Review of statistical modeling of highly inflected Lithuanian using very large vocabulary

Airenas Vaičiūnas, Gailius Raškinis

Department of Applied Informatics
Vytautas Magnus University, Kaunas, Lithuania
airenas@freemail.lt, idgara@vdu.lt

Abstract

This paper presents state of the art language modeling (LM) of Lithuanian, which is highly inflected free word order language. Perplexities and word error rates (WER) of standard n-gram, class-based, cache-based, topic mixture and morphological LMs were estimated and compared for the vocabulary of more than 1 million words. WER estimates were obtained by solving a speaker-dependent ASR task where LMs were used to rescore acoustical hypothesis. LM perplexity appeared to be uncorrelated with WER. Cache-based language models resulted in the greatest perplexity improvement, while class-based language models achieved the greatest though insignificant WER improvement over the baseline 3-gram.

1. Introduction

Language model (LM) is an important component of any large vocabulary continuous speech recognition (LVCSR) system. LM assigns a probability estimate to every word sequence and allows the LVCSR system to rate hypothesized sentences. Statistical language modeling of fixed word order non-inflected languages has been extensively studied [2]. In contrary, language modeling of inflected, free word order languages received much less attention. Conventional n-gram modeling of inflected languages results in difficulties known as huge vocabulary size, LM sparseness, high LM perplexity, and significant out-of-vocabulary (OOV) word rate. A set of statistical LM techniques including class-based, cache-based, morphology-based, topic mixture based and particle-based modeling have been developed to cope with these problems. A few of the abovementioned techniques were investigated and compared for inflected Czech [11, 12], Finnish [7, 13], Russian [4] and Slovenian [14].

In this paper, we compare class-based, cache-based, morphology based and topic mixture based LM techniques for Lithuanian, which is also a highly inflected, free word order language. Models are based on the vocabulary of over 1 million word forms. Both information-theoretic perplexity criterion and WER estimates obtained as the result of LM integration into LVCSR system are used to compare the performance of different LM techniques.

2. Statistical language modeling techniques

In this section, the set of investigated statistical LM techniques is briefly described.

All investigated LM techniques were based on a common 3-gram approach for estimating the probability of a word sequence $W = w_1\ w_2\ ...\ w_n$:

$$P(W) \approx \prod_{i=1}^{n} P(w_i \mid w_{i-2} w_{i-1}) \tag{1}$$

LM techniques differed by the way they estimated probabilities $P(w_i \mid w_{i-2}w_{i-1})$.

Our baseline model was a word 3-gram with an interpolated form of Kneser-Ney smoothing [8].

$$P_{KN}(w_i \mid w_{i-2} w_{i-1}) = \frac{C(w_{i-2} w_{i-1} w_i) - D_3}{C(w_{i-2} w_{i-1})} + \\ + \lambda(w_{i-2} w_{i-1}) P_{MKN}(w_i \mid w_{i-1}) \tag{2}$$

where $C()$ is a count function, D_3 is a discount value, $\lambda(w_{i-2}\ w_{i-1})$ is a normalizing factor and

$$P_{MKN}(w_i \mid w_{i-1}) = \frac{|\{v \mid C(v w_{i-1} w_i) > 0\}| - D_2}{\Sigma_w |\{v \mid C(v w_{i-1} w) > 0\}|} + \\ + \lambda(w_{i-1}) P_{MKN}(w_i) \tag{3}$$

Here, the term $|\{v \mid C(vw_{i-1}w_i) > 0\}|$ denotes a count of distinct words preceding a word pair $w_{i-1}\ w_i$. Discount values are optimal values for absolute discounting [9] $D_n = C_1/(C_1+2C_2)$, where C_1 and C_2 are the total number of n-grams with one and two counts, and n is the order of the higher-order model being interpolated. $P_{MKN}(w_i)$ can be defined in an analogous way to $P_{MKN}(w_i \mid w_{i-1})$ [8].

