Statistical Language Models for Croatian Weather-domain Corpus

Lucia Načinović
Department of Informatics, University of Rijeka
Omladinska 14, Rijeka, Croatia
lnacinovic@inf.uniri.hr

Sanda Martinčić-Ipšić
Department of Informatics, University of Rijeka
Omladinska 14, Rijeka, Croatia
smarti@inf.uniri.hr

Ivo Ipšić
Department of Informatics, University of Rijeka
Omladinska 14, Rijeka, Croatia
ivoi@inf.uniri.hr

Summary

Statistical language modelling estimates the regularities in natural languages. Language models are used in speech recognition, machine translation and other applications for speech and language technologies. In this paper we will present a procedure for language models building for the Croatian weather-domain corpus. Different types of n-gram statistic language models and smoothing methods for language modelling are presented. Those models are compared in terms of their estimated perplexity.

Key words: statistical language modelling, n-gram, smoothing methods, Croatian weather-domain corpus

Introduction

Language models are employed in many tasks including speech recognition, optical character recognition, handwriting recognition, machine translation, and spelling correction (Chen & Goodman, 1998). Speech recognition is concerned with converting an acoustic signal into a sequence of words. Through language modelling, the speech signal is being statistically modelled. Language model of a speech estimates probability Pr(W) for all possible word strings $W=(w_1, w_2,\ldots w_i)$. (Chou & Juang, 2003) Before language models can be estimated, text corpora must be appropriately processed. As can be seen in Figure 1, language models for automatic speech recognition are usually estimated from manual

transcriptions of speech signals and from normalized text corpora. Furthermore, text preparation includes locating appropriate sources of text data and audio transcriptions and processing them in homogeneous manner. (Chou & Juang, 2003)

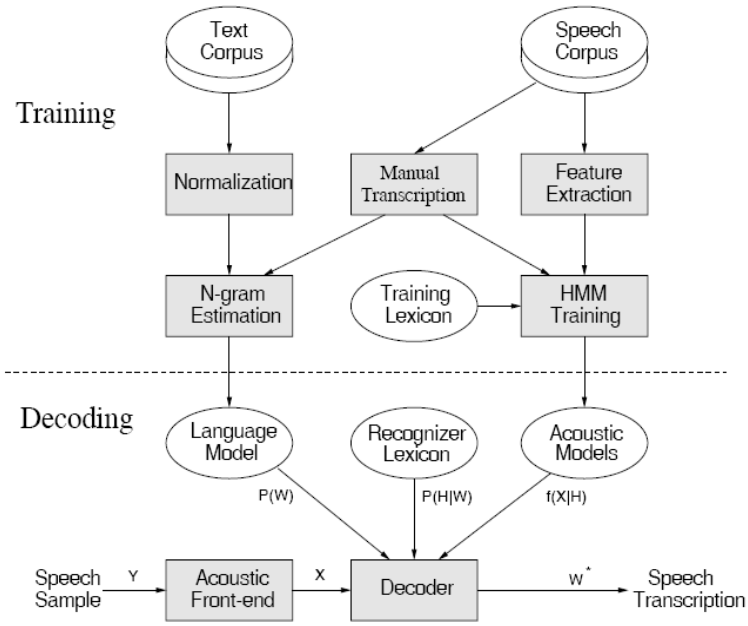

Figure 1: System diagram of a generic speech recognizer based on statistical models, including training and decoding processes and the main knowledge sources (Chou & Juang, 2003).

In this work, we carry out a comparison of the most widely-used smoothing techniques built on Croatian weather-domain corpus using n-grams of various order, and show how these factors affect the relative performance of smoothing techniques which is measured through the estimated perplexities of models.
This paper is organized as follows. The next section gives general information on language models followed by a section on concept of a language model perplexity which is used in this work as a metrics for language models comparison. Then we give information on smoothing and smoothing techniques that we used in our research. Afterwards we describe used text corpus and the implementation of smoothing techniques that we used and we give the obtained results. We end the paper with the conclusion.

Language models

Language models estimate the probabilities of word sequences which are usually derived from large collections of text material (Manning & Schütze, 1999).

