Language model reduction for practical implementation in LVCSR systems

Stevan Ostrogonac, Branislav Popović, Milan Sečujski
Faculty of Technical Sciences
University of Novi Sad
Novi Sad, Serbia
ostrogonac.stevan@uns.ac.rs, bpopovic@uns.ac.rs,
secujski@uns.ac.rs

Robert Mak, Darko Pekar
AlfaNum – Speech Technologies
Novi Sad, Serbia
robert.mak@alfanum.co.rs
darko.pekar@alfanum.co.rs

Abstract — **This paper presents a way to reduce the size of an *n*-gram based language model (LM). This reduction method has proven to be very efficient in the sense of minimal information loss. The list of *n*-gram probabilities becomes too large for use in large-vocabulary speech recognition (LVCSR) systems when medium or large vocabularies are used for language modeling. Two ways of reducing the *n*-gram file were compared in this research. The first way was implemented by setting a threshold on the counts of *n*-grams to be kept. The second method considered discarding the *n*-gram probabilities if their removal causes the language model's perplexity to increase by less than a predefined threshold. The experiment was done on word-based and lemma-based LMs for the Serbian language and in both cases it was shown that the latter method is more successful.**

Key words – language model; perplexity; n-gram;

I. INTRODUCTION

Large vocabulary speech recognition (LVCSR) systems generally consist of three modules which introduce distinctive information to the speech recognition process. These modules are the acoustic, lexical and language model [1]. This study is focused on the language model (LM), which provides linguistic information to the system. The common form of language models used in LVCSR systems are *n*-gram models [2]. An *n*-gram is a sequence of words of length *n*. An *n*-gram model represents a list of *n*-gram probabilities obtained by model training, using a preferably large textual corpus.

Languages which are morphologically very complex, like Serbian, require very large textual corpora for the training phase of the language modeling process. The corpus used in this research consists of over 16 million word instances. Different types of textual content written in Serbian were used in order to include different literary styles in the model [3]. However, it has been determined that the existing corpus of 16 million words is not sufficient to reach the approximate potential of the *n*-gram model [4]. This was the reason for developing lemma and class-based *n*-gram models, which are based on smaller vocabularies. These models will be described in more detail in the next section. Both the word-based and the lemma-based models turned out to be too large to be adequate for use in a LVCSR system because the search of the *n*-gram probability file was very time-consuming.

In order to reduce the *n*-gram model and make it more suitable for practical use, two methods of discarding information were compared in this study. Both methods are implemented as a part of the Stanford Research Institute Language Modeling (SRILM) toolkit [5], which was also used for training and evaluation of the models in this research. The first method for reducing the LM considered setting a threshold for *n*-gram counts (except for unigram counts) to be higher than the default value of 1. Since the counts are expressed as integers, incrementing the threshold enables somewhat abrupt reduction of the LM, especially for highly inflective languages, or if the training corpora are not large enough. The second LM reduction method is based on setting a perplexity threshold. This method is known as entropy-based pruning and it was proposed in [6]. Perplexity is a LM quality measurement calculated by using the expression:

$$ppl = \sqrt[K]{\prod_{i=1}^{K} \frac{1}{P(w_i \mid w_1 \ldots w_{i-1})}} \qquad (1)$$

where *ppl* is the perplexity value, *K* is the number of words contained in the training corpus, and $P(w_i|w_1...w_{i-1})$ is the conditional probability for a word w_i appearing in the context $w_1...w_{i-1}$. When two language models are compared, the one with lower perplexity is considered to be a better language representation. The LM reduction by using perplexity is done by discarding all *n*-gram probabilities which, when removed, cause the LM's perplexity to increase by less than the threshold value. This method offers more sophisticated control over the size of the resulting language model.

It is important to distinguish between the model's perplexity and the perplexity obtained by applying the model on some test data set. When reducing the LM using the second described method, the perplexity increase refers to the perplexity calculated on the training data set. The evaluation of the resulting model is done by calculating the perplexity on some textual corpus which is disjunctive with the training corpus, which was done in this study in order to compare the two language model reduction methods.