Kneser-Ney smoothing technique was also used for all class-based, cache-based, topic mixture, and morphological models wherever a 3-gram smoothing was necessary.

2.1. Class-based models

We have investigated class-based models of the type:

$$P_{Class}(w_i \mid w_{i-2} w_{i-1}) = \lambda P_{KN}(w_i \mid w_{i-2} w_{i-1}) + \\ + (1-\lambda) P_{3C}(c_i \mid c_{i-2} c_{i-1}) \cdot P_{WC}(w_i \mid c_i) \tag{4}$$

where $P_{3C}(c_i \mid c_{i-2}\ c_{i-1})$ are estimates of the conditional probability of seeing class c_i given the two previous classes $c_{i-2}\ c_{i-1}$, $P_{WC}(w_i \mid c_i)$ are 1-gram distributions of words assigned to the class c_i, and λ are model weights found by an expectation maximization (EM) algorithm [1] on the development corpus.

Model (4) assumes that each word w_i is assigned to a single class c_i. Word assignment to classes (or clustering) was performed by an iterative hill climbing algorithm [4, 10] by maximizing the log likelihood function:

$$L(f) = \frac{1}{n} \sum_{i=1}^{n} \log P(w_i \mid c_i) P(c_i \mid c_{i-1}) \tag{5}$$

where summing is over all training corpus. The clustering algorithm was run for two iterations with random initialization.

2.2. Cache-based models

We have investigated regular and decay cache language models. Decay cache LMs systematically outperformed regular cache LMs. Unigram decay cache LM [3] is given by:

$$P_{WH}(w_i \mid h) = \frac{1}{\beta} \sum_{j=i-K_d}^{i-1} I\{w_i = w_j\} d(i-j) \quad (6)$$

where h is the history of the word, β is a normalizing factor, $I\{condition\}$ is the indicator function, which takes the value of 1 if *condition* is true, and 0 otherwise, and $d(x)$ is the decay function that tends to zero as x increases (x corresponds to distance). The finite sum parameter K_d = 1000 was used instead of infinite decay in order to speed up the calculations.

We tested two types of decay functions: the common exponential decay function $d(x)=e^{-bx}$ [3] where decay speed b was found experimentally by minimizing perplexity on the development corpus and the function $d_{data}(x)$ which represented the histogram of distances of word reoccurrence estimated on the training corpus:

$$d^1_{data}(x) = \sum_{i=x+1}^{N} I\{w_i = w_{i-x} \wedge Occ(i,x) = 0\} \text{ and} \quad (7)$$

$$Occ(i,x) = \sum_{j=i-x+1}^{i-1} I\{w_i = w_j\}$$

Here $Occ(i, x) = 0$ means that the word w_i does not reoccur within the interval w_{i-x+1}, ..., w_{i-1}. An extended version of (7) taking into account not only the first but also the second word reoccurrence can be defined as:

$$d^2_{data}(x) = \sum_{i=x+1}^{N} I\{w_i = w_{i-x} \wedge Occ(i,x) \le 1\} \quad (8)$$

The decay function (8) slightly outperformed the best exponential decay function.

Bigram cache LM P_{2WH}, which could be defined in a way analogous to (6), contributed very much to perplexity reduction as well. Thus, the final interpolated cache-based language model was defined as follows:

$$P_{Cache}(w_i \mid w_{i-2}, w_{i-1}) = \lambda_1(h) P_{KN}(w_i \mid w_{i-2}, w_{i-1}) +$$
$$+\lambda_2(h) P_{WH}(w_i \mid h) + \lambda_3(h) P_{2WH}(w_i \mid w_{i-1}, h) \quad (9)$$

where $\lambda_i(h)$ are dynamically changing model weights. They were found on the previous word history w_{i-L}, ..., w_{i-1} using an iterative EM method [1]. For example, λ_1 was calculated as follows:

$$\lambda_1^{z+1} = \frac{1}{L} \sum_{m=i-L}^{i-1} \frac{\lambda_1^z P_{KN}(w_i \mid w_{i-2}, w_{i-1})}{P_{Cache}(w_i \mid w_{i-2}, w_{i-1}, \lambda_1^z \lambda_2^z, \lambda_3^z)} \quad (10)$$

where L was found by minimizing perplexity on the development corpus. EM algorithm was initialized with $\lambda_1^0 = \lambda_2^0 = \lambda_3^0 = 1/3$. Experiments showed that dynamic interpolation weights outperform static weights.

