Comparative Study on Bigram Language Models for Spoken Czech Recognition

Dana Nejedlova

SpeechLab, Technical University of Liberec,
Halkova 6, 461 17 Liberec, Czechia
dana.nejedlova@vslib.cz

Abstract. The article deals with the problem of continuous speech recognition of Czech language. The main goal of this study is to compare various kinds of bigram language models with respect to the accuracy and speed of speech recognition. The main types of bigram language models are described here as well as multiple parameters that affect the performance of a speech recognition system. A comparison with a zerogram model is also made. Different models and various parameter settings are compared by means of the accuracy rate in extensive experiments done with a large test database of 1,600 Czech sentences recorded by 40 speakers.

1 Introduction

Systems that recognize voice input in a form of isolated words or short discrete phrases have been available for Czech language for several years. Some perform well even for lexicons with thousands of words, and they have been already applied in public services [1]. The recognition of fluently spoken language is a much more challenging task. So far no usable system capable of doing this in Czech language has been brought to market. Its development is the main goal of the long-term research in our lab. Though some specific problems of Czech language make the task extremely difficult. Czech language has approximately 1 million word-forms, which is about 20 times more than in English. Such a large lexicon is a result of complex rules for inflection of Czech words. Many Czech words sound similarly having only different prefixes or suffixes. In comparison to English, the word order in Czech sentences is freer, which worsens the utility of language models. Nevertheless, this article shows that language models help substantially in the recognition of Czech. Experiments described in this article have been carried out with the use of the software system described in the complementary paper [2].

2 Test Speech Database

The people recorded for our both training and test databases were not professional speakers. In some cases it was hard even for a human to understand the complete sentence uttered by some of our speakers.

But even if the sentences had been uttered correctly, there would still have remained a space for some mistakes that could not be prevented. Those mistakes originate in words that have a different spelling but the same phonetic transcription. 1,600 test sentences contained 126 words of that kind having 44 different spelling word-forms. Our test speech database had 16,027 words having 7,033 different word-forms. It means that nearly 0.8% of words could be messed up, if they had been classified only with the use of acoustic model by some perfect recognizer.

More details about our test (evaluation) speech database can be seen in [2].

3 The Role of Various Parameters in Speech Recognition System

3.1 Searching for the Optimal Parameter Settings

Our speech recognition system was controlled by several parameters whose optimal settings had to be searched for by making small perturbations of one of them, the rest staying constant, picking up the best value, and then making perturbations of another parameter. All these tuning experiments had to be done on a relatively large amount of the test data. We have used 800 sentences that were a subset of our test database, and our working vocabulary contained 3,622 words. If the database for finding the best settings had been smaller, the optimal settings of parameters found for it would have been far from optimal for the whole test database. We have used the following parameters to control the speech recognition:

1. acoustic model,
2. language model,
3. language model factor,
4. word insertion penalty,
5. number of word-end hypotheses,
6. prune threshold.

3.2 Acoustic Model

We have used either 16 or 32-mixture acoustic models trained on an independent speech database. Paper [2] describes the training speech database and compares the performances of the two types of acoustic models. 32-mixture model works more precisely at the expense of somewhat higher time consumption. All results shown in this article have been done using the 32-mixture acoustic model which we prefer to the 16-mixture one.

3.3 Language Model

Admissible values of the parameter specifying the language model can have the following effects:

1. using no language model (i.e. using a zerogram model),
2. using a dependent language model which has been made of the test data,
3. using some independent language model,
   a) unsmoothed,
   b) smoothed,
      - by adding one,
      - by linear interpolation,
      - by Witten-Bell discounting.

3.4 Language Model Factor and Word Insertion Penalty

Higher values of language model factor (C_{LM}) mean greater influence of language model on recognition. As it is shown below, the optimal value of this parameter was different for different language models.

Our recognition system has a tendency of preferring shorter words to longer ones. To suppress this phenomenon we have introduced the parameter called word insertion penalty (C_{IP}) that worsens each candidate word's score, which really had an improving effect on the accuracy of recognition. Similarly to the case of the language model factor mentioned previously, there is some optimal value of C_{IP} for each type of language model. We have used higher absolute values of C_{IP} with a negative sign for a greater word penalization. More details about C_{LM} and C_{IP} parameters are shown in paper [2].

