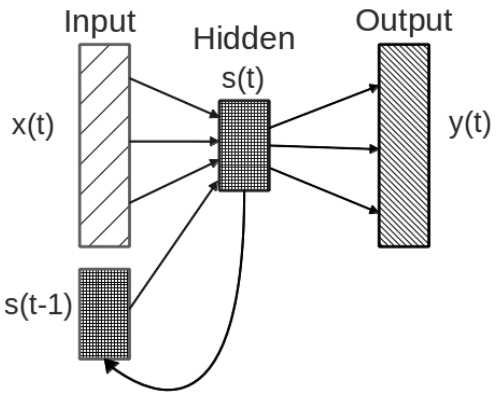

Fig. 2. Recurrent neural network architecture for language modeling

The RNN used in our experiments is shown in Fig.2 and has three layers. An 1-of-N vector representation x(t) of the current word is fed to the input layer, which also takes

the output s(t-1) of the hidden layer. This recurrent feedback stores the word history information in the hidden layer. The output layer gives the probability distribution of the next word y(t) and therefore uses neurons with softmax activation function. The size of the input and output layers is defined by vocabulary size, but the size of the hidden layer is a free parameter, which has to be determined experimentally.

Learning of the weight matrix between the hidden and the output layer is the most time consuming part in the NN training. To reduce the computational complexity, it was proposed to factorize the output layer using classes [11]. Words are mapped into classes with frequency binning, which proportionally assigns words to classes based on their frequency. Thus, the number of classes becomes another free parameter of the model.

4 Language Resources

Our text corpus contains 41M words with vocabulary size of about $\sim$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 the first experiments with FLM and RNN LM 10M words were separated from the full train set and used as small train set.

Word features for the FLM were obtained using the TreeTagger tool [12] with tagset described in [13]. This tool annotates words with lemma and morphological tag, which contains detailed grammatical information about the word, such as POS category, gender, number, case, etc.

In addition to the word, lemma and morphological tag factor types, we use two extra: POS category and gender-number-person factor, which contains important grammatical information for the word relations in a sentence. The list of all factor types, we experimented with, is shown in Table 1.

Table 1. Factor types for the Russian language used in the experiments

Factor type	Description	Size
W	word	99 958
L	lemma	23 742
T	morphological tag	819
G	gender, number and person	30
P	part-of-speech (POS) category	10

The LM training corpus is preprocessed so that every word is replaced with a vector of all factor types. For instance, word ’брэнды’ (brands) is replaced with the vector {W-брэнды:P-N:T-Ncmpnn:L-брэнд:G-MP}.

5 Experiments

5.1 Speech Database and Feature Extraction

In our experiments, we used the SPIIRAS [14] 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 [15]) produces 500-best hypothesis list, which we use for re-scoring by the selected FLMs.

The FLMs we built using SRILM toolkit (v.1.5.11) [16]and the RNN LMs were implemented using the RNNLM toolkit (v.0.3b) [11].

5.2 Experimental Results

To determine parameters of the FLM and RNN LM, we used the small train set of 10M words, which speeds up this step. As a baseline LM, we use conventional 3-gram trained on same train data using Kneser-Ney discount. Its perplexity is 537 and the word error rate is 35.4%.

FLM Evaluation

First, we evaluated each factor type from Table 1 individually. For this reason, we built several small FLMs using the word and one of the other factor types for time context 1 and 2 corresponding to 2-gram and 3-gram contexts respectively. In other words, train FLMs, which model probability distributions $p_k(w_t|w1, f_k1, w2, f_k2)$ for $k = 1...K$. On this step, we set back-off path manually, in a manner similar to the conventional 3-gram back-off path, which has two possible variations:

- Back-off path 1: Drop the words in time distance order: $w2, w1$, then drop factors in the same order: f_k2, f_k1.
- Back-off path 2: First drop the most distant word and factor $w2, f_k2$, then less distant ones $w1, f_k1$.

The performance of these FLMs presented in Table 2 shows that the back-off path has big influence on the perplexity. Since factors G and P showed the perplexity worse than baseline in both cases, we choose to continue with L and T factor types only.

Then, using GA, we find the optimal factor set and back-off trees for factor types W, L and T using time context 1, 2 and 3. In Table 3, results obtained using FLMs with lowest perplexities are presented. The best perplexity was achieved using the largest model F1, which is a significant 19.9% reduction relative to the baseline. Using model F3 with longer context, the lowest WER was achieved. However, the biggest relative WER reduction of 6.9% is obtained by the interpolated with 3-gram model F2, which is built with quite small factor set and not highly branched back-off tree. So, we chose this model for further experiments.

Table 2. Perplexities of FLMs built from the small training set with different back-off paths. The baseline perplexity is 537.

Factor types	Back-off path 1	Back-off path 2
WL	611	525
WT	685	488
WG	988	714
WP	652	549

Table 3. Performance of FLMs built using most effective factors

Model	Factors (# back-off tree nodes)	FLM		3-gram + FLM		
		perpl.	WER,%	Int.coef	perpl.	WER,%
3-gram				1.0	537	35.4
F1	W1,L1,T1,W2,L2,T2 (55)	430	34.5	0.51	437	33.3
F2	W1,L1,T1,L2,T2 (20)	440	34.4	0.45	445	**33.0**
F3	W1,L1,T1,L2,T2,T3 (43)	460	**34.0**	0.43	463	33.4

RNN LM Evaluation

To determine main parameters of RNN LMs: the number of hidden nodes and the number of word classes, we trained several models using the small train set. Perplexity and WER of those models were much higher than baseline, about 1100 and 38% respectively. However, when interpolated with the baseline 3-gram, their performance improved significantly. In Table 4, we summarize results obtained from the 3-gram and RNN LM interpolation. The interpolation coefficient (Int.coef.) was manually tuned for each model.