2.3. Topic mixture models

We have investigated topic mixture language models of the type:

$$P_{Topic}(w_i \mid w_{i-2}, w_{i-1}) = \sum_{j=0}^{K} \lambda_j(h) P_j(w_i \mid w_{i-2}, w_{i-1}) \quad (11)$$

where K is the number of distinct topics within training corpus, P_j $1 \le j \le K$ are the trained 3-gram models for each topic, $P_0 = P_{KN}$ is a general 3-gram model (2), and $\lambda_j(h)$ are dynamic model weights what were obtained in the same way as for the cache models (10).

Automatic text clustering into a fixed number of topics was performed by a simple greedy search algorithm [3, 10] using 1-gram perplexity criterion. Clustering algorithm was run for 2 iterations with random initialization. It should be noted that text clustering at the base of word base forms (lemmas) outperformed clustering based on the words themselves.

2.4. Morphological models

We have investigated LMs based on the morphological decomposition of words in an attempt to achieve lower OOV rates. Morphological lemmatizer-based [15] word splitting function M: $w \rightarrow <s, g>$ was designed which divided every word w into its base form s and its part of speech (POS) g. In case of morphological ambiguity s was assigned the first base form from the list of base forms returned by the lemmatizer, and g was assigned the hyper-tag consisting of the concatenation of all POS tags in the list. As some non inflected frequent words such as prepositions were supposed to be important to the sequence of morphological information, a list of about 150 words was prepared for which we set $s = w$ and $g = w$. An illustration of word splitting function is given below:

Table 1: Illustration of morphological word splitting

Word, w	Base form, s	POS tag, g
vaikas (child)	s=vaikas	g=(noun masc. sg. nom.)
vaikai (children; [you] dissipate)	s=vaikas	g=(noun masc. pl. nom. + noun masc. pl. voc. + verb ind. pres. t. sg. 2nd pers.)
po (after)	s=po	g=po

We have investigated 2 types of morphological language models:

$$P_{Morf-1}(w_i \mid w_{i-2} w_{i-1}) = P_{3S}(s_i \mid s_{i-2} s_{i-1}) \cdot \cdot P_{3GS}(g_i \mid g_{i-2} g_{i-1} s_i) \quad (12)$$

$$P_{Morf-2}(w_i \mid w_{i-2} w_{i-1}) = P_{3S}(s_i \mid s_{i-2} s_{i-1}) \cdot \cdot (\lambda P_{GS}(g_i \mid s_i) + (1-\lambda) P_{3G}(g_i \mid g_{i-2} g_{i-1})) \quad (13)$$

where $P_{3S}(s_i \mid s_{i-2}\ s_{i-1})$ is the probability estimate of seeing word base form s_i given the two preceding word base forms s_{i-1} and s_{i-2}, $P_{3G}(g_i \mid g_{i-2}\ g_{i-1})$ is the probability estimate of seeing POS tag g_i given the two preceding POS tags g_{i-1} and g_{i-2}, $P_{3GS}(g_i \mid g_{i-2}\ g_{i-1}\ s_i)$ is the probability estimate of seeing POS tag g_i given the word base form s_i and the two preceding POS tags g_{i-1} and g_{i-2}, $P_{GS}(g_i \mid s_i)$ is the probability estimate of seeing POS tag g_i given the word base form s_i. Models P_{3S}, P_{3GS}, P_{GS} and P_{3G} were smoothed using Kneser-Ney discounting technique (2).