3.5 Number of Word-End Hypotheses and Prune Threshold

The parameter that we denote as the number of word-end hypotheses (C_{WE}) influences the number of bigrams in language model that are examined while inter-word transitions are evaluated. The higher its value is the better is the performance of the recognizer, but when this value reaches a certain level, the performance is no longer better, only the time consumption rises. The role of the parameter called prune threshold or state pruning threshold (C_{PT}) is to reduce the number of words that are kept in memory as the most probable. From the practical point of view, the prune threshold parameter behaves the similar way as the number of word-end hypotheses.

For both C_{WE} and C_{PT} parameters it was necessary to choose optimal values assuring the highest possible performance without letting the time consumption soar too high. Figure 1 in Chapter 5 shows the results of changing the prune threshold or the number of word-end hypotheses while all other parameters are constant. Mathematical details are in paper [2].

4 Language Models

4.1 *N*-gram Language Models

When computing the probability of a given word sequence from sufficiently large amount of training text, we can see that this probability is very close to the probability of the same word sequence computed from some other amount of training text if these two training corpora share the same language characteristics. This property helps us to predict what word will follow if we know some sequence of preceding words in some utterance that has also the same language characteristics as the corpus from which we have derived the probability of the word sequence. This is the basic concept of *n*-gram language modeling. *N*-gram is a conditional probability that if we observe some sequence of $n - 1$ words w_1^{n-1} then some particular *n*-th word w_n will follow. It can be computed as the frequency of the word sequence w_1^n, also called w_1^n count and in further text denoted as $C(w_1^n)$, divided by the frequency of the word sequence w_1^{n-1}. *N*-gram language model is a collection of conditional probabilities for all *n*-word long sequences that can be composed of a given vocabulary. When $n = 2$, we speak about bigrams. In languages with some relatively small vocabulary like English also probabilities of word triples can be computed and these are called trigrams.

N-gram also called statistical language models are successfully employed in tasks where spontaneous speech is recognized because they are robust and flexible in comparison to rule-based models that use some strictly described grammar.

4.2 Maximum Likelihood Estimate

Maximum likelihood estimate (MLE) is the simplest possible *n*-gram model. It contains conditional probabilities for all possible *n*-grams as they appeared in the training data. Word sequences not present in the training corpus have their probabilities equal to 0. In a more formal way, the values in the model are computed according to the equation

$$P\left(w_n \middle| w_1^{n-1}\right) = \frac{C\left(w_1^n\right)}{C\left(w_1^{n-1}\right)} . \tag{1}$$

MLE model is the starting probability database from which all smoothed models can be derived. Our dependent language model was MLE of probabilities in our evaluation database the sentences of which were formed of 14,387 different word pairs. Source data for our independent language models were the 55,841,099-word corpus described in more details in [2]. From this corpus we have extracted all its word pairs that were formed of the 7,033-word vocabulary of our evaluation speech database. The result of this extraction was the database of 1,785,458 different word pairs. It means that non-zero bigrams covered 3.6% of the matrix of all possible word pairs made of our working vocabulary. Albeit there are several software tools for *n*-gram language model smoothing available in the research community, we have developed our own ones.

4.3 Add-One Smoothing

Even in the relatively simple case of bigram model and some limited vocabulary there is no chance to collect so much meaningful texts written in a particular natural language so that the resulting MLE language model contains no zero probabilities. Even the biggest corpora ever collected give most of the probabilities in the MLE model equal to zero. On the other hand, nearly every written text or spoken utterance not present in the training corpus, from which MLE model has been created, contains some bigrams with non zero probabilities that have zero probabilities in the training corpus. Language models work better in speech recognition when they have all their probabilities equal to some number bigger than zero. The process of turning the MLE language model into some form with all probabilities above zero is called smoothing.

The simplest idea of smoothing is so-called add-one smoothing. All possible combinations of *n*-word sequences counts are incremented by one and MLE model is computed from the resulting data. The equation for add-one smoothing is

$$P\left(w_n \middle| w_1^{n-1}\right) = \frac{C\left(w_1^n\right) + a}{C\left(w_1^{n-1}\right) + Va}, \qquad (2)$$

where a is the constant added to all the counts (the model works better when we add less than one) and *V* is the number of words in vocabulary.