The models of word sequences we will consider in this work are probabilistic models - ways to assign probabilities to strings of words, whether for computing the probability of an entire sentence or for giving a probabilistic prediction of what the next word will be in a sequence.
The most widely-used language models are n-gram language models. The central goal of the most commonly used - trigram models, is to determine the probability of a word given the previous two words:

$$p(w_i \mid w_{i-2} w_{i-1})$$

The simplest way to approximate this probability is to compute

$$pML(w_i \mid w_{i-2} w_{i-1}) = \frac{c(w_{i-2} w_{i-1} w_i)}{c(w_{i-2} w_{i-1})}$$

i.e. the number of times the word sequence $w_{i-2}w_{i-1}w_i$ occurs in some corpus of training data divided by the number of times the word sequence $w_{i-2}w_{i-1}$ occurs. This value is called the maximum likelihood (ML) estimate.

Language model perplexity
The most common metric for evaluating a language model is the probability that the model assigns to test data, or the derivative measures of cross-entropy and perplexity. The cross-entropy $H_p(T)$ of a model *p(T)* on data T is defined as

$$H_p(T) = -\frac{1}{W_T} \log_2 p(T)$$

where W_T is the length of the text *T* measured in words. This value can be interpreted as the average number of bits needed to encode each of the W_T words in the test data using the compression algorithm associated with model *p(T)*. The perplexity $PP_p(T)$ of a model *p* is the reciprocal of the average probability assigned by the model to each word in the test set *T*, and is related to cross-entropy by the equation

$$PP_p(T) = 2^{H_p(T)}$$

Clearly, lower cross-entropies and perplexities are better. (Chen & Goodman, 1998)

Smoothing
One major problem with standard N-gram models is that they must be trained from some corpus, and because any particular training corpus is finite, some perfectly acceptable N-grams are bound to be missing from it. (Jurafsky & Martin, 2000). To give an example from the domain of speech recognition, if

the correct transcription of an utterance contains a trigram $w_{i-2}w_{i-1}w_i$ that has never occurred in the training data, we will have $pML(w_i|w_{i-2}w_{i-1})=0$ which will preclude a typical speech recognizer from selecting the correct transcription, regardless of how unambiguous the acoustic signal is.
Smoothing is used to address this problem. The term smoothing describes techniques for adjusting the maximum likelihood estimate of probabilities to produce more accurate probabilities. These techniques adjust low probabilities such as zero probabilities upward, and high probabilities downward. Not only do smoothing methods generally prevent zero probabilities, but they also attempt to improve the accuracy of the model as a whole. (Chen & Goodman, 1998)

Smoothing techniques used in our research
In our research, we used four different smoothing techniques including additive smoothing, absolute discounting, Witten-Bell smoothing technique and Kneser-Ney discounting. General information on each of those smoothing techniques is given bellow.

Additive smoothing
Additive smoothing is one of the simplest types of smoothing. To avoid zero probabilities, we pretend that each n-gram occurs slightly more often than it actually does: we add a factor δ (0< δ ≤1) to every count. Thus, we set

$$p_{add}(w_i \mid w_{i-n+1}^{i-1}) = \frac{\delta + c(w_{i-n+1}^{i})}{\delta \mid V \mid + \sum_{w_i} c(w_{i-n+1}^{i})}$$

where V is the vocabulary, or set of all words considered and *c* is the number of occurrences. Lidstone and Jeffreys advocate taking δ=1 (Chen & Goodman, 1998). We used three different values of δ parameter (0.1, .05 and 1) in our research. More information on how the change of those values affected language models is given in section *Results* bellow. Although additive smoothing does not perform well and is not commonly used, it makes a basis for other smoothing techniques.

Absolute discounting
In absolute discounting techniques, the linear interpolation algorithm is used. When there is little data for directly estimating an *n-gram* probability useful information can be provided by the corresponding *(n-1)-gram*. A simple method for combining the information from lower-order n-gram models in estimating higher-order probabilities is linear interpolation, and a general class of interpolated models is described by Jelinek and Mercer (1980) (Chen & Goodman, 1998):

$$p_{\text{int}erp}(w_i \mid w_{i-n+1}^{i-1}) = \lambda_{w_{i-n+11}^{i-1}} p_{ML}(w \mid w_{i-n+1}^{i-1}) + (1-\lambda_{w_{i-n+1}^{i-1}}) p_{\text{int}erp}(w_i \mid w_{i-n+2}^{i-1})$$

The nth-order smoothed model is defined recursively as a linear interpolation between the nth-order maximum likelihood model and the (n-1)-th-order smoothed model. Given fixed pML, it is possible to search efficiently for the

$$\lambda_{w_{i-n+1}^{i-1}}$$

that maximizes the probability of some data using the Baum–Welch algorithm (Chou & Juang, 2003).