II. LANGUAGE MODELS FOR SERBIAN

When creating a language model for the Serbian language, the most important problem that had to be overcome was the existence of a large number of out-of-vocabulary (OOV) words in the test data set. Since there are a very large number of inflected forms for most canonical word forms, many of the possible inflections are not contained in the training corpus. This causes the language model to give poor probability estimates for a given textual content in some cases.

In order to address this problem, the vocabulary with which the LM operates had to be reduced significantly. For special purposes, the vocabulary can be specified in advance, and the training corpus can be used to model only the probabilities of *n*-grams consisting of the words contained in the vocabulary. This is a good solution for applications in which the vocabulary is restricted and predictable, but this is usually not the case.

Another way to deal with the OOV words problem is to define a number of word classes which is significantly smaller than the number of actual words observed in the training data set. Then the words in the training set can be replaced by their corresponding classes. In the test phase, the LM will return an estimate for a word probability according to the probability of the class to which the word belongs to. If an OOV word occurs, but its corresponding class can be estimated in some way, the model will not return a default probability value but rather the class probability. The class probability is usually a better estimate than zero (or some other value) probability for OOV words. Furthermore, if a word appeared in the training corpus more times than it usually appears in the textual documents, the corresponding class represents a more adequate source for determining the word's probability. On the other hand, not all words in the training corpus are represented well by their class probabilities. This is why class *n*-gram modeling is usually done when there is not sufficient textual data available for the training.

For the Serbian language, two types of word classification have been developed so far. The first type of classification was done according to the canonical word forms (also referred to as lemmas) which correspond to the word instances appearing in the training corpus. Replacing the words with adequate lemmas reduces the vocabulary of the LM by approximately 50% [3]. The probability estimates for lemmas represent mostly semantic information contained in the textual corpus. The language model obtained by estimating the probabilities of *n*-grams consisting of lemmas will be referred to as the lemma-based LM in the rest of this paper. The second classification method takes into account a group of morphological categories which defines each particular inflected word form. The relevant group of morphological features is different for each word type. For example, this group for nouns can contain information on case, gender, number etc. For verbs, it can include categories such as tense, grammatical mood and aspect. This type of classification distinguishes mostly syntactic information. Currently, 1124 word classes are defined using morphological properties for this class-based model for the Serbian language. The introduction of these classes leads to a great reduction of the vocabulary since the number of different inflected forms in the training corpus for Serbian is approximately 350,000.

The most important task to be solved when using word classes is the classification of the OOV words. For the purpose of this research, morphological dictionary for Serbian [7] and part-of-speech (POS) tagging software [8] were used in order to determine the lemmas and the classes corresponding to the particular words in the training corpus. The morphologic dictionary contains more than 4 million inflected word forms. Compared to the number of different inflected word forms contained in the training corpus, the number of dictionary entries is significantly larger. This enabled the classification of OOV words. The POS tagging software currently achieves accuracy of 93.7%.

The word-based, the lemma-based, and the class-based model can be combined in different ways in order to obtain the optimal results for a particular application. Unfortunately, the word-based and the lemma-based model contain vast numbers of entries, which slows the search process and therefore they are inadequate for most practical purposes. In order to create the LMs of acceptable sizes, a number of *n*-grams which contribute the least to the quality of language representation have to be eliminated from the *n*-gram file. This reduction should be done simultaneously with the iterative model evaluation in order to closely determine the minimal quality of the model needed for a particular application.

III. CREATING THE LANGUAGE MODELS AND REDUCING THE SIZE OF THE *N*-GRAM PROBABILITIES LIST

In order to determine the language model quality as a function of the size of the *n*-gram file, a group of models was created for both word and lemma corpus. All models were created using the SRILM toolkit. Good-Turing discounting has been applied on the initial counts in order to estimate the probabilities of the sequences which appeared in the training corpus [3]. The highest *n*-gram sequence length which was used for modeling was 3. Back-off coefficients were calculated for unigrams and bigrams, when needed. Back-off coefficients are a part of the Katz back-off language model [4].