2.5. Model evaluation

Perplexity PP_M was estimated for every investigated language model P_M on test corpus:

$$PP_M = 2^{\hat{H}}, \quad \hat{H} = -\frac{1}{N} \log_2 P_M(w_1 ... w_N) \quad (14)$$

where N is a word count of the test corpus. Perplexity improvement the language model P_M brought with respect to the baseline model P_{KN} was defined as follows:

$$improvement = ((PP_{KN} - PP_M) / PP_{KN}) \times 100\% \quad (15)$$

3. Impact of the language model on perplexity

3.1. Text corpora and tools for building LMs

Our experiments were based on a 84 million word Lithuanian text corpus compiled by the Center of Computational Linguistics at Vytautas Magnus University (henceforth, VMU text corpus). VMU text corpus represented a great variety of genres and topics of the present day written Lithuanian. It included texts from newspapers, journals, novels, books, law and administrative documents. VMU text corpus was manually divided into training, development (optimization) and test sub-corpora consisting of 98%, 1% and 1% of the original corpus respectively. An effort was made to have the same proportions of text genres within training, development and testing sub-corpora. Some text clearing was performed. All punctuation was removed and all numbers where replaced by the single tag <num>. The vocabulary of the training corpus contained 1158383 distinct word forms.

Majority of our investigations were carried out using locally developed LM tools. We also used CMU-Cambridge Statistical Language Modeling Toolkit [5]. Morphological analysis was performed by the lemmatizer of Lithuanian [15].

3.2. Baseline language model

Word 3-gram P_{KN} with no cutoffs and smoothed by Kneser-Ney discounting technique (2) was selected to be our baseline model. Table 2 shows perplexities and OOV rates of P_{KN} as the function of vocabulary size. Perplexity obtained with the full vocabulary of 1158383 words was taken as the baseline result.

Table 2: Perplexities and OOV rates of P_{KN} model for different vocabulary sizes

Vocabulary size	Perplexity	OOV, %
65k	561.87	11.22
100k	635.72	8.56
500k	919.34	2.89
1M	1008.22	1.90
1158383(all)	**1027.20**	**1.72**

3-gram language model P_{GT} using Good-Turing smoothing with Katz backoff scheme [1] was also created. However, it showed 8.78% degradation in comparison to P_{KN} model (table 3).

3.3. Comparison of LM techniques

In this section we briefly discuss the results obtained by LMs built using techniques described in section 2.

Table 3: Impact of LM technique

Model	Perplexity	Impr., %	OOV, %
P_{GT}	1117.42	-8.78	1.72
P_{KN}	**1027.20**	---	**1.72**
P_{Class}	843.73	17.86	1.72
P_{Cache}	655.25	36.21	1.72
P_{Topic}	732.72	28.67	1.72
$P_{Morph-1}$	1696.08	-65.11	1.15
$P_{Morph-2}$	1768.10	-72.12	1.15

Class-based LM. A few different class-based language models were built and investigated. Words were automatically clustered into 100, 500, 1000, 3000 and 5000 classes. The improvement of 3.93%, 13.85% and 17.62% was obtained for LMs based on 100, 1000 and 3000 classes respectively. The best improvement was obtained for 5000 classes (table 3). Though LMs based on more than 5000 classes weren't investigated the relative improvement gained by passing from 3000 to 5000 classes appeared to be insignificant. This suggested that the perplexity would not be improved very much if the number of classes would be still further increased.

Cache-based LM. Cache-based language models appeared to be the best among investigated types of statistical LMs in terms of perplexity reduction. Regular 1-gram cache LM improved the perplexity up to 24.71% (the best cache size was equal to 500 words). Decay cache based on function (8) added about 3% of improvement. Decay 2-gram cache LM resulted in 33.31% perplexity improvement. The best result was achieved with the interpolated 1-gram, 2-gram cache LM (9) using dynamically updated model weights (table 3). The optimum word history length L = 200 for dynamically updating model weights was obtained on the development text corpus.