Table 4. Performance of the RNN LM interpolated with 3-gram. The baseline perplexity is 537 and the WER is 35.4%.

# hidden nodes	# classes								
	150			500			1000		
	Int.coef	perpl.	WER,%	Int.coef	perpl.	WER,%	Int.coef	perpl.	WER,%
100	0.68	466	33.5	0.64	470	33.3	0.68	503	33.3
150	0.67	474	33.6	0.67	457	33.3	0.66	462	**33.1**
200	0.69	454	33.3	0.77	458	33.5	0.64	471	34.1
250	0.81	459	33.9	0.63	469	33.5	0.67	459	33.5

Perplexity of all interpolated models is lower than the baseline and is in the same range as of the FLM. The best relative WER improvement of 6.5% was obtained using 150 hidden nodes and 1000 classes. These parameters were chosen for the further experiments.

Results Using the Full Training Set

After defining optimal parameters for both LM types, the 3-gram and FLM (F2) models were re-trained using all training text. On the other hand, the RNN model with 150 hidden nodes and 1000 classes (h-150, c-1000) was updated with one more iteration using the full train data. Using the full training set, the baseline 3-gram LM perplexity became 293 and the word error rate - 33.9%. In the Table 5, results obtained using the new models and their interpolations are presented. While FLM model WER improved, the result of its interpolation with the 3-gram did not change. However, in terms of WER, performance of the stand-alone RNN model improved significantly from 38.3% to 32.9%. The best WER relative improvement of 7.4% was achieved by the linear interpolation of all the 3 models.

Table 5. Performance of FLM and RNN LMs built using all train data

#	Model	Int.coef.	perpl.	WER, %
1	3-gram	1.0	293	34.0
2	FLM (F2)	1.0	242	33.7
3	(1) + (2)	0.55 + 0.45	241	33.0
4	RNN (h-150, c-1000)	1.0	393	32.9
5	(1) + (4)	0.49 + 0.51	244	31.9
6	(2) + (4)	0.6 + 0.4	215	32.0
7	(1) + (2) + (4)	0.3 + 0.4 + 0.3	216	**31.5**

6 Conclusions

This paper presents implementation of some advanced language modeling techniques such as factored language model and recurrent neural network language model for Russian speech recognition task. We evaluate those models and present the results obtained using small and large training sets. Obtained WER relative improvement of 7.4% is quite high, in comparison to improvements achieved for other morphologically rich languages such as Arabic and Turkish using advanced language modeling techniques [7,8].

Factored language models seems to be able to capture additional information and improve LM probability estimates, when the amount of training data is limited. On the other hand, RNN LM evaluation showed significant improvement applying large training set.

References

1. Cubberley, P.: Russian: a linguistic introduction. Cambridge University Press (2002)
2. Whittaker, E.W., Woodland, P.C.: Comparison of language modelling techniques for Russian and English. In: Proc. ICSLP (1998)
3. Stuker, S., Schultz, T.: A grapheme based speech recognition system for Russian. In: Proc. SPECOM, St. Peterburg, Russia, pp. 297–303 (September 2004)

4. Vazhenina, D., Markov, K.: Phoneme set selection for Russian speech recognition. In: Proc. IEEE NLP-KE, Tokushima, Japan, pp. 475–478 (November 2011)
5. Karpov, A., Kipyatkova, I., Ronzhin, A.: Very large vocabulary ASR for spoken Russian with syntactic and morphemic analysis. In: Proc. InterSpeech, pp. 3161–3164 (August 2011)
6. Kirchhoff, K., Vergyri, D., Bilmes, J., Duh, K., Stolcke, A.: Morphology-based language modelling for conversational Arabic speech recognition. Computer Speech and Language 20(4), 589–608 (2006)
7. El-Desoky Mousa, A., Schluter, R., Ney, H.: Investigations on the use of morpheme level features in language models for Arabic LVCSR. In: Proc. ICASSP, Kyoto, Japan, pp. 5021–5024 (March 2012)
8. Sak, H., Saraclar, M., Gungor, T.: Morphology-based and sub-word language modelling for Turkish speech recognition. In: Proc. ICASSP, Dallas, USA, pp. 5402–5405 (March 2010)
9. Mikolov, T., Kopecky, J., Burget, L., Glembek, O., Cernocky, J.: Neural network based language models for highly inflective languages. In: Proc. ICASSP, Taipei, Taiwan, pp. 4725–4728 (April 2009)
10. Mikolov, T., Karafiat, M., Burget, L., Cernocky, J., Khudanpur, S.: Recurrent neural network based language model. In: Proc. InterSpeech, Makuhari, Japan, pp. 1045–1048 (September 2010)
11. Mikolov, T., Kombrink, S., Burget, L., Cernocky, J., Khudanpur, S.: Extentions of recurrent neural network language models. In: Proc. ICASSP, Prague, Czech Republic, pp. 5528–5531 (May 2011)
12. Schmid, H.: Probabilistic part-of-speech tagging using decision trees. In: Proc. NeMLaP, Manchester, UK, pp. 44–49 (1994)
13. Sharoff, S., Kopotev, M., Erjavec, T., Feldman, A., Divjak, D.: Designing and evaluating Russian tagsets. In: Proc. LREC, Marrakech, pp. 279–285 (May 2008)
14. 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: Proc. SPECOM, St. Petersburg, Russia, pp. 515–520 (June 2009)
15. Lee, A., Kawahara, T.: Recent development of open-source speech recognition engine Julius. In: Proc. APSIPA ASC, Sapporo, Japan, pp. 131–137 (October 2009)
16. Stolcke, A.: SRILM - an extensible language modeling toolkit. In: Proc. ICSLP, vol. 2, pp. 901–904 (2002)