Entries from the resulting *n*-gram files for this type of the word-based and the corresponding lemma-based LM, respectively, look as in the following example:

-4.53320799 *njegovih filmova* -0.4985573

-3.152893762 *on film* -0.28557144

where the numeric values on the left side represent the estimated log-probabilities for the *n*-grams (in this case bigrams) "*njegovih filmova*" and "*on film*", and the values on the right side represent the logarithm of the Katz back-off coefficients.

The Katz back-off model is used to determine the probabilities of *n*-gram sequences by using the following expressions:

$$P_{Katz}(z \mid x, y) = \begin{cases} P^*_{Katz}(z \mid x, y) & if\ C(x, y, z) > 0 \\ \alpha(x, y) P^*_{Katz}(z \mid y) & else\ if\ C(x, y) > 0 \\ P^*_{Katz}(z) & otherwise \end{cases} \tag{2}$$

$$P_{Katz}(z \mid y) = \begin{cases} P^*_{Katz}(z \mid y) & if\ C(y, z) > 0 \\ \alpha(x, y) P^*_{Katz}(z) & otherwise \end{cases} \tag{3}$$

where P_{katz} values represent the probabilities contained within the entries of the n-gram file, α values represent the Katz back-off coefficients, and C marks the counts of word sequences. The equations (2) and (3) correspond to models which contain information on word sequences consisting of up to 3 words. A generalisation of these expressions is straightforward. As it can be seen, if an n-gram has not been observed in the training corpus, its probability will be estimated according to a lower order n-gram probability and a back-off coefficient corresponding to the (n–1)-gram consisting of the first n–1 words from the original n-gram. These back-off coefficients are estimated using:

$$\alpha_{w_{i-n+1}\dots w_{i-1}} = \frac{\beta_{w_{i-n+1}\dots w_{i-1}}}{\sum_{\{w_i:C(w_{i-n+1}\dots w_{i-1})\leq k\}} P_{Katz}(w_i \mid w_{i-n+2}\dots w_{i-1})} \qquad (4)$$

where β represents the probability mass which is left over for the (n–1)-gram, and it is calculated as:

$$\beta_{w_{i-n+1}\dots w_{i-1}} = 1 - \sum_{\{w_i:C(w_{i-n+1}\dots w_i)>k\}} d_{w_{i-n+1}\dots wi} \frac{C(w_{i-n+1}\dots w_i)}{C(w_{i-n+1}\dots w_{i-1})} \qquad (5)$$

and d is the amount of discounting found by Good-Turing estimation [9]. This means that if the count C is estimated by the Goot-Turing discounting algorithm to be C^*, then:

$$d = \frac{C^*}{C} \qquad (6)$$

Expressions (2)-(6) are used to determine the probability values needed to calculate the perplexity value on the test data set. Besides perplexity values, the evaluation of all the models in this experiment was done by calculating discrimination coefficients. A discrimination coefficient represents a quantitative description of the language model's capability to distinguish between the authentic textual context and the textual content which carries no semantic information. This quality measure was defined in the previous research on LMs for Serbian, but it is not language-dependent [10]. It is calculated as the inverse ratio of the perplexity obtained on the original, authentic text, and the perplexity calculated on the text created by randomizing the word order on a sentence level in the original text, as shown in the equation (7), where *ppl* marks perplexity and *KD* is the discrimination coefficient.

$$KD = \frac{ppl(randomized\_text)}{ppl(original\_text)} \qquad (7)$$

The *KD* values are practically uncorrelated with the test data set used in the evaluation, which makes them more appropriate for the comparison of the models evaluated under different conditions.

The sizes of the models created for this experiment were reduced gradually by adapting the counts threshold for the first method, and the model's perplexity increase threshold for the second method. The results of the evaluation of these models are presented in the following section.

IV. Experiment, Results and Discussion

The test data set for LM evaluation has been created by extracting every 100th sentence from the original corpus in order to create a mixture of as many literary styles and thematic categories as possible. These sentences represent around 1% of the entire corpus and they were, of course, excluded from the training corpus. The models were trained on previously defined vocabularies. One of them consisted of 2000 words and the other one of 10000. The corresponding lemma-based vocabularies consisted of 1770 and 6753 entries, respectively.