4.4 Witten-Bell Discounting

Witten-Bell discounting is introduced as Method C in [3]. Let $C(w_1^n)$ be the count of a particular sequence of *n* words, $C(w_1^{n-1})$ be the count of the first $n - 1$ word sequence (w_1^{n-1} is also called the history of w_1^n), $T(w_1^{n-1})$ be the number (not count) of all *n*-word sequences that begin with the same $n - 1$ word sequence, i.e. the number of all distinct word types that followed a particular $n - 1$ word sequence, and *V* be the size of vocabulary. Then the equations for Witten-Bell discounting are

$$P\left(w_n \middle| w_1^{n-1}\right) = \frac{C\left(w_1^n\right)}{C\left(w_1^{n-1}\right) + T\left(w_1^{n-1}\right)} \quad \text{if} \quad C\left(w_1^n\right) > 0 \qquad (3)$$

and

$$P\left(w_n \middle| w_1^{n-1}\right) = \frac{T\left(w_1^{n-1}\right)}{\left(V - T\left(w_1^{n-1}\right)\right) \cdot \left(C\left(w_1^{n-1}\right) + T\left(w_1^{n-1}\right)\right)} \quad \text{if} \quad C\left(w_1^n\right) = 0\,. \qquad (4)$$

Every smoothing method distributes some probability mass from the n-grams with probabilities above zero to all the n-grams not seen in the training corpus. Witten-Bell discounting does this separately for each group of n-grams that share the same $n - 1$ word history with the purpose to discount more probability mass from those n-gram groups that have relatively more ending words, i.e. that have relatively large $T(w_1^{n-1})$. This probability mass is then distributed among all the other possible words that have not followed the $n - 1$ word sequence in the training corpus. The reason for this strategy is that if some word sequence is followed by a relatively large number of word types in the training corpus then the probability that some other word types can follow it too is bigger.

4.5 Linear Interpolation Smoothing

Linear interpolation smoothing is published as deleted interpolation algorithm in [4]. Its idea is the following: If we have no examples of a particular word triple w_1^3 to help us to compute its trigram $P(w_3|w_1^2)$, we can estimate its probability by using the bigram probability $P(w_3|w_2)$, and if there is no occurrence of the word pair w_2^3 in the training corpus, we can look to the unigram $P(w_3)$. The method in which lower order n-grams are used for higher order n-grams estimation only if higher order n-grams have zero probability in the training corpus is called backoff. Its algorithm was introduced in [5]. The method in which lower order n-grams are always used for higher order n-grams estimation is called interpolation smoothing. Linear interpolation smoothing estimates the trigram probability $P(w_3|w_1^2)$ by adding together trigram, bigram, and unigram probabilities. Each of these is weighted by a linear weight l.

$$P\left(w_3 \middle| w_1^2\right) = l_3 P\left(w_3 \middle| w_1^2\right) + l_2 P(w_3|w_2) + l_1 P(w_3) + l_0 / V \tag{5}$$

and

$$\sum_i l_i = 1 \tag{6}$$

The last term l_0/V in equation (5) represents uniform distribution of unigram probabilities (V is the size of vocabulary). The set of l values is trained using a version of expectation-maximization (EM) algorithm which is also shown in [4]. The training corpus is divided into training and held-out data and the target l values must minimize the cross entropy of smoothed models computed from both training and held-out corpora. Due to compression reasons explained in [2] we prefer language models that have large groups of the same probability values. Models smoothed by means of linear interpolation do not have this property. It can be seen in the equation (5). N-grams that share the same history get different probabilities according to the frequencies of their ending words as well. In the initial phase of the search for the optimal parameter settings described in Chapter 3.1 we have found that linear interpolation models' performance is the second best after Witten-Bell discounting. Because of these reasons we have not implemented linear interpolation models for our whole evaluation database.

5 Experimental Results

Figure 1 illustrates the information given in Chapter 3.6 about the two parameters that are used for the reduction of a search tree in our recognizer.