Topic mixture LM. The impact of a number of mixtures, upon which the topic mixture LM is based, was investigated. 1910 texts of the training corpus were automatically clustered into K = 4, 8, 11, 16, 32, 64, and 128 topics. Perplexity estimates were almost identical for LMs based on 32, 64 and 128 topics with the best estimate at K = 32 (table 3). Models having less than 32 mixtures performed significantly worse. The optimum length L of the word history used for dynamic mixture weight adaptation (10) grew as the number of mixtures increased. The best values for L were 100, 200 and 300 for the number of mixtures under 32, 32, and above 32 mixtures respectively. The automatic and expert-based manual text clustering into topics were also compared. LMs based on 11 automatically clustered topics (PP = 784.82) outperformed LMs built upon manually designed topic clusters (PP = 834.73).

Morphological LM. Though morphological language models achieved lower OOV rates, they had much greater perplexities. It is interesting to note that perplexity optimization of P_{Morf-2} (13) resulted in $\lambda = 0.995$ meaning that POS 3-gram P_{3G} component of P_{Morf-2} was practically ignored.

4. Impact of the LM on WER

In order to assess the relative impact of various LM techniques on speech recognition performance, HMM speech recognition system was developed using HTK toolkit [6]. Acoustic models of Lithuanian phones were based on a 3-state left-to-right HMMs. HMM states were defined as a mixture of 8 Gaussian distributions. Word-internal triphone HMMs were built and clustered using decision tree HMM state clustering approach.

Acoustic feature vectors consisted of 39 normalized parameters (12 MFCC + log energy, deltas, accelerations). Features vectors were calculated every 10ms over a 25ms frames.

The speaker-dependent LVCSR system was trained on 13h of speech data coming from a single speaker. Test data consisted of 40 min of speech of the same speaker (4461 words, 1096 utterances).

Recognition network represented a word-loop of the 65k most frequent Lithuanian words. No word probability information was added to it. About 14% of words of the test utterances were absent from the recognition network.

During the recognition phase, the list of 1000 acoustically most probable word sequences was generated for each test utterance and later rescored taking into account LM probabilities. Let W denote the word sequence, let $P(A|W)$ be the acoustic probability of W, let $P(W)$ be the LM probability of W, and let $C(W)$ denote the word count of W. The score was computed as follows:

$$score(W) = \log P(A \mid W) + s \log P(W) + ipC(W) \quad (16)$$

where s is a LM scale factor, and ip is a word insertion penalty.

Word sequence having the best score was taken as the recognized utterance. It was added to the word history, so it could be used for the subsequent estimation of LM probabilities of the next utterance.

The values of the parameters s=25 and ip=-30 were optimized on pilot experiments. Results of the speech recognition experiments are summarized in table 4.

Table 4: Impact of the LM to WER

LM Model	WER,%	Perplexity	Oov, %
None	58.35	---	---
Oracle	36.16	---	---
P_{KN}	43.91	3260.42	1.63
P_{GT}	44.45	3591.50	1.63
P_{Class} (3000 classes)	**43.56**	2583.61	1.63
P_{Cache}	44.79	2817.67	1.63
P_{Topic}(32 topics)	44.68	1934.15	1.63
$P_{Morph-1}$	46.76	5503.07	0.85

ASR system without LM rescoring resulted in 58% WER. The WER of 36% was the best theoretically possible WER that could be obtained by manually picking the closest word sequence within 1000-best list of each test utterance. LMs used for rescoring were built upon the VMU text corpus (see section 3) having vocabulary size of 1158383. Table 4 shows LM perplexities and OOV rates estimated on test utterances. LM perplexities showed that the language complexity of the utterances to be recognized is greater in comparison to the test part of VMU text corpus. Perplexities of various LMs on test utterances followed the same trend as shown in table 3, except for the cache model which performed significantly worse. Detailed investigation of the cache-based LM performance showed that worsening is related to the particularities of the text of test utterances.