The results of the perplexity and discrimination coefficient evaluation for the word-based models trained on both smaller and larger vocabularies are given in Table I along with the n-gram file sizes corresponding to count thresholds of 1, 5, 6, 7, 8, and 9. The perplexity values for the models trained on the vocabulary consisting of 2000 words show almost no change in the model's quality even when the number of kept n-gram probabilities is reduced to less than 20% of the original.

TABLE I. Evaluation results and sizes of the word- based language models for different counts threshold

Counts	**Vocabulary of 2000 words**			**Vocabulary of 10000 words**		
	Size in entries (x10^3)	***Perplexity***	***Discrimin. coefficient***	***Size in entries (x10^3)***	***Perplexity***	***Discrimin. coefficient***
1	410	58	5,5	1580	129	10,9
5	134	59,8	4,9	358	148,1	8
6	114	60,2	4,8	297	151,9	7,6
7	100	60,5	4,7	254	155	7,3
8	89	60,8	4,6	223	157,7	7
9	81	61,1	4,6	**200**	**160,1**	**6,8**

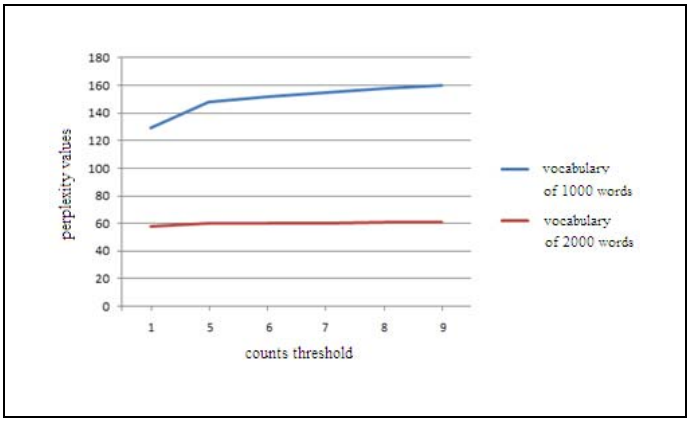

Figure 1. Evaluation perplexity as a function of counts threshold for models trained on vocabularies of 2000 and 10000 words

TABLE II. Evaluation results and sizes of the word-based language models for different perplexity threshold

Perplexity increase threshold (x10^{-7})	**Evaluation perplexity**	**Size in entries (x10^3)**
1	130	1000
10	145	280
27	158.4	138
29.9	**160.1**	**128**
40	165	100
50	169	88
100	186	53

The evaluation of the models trained on the larger vocabulary show more significant quality deterioration when high counts threshold is set, which is also shown in Fig. 1 for clarity purposes. This was to be expected because the models trained on large vocabularies are generally more sensitive to the size reduction because a greater number of significant n-grams have low counts (especially when small training corpora are used).

It is interesting to compare the perplexity values obtained for the models trained on the small vocabulary with the other group of models. The evaluation shows that the models trained on the smaller vocabulary are generally a much better language representation, which is not correct. This is only a consequence of the perplexity calculation method in which the OOV words are omitted from the evaluation process, which can be seen in the SRILM evaluation reports. On the other hand, the discrimination coefficients show a more realistic situation because, even though the perplexity values are greater for the models trained on the larger vocabulary, the ratio of these values and the values obtained on the random word sequences shows that this group is in fact more powerful in determining whether the textual content is a valid word sequence.

The reduction of the models using the entropy-based pruning technique was only done on the models trained on the 10000-word vocabulary. The perplexity threshold was varied in order to find a value at which the resulting model can be directly compared to a model obtained by the previous reduction method. The results are given in Table II. The sizes of the models reduced by using the entropy-based are significantly smaller than the models reduced by increasing the counts threshold with the same evaluation perplexity values. The values marked in bold of the text in Tables I and II represent the two models of the same quality, but very different sizes. The size of the model acquired by using the entropy-based method is in this case smaller then the size of the corresponding model by more than 35%.