In absolute discounting smoothing instead of multiplying the higher-order maximum-likelihood distribution by a factor

$$\lambda_{w_{i-n+1}^{i-1}}$$

the higher-order distribution is created by subtracting a fixed discount D from each non-zero count:

$$p_{abs}(w_i \mid w_{i-n+1}^{i-n}) = \frac{\max\left\{c(w_{i-n+1}^{i}) - D, 0\right\}}{\sum_{w_i} c(w_{i-n+1}^{i})} + (1-\lambda_{w_{i-n+1}^{i-1}}) p_{abs}(w_i \mid w_{i-n+2}^{i-1})$$

To make this distribution sum to 1, we take:

$$1-\lambda_{w_{i-n+1}^{i-1}} = \frac{D}{\sum_{w_i} c(w_{i-n+1}^{i})} N_1 + (w_{i-n+1}^{i-1}\bullet)$$

Ney et al. (1994) suggest setting D as follows:

$$D = \frac{n_1}{n_1 + 2n_2}$$

where n_1 and n_2 are the total number of n-grams with exactly one and two counts in the training data. According to that formula we came to the value of D=0.06 in our research. Besides this particular value, we also experimented with three more values: 0.3, 0.5 and 1. Information on how changing those values affected language models is given in section *Results*.

Witten-Bell smoothing

The nth-order smoothed model is defined recursively as a linear interpolation between the nth-order maximum likelihood model and the (n-1)-th-order smoothed model. To compute the parameters

$$\lambda_{w_{i-n+1}^{i-1}}$$

for Witten–Bell smoothing, we will need to use the number of unique words that follow the history

$$w_{i-n+1}^{i-1}$$

We will write this value as follows:

$$N_1 + (w_{i-n+1}^{i-1}\bullet) = |\{w_i : c(w_{i-n+1}^{i-1} w_i) > 0\}|$$

We assign the parameters

$$\lambda_{w_{i-n+1}^{i-1}}$$

for Witten–Bell smoothing such that

$$1 - \lambda_{w_{i-n+1}^{i-1}} = \frac{N_1 + (w_{i-n+1}^{i-1}\bullet)}{N_1(w_{i-n+1}^{i-1}\bullet) + \sum_{w_1} c(w_{i-n+1}^{i})}$$

Kneser-Ney smoothing

Kneser and Ney (1995) have introduced an extension of absolute discounting where the lower-order distribution that one combines with a higher-order distribution is built in a novel manner. In previous algorithms, the lower-order distribution is generally taken to be a smoothed version of the lower-order maximum likelihood distribution. However, a lower-order distribution is a significant factor in the combined model only when few or no counts are present in the higher-order distribution. Consequently, they should be optimized to perform well in these situations.

According to Kneser-Ney smoothing, the lower-order distribution such that the marginals of the higher-order smoothed distribution match the marginals of the training data are being selected. For example, for a bigram model we would like to select a smoothed distribution p_{KN} that satisfies the following constraint on unigram marginals for all w_i:

$$\sum_{w_{i-1}} p_{KN}(w_i - 1 w_i) = \frac{c(w_i)}{\sum_{w_i} c(w_i)}$$

The left-hand side of this equation is the unigram marginal for w_i of the smoothed bigram distribution p_{KN}, and the right-hand side is the unigram frequency of w_i found in the training data. (Chen & Goodman, 1998)

Smoothing implementations

In this section we describe our smoothing techniques implementations. 2-gram, 3-gram and 4-gram language models were built. On each of these models, we applied four different smoothing techniques – additive smoothing, Witten-Bell smoothing, absolute discounting and Kneser-Ney smoothing.

Language models were built from the Croatian weather-domain corpus (Martinčić-Ipšić, 2007). Corpus contains 290 480 words, 2 398 1-grams, 18 694 2-grams, 23 021 3-grams and 29 736 4-grams.