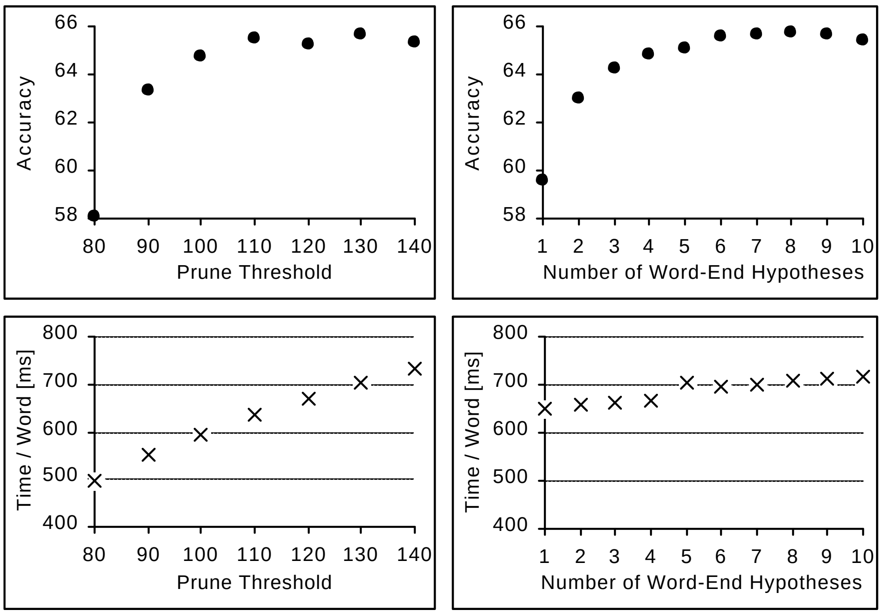

Fig. 1. When the value of prune threshold parameter increases, the accuracy of the recognition rises until it reaches some maximal point. The same rule applies for the case of the number of word-end hypotheses. Time measured on a PC (Athlon 1.3 GHz, 512 MB RAM) rises linearly in proportion to the both parameters

Table 1 compares all language models tested on our 1,600-sentence evaluation database.

Table 1. The optimal parameter settings and the best accuracy achieved for different language models

Language Model	Language Model Factor	Word Insertion Penalty	Number of Word-End Hypotheses	Prune Threshold	Accuracy
Zerogram	0	-42	1	130	48.63
Independent MLE	5	-12	10	130	47.52
Add-One Smoothing	5	-3	10	130	61.58
Witten-Bell	6	-5	10	130	65.48
Dependent MLE	2	-9	10	130	95.82

Accuracy values in Figure 1 and Table 1 are computed according to the standard definition - see for example [2].

6. Conclusion

In this paper we have applied several language models for the recognition of continuous speech in Czech. The results in Table 1 show that Witten-Bell discounting is the best independent bigram model used so far. An interesting demonstration of the necessity of language model smoothing is the fact that accuracy of unsmoothed independent MLE model is slightly worse than zerogram (i.e. when no language model is used). The result of dependent MLE model marks the upper bound of accuracy that can be possibly achieved. Another result shown in Table 1 is the fact that poor language models need greater penalization of each word insertion.

We have compared our language models also by means of their ability to be compressed for the operating memory savings reasons (details are in paper [2]). We are showing in Chapter 4.5 that linear interpolation model's compression feasibility is very low.

When using Witten-Bell language model and its best parameter setting (shown in Table 1) about 12% of sentences were recognized with 100% accuracy.

Acknowledgments. This work has been supported by the Grant Agency of the Czech Republic (grant no. 102/02/0124) and through research goal project MSM 242200001.

References

1. Nouza, J.: A Czech Large Vocabulary Recognition System for Real-Time Applications. Proc. 3rd International Workshop on Text, Speech, Dialogue, Springer-Verlag, Heidelberg, Germany (2000) 217-222.
2. Nouza, J.: Strategies for Developing a Real-Time Continuous Speech Recognition System for Czech Language. In this volume.
3. Witten, I. H. and Bell, T. C.: The Zero-Frequency Problem: Estimating the Probabilities of Novel Events in Adaptive Text Compression. IEEE Transactions on Information Theory, 37(4), (1991) 1085-1094.
4. Jelinek, F. and Mercer, R. L.: Interpolated Estimation on Markov Source Parameters from Sparse Data. In Gelsema, E. S. and Kanal, L. N. (Eds.), Proceedings, Workshop on Pattern Recognition in Practice. North Holland, Amsterdam (1980) 381-397.
5. Katz, S. M.: Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recogniser. IEEE Transactions on Acoustics, Speech, and Signal Processing, 35(3), (1987) 400-401.