The same experiment was also done on the lemma-based models. The results for the reduction by increasing the counts threshold are given in Table III, and for the entropy-based pruning in Table IV. These results also show that the entropy-based reduction method is much more successful. Here, the models obtained by different reduction methods but equivalent from the point of view of that the entropy-based method resulted in a model of a size smaller by more than 36%.

V. Conclusion

This paper presents the results of a research the object of which was finding a suitable method for reducing the size of the language model for the Serbian language. This experiment was a part of a research on the language modeling conducted in order to improve the language model quality and implementation for use within a LVCSR system for Serbian. Two methods have been compared, one which considers increasing the n-gram counts threshold, and the other one based on entropy-based pruning. The latter method showed significantly better results.

TABLE III. Evaluation results and sizes of the lemma-based language models for different counts threshold

Counts	**Vocabulary of 1170 lemmas**			**Vocabulary of 6753 lemmas**		
	Size in entries (x10^3)	***Perplexity***	***Discrimin. coefficient***	***Size in entries (x10^3)***	***Perplexity***	***Discrimin. coefficient***
1	824	69	6.7	2906	121	15.3
5	266	72	5.7	680	142	10.6
6	223	72.8	5.6	556	146.7	10.1
7	193	73.5	5.4	470	150.9	9.5
8	170	74.1	5.3	407	154.4	9.1
9	153	74.6	5.2	**360**	**157.5**	**8.7**

TABLE IV. Evaluation results and sizes of the lemma-based language models for different perplexity threshold

Perplexity increase threshold (x10^{-7})	**Evaluation perplexity**	**Size in entries (x10^3)**
1	125.1	1396
10	150.9	295
13	155.8	241
14	**157.3**	**228**
15	158.6	216
30	174.9	124
100	215.4	46

Acknowledgement

This work was supported by the Ministry of Education, Science and Technological Development of Serbia within the Project “Development of Dialogue Systems in Serbian and other South Slavic Languages” (TR-32035).

References

[1] G. Zweig, M. Picheny, “Advances in large vocabulary continuous speech recognition”, Advances in Computers, Elsevier, 2004, vol. 60, pp. 249-291.

[2] T. Brants, A. C. Popat, P. Xu, F. J. Och, J. Dean, “Large language models in machine translation”, Proceedings of the EMNLP-CoNLL, pp. 858-867, 2007.

[3] S. Ostrogonac, D. Mišković, M. Sečujski, D. Pekar, V. Delić, “A language model for highly inflective non-agglutinatve languages,” SISY, Subotica, 2012.

[4] S. Ostrogonac, M. Sečujski, D. Mišković, “Impact of training corpus size on the quality of different types of language models for Serbian”, 20. Telecommunications forum TELFOR, Belgrade, 20-22 November, 2012.

[5] A. Stolcke, “SRILM - an extensible language modeling toolkit,” Proceedings of ICSLP, vol. 2, pp. 901-904, Denver, USA, 2002.

[6] A. Stolcke, “Entropy-based pruning of backoff language models”, Proceedings DARPA Broadcast News Transcription and Understanding Workshop, pp. 270-274, Lansdowne, VA, 1998.

[7] M. Sečujski, R. Obradović, D. Pekar, LJ. Jovanov, V. Delić, “AlfaNum system for speech synthesis for Serbian language,” Proceedings of Text, Speech and Dialogue, LNAI 2448, pp. 237-244, London, UK, 2002.

[8] A. Kupusinac, M. Sečujsk, “An algorithm for part-of-speech tagging in Serbian language”, 9. ISIRR, Novi Sad, Serbia, 21-22 June, 2007, pp. 43-43.

[9] http://en.wikipedia.org/wiki/Katz's_back-off_model, January 2013.

[10] S. Ostrogonac, D. Mišković, M. Sečujski, D. Pekar, “Discriminative potential of a language model based on the class n-gram concept,” in Serbian (“Diskriminativne mogućnosti modela jezika zasnovanog na konceptu klasnog n-grama”), DOGS, Kovačica, 2012.