Only class-based language model achieved insignificant WER improvement with respect to the baseline trigram. Topic mixture model showed the lowest perplexity, but did not resulted in reduction of WER. Though more detailed investigation of this phenomenon is necessary, it seems that adaptive models (cache, topic mixture) suffer from the "error locking", i.e. from adaptation to the recognized text where almost every second word is erroneous.

Morphological model (simplified formula (13) with λ=1) showed the worst result. However it should be too early to conclude that this model is really worse than others presented in table 4. The experimental scheme did not allow to exploit the advantage of morphological models – its lower OOV rate.

5. Conclusions

In this paper we compared several LM techniques for highly inflected Lithuanian. Up to 36% perplexity improvement was achieved with the cache-based language models. Class-based and topic mixtures LMs also resulted in a noticeable improvement in perplexity. Though morphological language models reduced OOV rates from 1.72% to 1.15%, their perplexities were much greater.

Only class-based models resulted in slight reduction of WER in speech recognition experiments. Other types of LMs resulted in WER degradation with respect to the baseline 3-gram.

6. References

[1] F. Jelinek, "Statistical Methods for Speech Recognition", *Massachusetts Institute of Technology, Cambridge,* 2001.

[2] J. T. Goodman, "A Bit of Progress in Language Modeling", *Computer Speech and Language, 15(4):403–434,* 2001.

[3] P. Clarkson, "Adaptation of Statistical Language Models for Automatic Speech Recognition", PhD thesis, *Cambridge University Engineering Department, Cambridge*, 1999.

[4] E. W. D. Whittaker, "Statistical Language Modelling for Automatic Speech Recognition of Russian and English", PhD thesis, *Cambridge University, Cambridge,* 2000.

[5] P. Clarkson, R. Rosenfeld, "Statistical Language Modeling Using the CMU-Cambridge Toolkit", *Eurospeech'1997, 2707-2710*, 1997.

[6] S. Young, D. Kershaw, J. Odell, D. Ollason, V. Valtchev, P. Woodland, "The HTK Book", *Entropic, Cambridge*, 2000.

[7] J. Kneissler, D. Klakow, "Speech recognition for huge vocabularies by using optimized sub-word units", *Eurospeech'2001, 69–72*, 2001.

[8] R. Kneser, H. Ney: "Improved Backing-Off for m–gram Language Modeling", *ICASSP, 181*-184, 1995.

[9] Ney H., U. Essen and R. Kneser, "On structuring probabilistic dependences in stochastic language modelling", *Computer Speech and Language*, *8(1):1-38*, 1994.

[10] Kneser, R. and J. Peters, "Semantic clustering for adaptive language modeling", *ICASSP, 779-782*, 1997.

[11] W. Byrne, J. Hajic, P. Ircing, F. Jelinek , S. Khudanpur, P. Krbec, J. Psutka, "On Large Vocabulary Continuous Speech Recognition of Highly Inflectional Language – Czech", *Eurospeech'2001, B14, 487-491*, 2001.

[12] P. Ircing, J. Psutka, "Comparison of Word-based and Class-based Language Models for Speech Recognition of the Czech Weather Forecast", *IICSPAT, Dallas*, 2000.

[13] V. Siivola , M. Kurimo, K. Lagus, "Large Vocabulary Statistical Language Modeling for Continuous Speech Recognition in Finnish", *Eurospeech'2001, B25 737–741*, 2001.

[14] M. Sepesy Maucec, Z. Kacic, "Topic Detection for Language Model Adaptation of Highly-Inflected Languages by Using a Fuzzy Comparison Function", *Eurospeech'2001, A42 243 – 247,* 2001.

[15] V. Zinkevičius, "Lemuoklis – tool for morphological analysis", *Darbai ir dienos, Kaunas, 245-274*, 2000 (in Lithuanian).