Major part of the corpus was developed in the period from 2002 until 2005 by recording radio weather forecasts and some parts were added later. It includes the vocabulary related to weather, bio and shipping forecast, river water levels and weather reports.

We divided corpus into ten parts. We used nine parts as train data for building language models and one part as test data for evaluating those models in terms of their estimated perplexities.

Different language models were built and tested with SRILM language modelling toolkit. (Stolcke, 2002)

Results

After building 2-gram, 3-gram and 4-gram language models and applying different smoothing techniques on those models, the perplexities of models were estimated. Those perplexities are given in Table 1.

As mentioned before, it is usually considered that models with lower perplexities are better. According to that, additive smoothing gave the worst results. The perplexities of models after applying additive smoothing were even higher than those of the models built without implementing smoothing techniques. By increasing the parameter δ in additive smoothing, the perplexities of the built language models increased as well.

Absolute discounting gave the best results with the parameter D=0.3 and the worst with the parameter D=1. With the parameter 0.3, perplexities of 2-gram, 3-gram and 4-gram models were lower than the perplexities of those models without smoothing. However, it gave poor results with the parameter D=1. According to the obtained perplexities, we can also come to the conclusion that absolute discounting gives better results on higher-order n-grams such as 4-grams.

Witten-Bell smoothing gave good results on 2-gram, 3-gram and 4-gram models. The perplexities of the models after implementing the smoothing are lower than the perplexities of those models without smoothing implementation.

The best results gave the implementation of Kneser-Ney smoothing. The perplexities of the models after implementing that smoothing technique are lower than all other perplexities.

The presented results were expected because the used text covered only the weather domain vocabulary.

Table 1: The perplexities of tested language models

	Without smoothing	Additive smoothing			Absolute discounting				Witten-Bell	Kneser-Ney
		δ parameter			D parameter					
		0,1	0,5	1	0,06	0,3	0,5	1		
2-gram	19,87	28,8	51,6	73,5	20,39	19,61	19,64	21,6	19,75	18,96
3-gram	8,45	30,04	86,9	144,2	8,55	8,17	8,22	9,30	8,25	7,63
4-gram	6,04	42,9	142,6	239,87	5,93	5,64	5,71	6,76	5,76	5,24

Conclusion

In this paper we described our research on different smoothing techniques which were applied to language models built from the Croatian weather-domain corpus. 2-gram, 3-gram and 4-gram language models were built. On each of these models, we applied four different smoothing techniques – additive smoothing, Witten-Bell smoothing, absolute discounting and Kneser-Ney smoothing. After we built the models, we estimated and compared perplexities of those models. We came to the conclusion that Kneser-Ney smoothing technique gives the best results.

Further we will prepare the more balanced corpus of Croatian text and thus build more complete language model.

References

Chen, Stanley F.; Goodman, Joshua. An empirical study of smoothing techniques for language modelling. Cambridge, MA: Computer Science Group, Harvard University, 1998

Chou, Wu; Juang, Biing-Hwang. Pattern recognition in speech and language processing. CRC Press, 2003

Jelinek, Frederick. Statistical Methods for Speech Recognition. Cambridge, MA: The MIT Press, 1998

Jurafsky, Daniel; Martin, James H. Speech and Language Processing, An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. Upper Saddle River, New Jersey: Prentice Hall, 2000

Manning, Christopher D.; Schütze, Hinrich. Foundations of Statistical Natural Language Processing. Cambridge, MA: The MIT Press, 1999

Martinčić-Ipšić, Sanda. Raspoznavanje i sinteza hrvatskoga govora konteksno ovisnim skrivenim Markovljevim modelima, doktorska disertacija. Zagreb, FER, 2007

Milharčič, Grega; Žibert, Janez; Mihelič, France. Statistical Language Modeling of SiBN Broadcast News Text Corpus.//Proceedings of 5th Slovenian and 1st international Language Technologies Conference 2006/Erjavec, T.; Žganec Gros, J. (ed.). Ljubljana, Jožef Stefan Institute, 2006

Stolcke, Andreas. SRILM – An Extensible Language Modeling Toolkit.//Proceedings Intl. Conf. on Spoken Language Processing. Denver, 2002, vol.2, pp. 901-904