Statistical Language Modeling of SiBN Broadcast News Text Corpus

Grega Milharčič*, Janez Žibert†, France Mihelič†

*Department of Comparative and General Linguistics
Faculty of Arts
University of Ljubljana
Aškerčeva 2, SI-1000 Ljubljana
grega.milharcic@guest.arnes.si

†Laboratory of Artificial Perception, Systems and Cybernetics
Faculty of Electrical Engineering
University of Ljubljana
Tržaška 25, SI-1000 Ljubljana
{janez.zibert, france.mihelic}@fe.uni-lj.si

Abstract

The paper presents acquisition of the Slovenian broadcast news text corpus and its application in statistical language modeling. The problems encountered during the acquisition are described as well as constructing of language models for speech recognition purposes. Three different types of language models are built: a word-based 4-gram model, a class-based model with statistically-derived class maps and an interpolated model, combining the previous two. The comparison of language models is represented in terms of estimated perplexities on one third and one tenth of overall data used in evaluation experiments.

Statistično jezikovno modeliranje tekstovnega korpusa informativnih oddaj SiBN

V članku je predstavljena gradnja slovenskega besedilnega korpusa televizijskih informativnih oddaj SiBN in njegova uporaba pri statističnem jezikovnem modeliranju, namenjenemu samodejnemu razpoznavanju govora. Opisana je priprava in zbiranje besedil za gradnjo korpusa ter preizkusi statističnega jezikovnega modeliranja z uporabo teh podatkov. Zgrajeni so bili trije različni tipi jezikovnih modelov: besedni 4-gramski model, razredni model s statistično pridobljenimi razredi in interpolacijski model, ki je kombinacija prejšnjih dveh. Jezikovne modele smo primerjali med seboj na podlagi ocenjenih perpleksnosti, ki smo jih pridobili na testnem besedilu ene tretjine in ene desetine celotne zbirke.

1. Introduction

Statistical language models estimate the probabilities of word sequences which are usually derived form large collections of text material. Statistical language models can be applied in several tasks of language technologies (Manning & Schütze, 1999), including automatic speech recognition, optical character and handwriting recognition, machine translation, spelling correction. In our case, a text corpus of Slovenian broadcast news was acquired.

The corpus currently consists of text transcriptions of daily TV news shows provided by the national broadcast company RTV Slovenija for a period of one year. The collected text material corresponds to the spoken language transcriptions and is therefore suitable for building proper language models for automatic speech recognition.

Properly annotated and documented text data was used for the evaluation of different statistical language models for speech recognition. They will be applied in a system for large vocabulary continuous speech recognition for automatic transcription of Slovenian broadcast news, which is being developed at the Laboratory of Artificial Perception, Systems and Cybernetics at the University of Ljubljana.

We derived three different types of statistical language models which were based on n–gram statistics. This approach is a standard way of building language models for speech recognition. However, it should be noted that other approaches can be also successfully applied. For example, (Chelba, 2000) makes use of syntactic structure in language modeling, (Beutler et al., 2005) integrates a non-probabilistic grammar into large vocabulary continuous speech recognition, and (Zheng et al., 2005) defines an ontology-based language model. Other possibilities, which were also tried for Slovenian language, include specially designed techniques for modeling highly inflected languages (Sepesy Maučec et al., 2004) and topic-sensitive language modeling (Sepesy Maučec & Kačič, 2000).

The paper is organized as follows. The next section describes text corpus acquisition, followed by a section on applied language models. In the evaluation section the constructed language models are compared in terms of estimated perplexities in different experiments. We end with the conclusion and possible future work.

2. Text Corpus Acquisition

Text material for the corpus was collected from the subtitling transcriptions of the daily broadcast news (BN) shows transmitted live via teletext from a TV station RTV Slovenija. This service is used for providing a subtitling information of BN shows and is intended to help deaf and partially deaf people to follow the BN shows.

The corpus currently consists of the teletext subtitling transcriptions from December 2003 to December 2004 of daily BN shows Poročila, Dnevnik and Odmevi at 7.00 AM, 8.00 AM, 9.00 AM, 1.00 PM, 4.30 PM, 7.00 PM and 10.00 PM. The text material cover different BN news stories, different topics and different kind of information. The main part of the text data represents international news and news from Slovenia, but there are also cultural, financial and sport news, weather reports, traffic information, etc. The basic information of the corpus is given in Table 1.

number of BN shows	1 358
number of sentences	139 251
number of word tokens	2 295 664
number of word types	102 895

Table 1: Basic information of the SiBN text corpus.

As could be seen from the Table 1, the corpus is currently comprised from 2 million word tokens with vocabulary size of 100k different words. Thus, we could consider this corpus as a relatively small corpus with big amount of different words. This is typical for such kind of data representing broadcast news, where there exist many different word types belonging to different proper names, geographical and geopolitical names, technological and scientific terms, sport's results, etc.

The similar data can be captured also from newspapers or internet sites, which provide many more resources for building such text corpora, but there exist one important difference. The text material, which is acquired directly from subtitling transcripts of BN shows, is closer to spoken language, and therefore is more suitable for deriving language models for speech recognition purposes.

Another issue of capturing the text data via teletext is concerning the annotation and organization of the data into the corpus. In next sections we describe the acquisition of the data, text segmentation and annotation, and text normalization.

2.1. Text Segmentation

In the text acquisition process we dealt with two groups of problems. The first was connected with teletext data transmissions and the second with subtitling information. The major problems encountered in the acquisition process were:

- unwanted teletext marks, which did not belong to text transcripts;
- unwanted text and characters can appear anywhere in the texts due to the signal disturbance;
- punctuation marks . ! ? do not necessarily mean the end of the sentence (abbreviations, ordinal numbers, one-word exclamations, etc.), sometimes they are absent or appear randomly in the text, ? sometimes replaces other characters;
- punctuation marks : ; – also sometimes designate the end of the sentence;
- one of punctuation marks that otherwise appear in pairs, like ' ' " " ( ) may be absent;
- direct speech were designated in many different ways;
- sometimes sentences did not start with a capital;
- some sentences were interrupted, unfinished, repeated or started more than once;
- abbreviations: some were standard and common (oz., itd., itn., npr., g., ga., dr., C., &), other arbitrary and thus unpredictable;
- acronyms: with full stops or without, with all capitals or not, different spellings when declined;
- multipart words and proper names: with hyphens or not, written together or separately;
- numbers: cardinal, ordinal, decimal, fractions, percents, sport results, dates, character and product codes;
- Internet addresses and other strings;

In the text segmentation process we had to organize and annotate the subtitling data in a text corpus, which could be used for building a language models. Thus, we had to provide several automatic tools and also make some manual checking to transform the erroneous subtitling data in a way we needed.

An example of the raw text data, which was captured during data acquisition, is shown in Figure 1. The raw text data (in the top window) was constructed from several teletext subtitling records. Each record was represented by a time stamp of capturing and with transmitted text. The time stamps were additionally provided by our captioning tool, but they were not used in extracting text material from subtitling data, which was our main concern in a text segmentation process. As it can be seen from Figure 1, during the text segmentation process the teletext data from subtitling records shown in the top window should be transformed into the well–organized text showed in the bottom window.

In the example in Figure 1 the subtitling data posses several incorrectness or inconsistencies, which we had to change or remove. Due to the teletext service each subtitling record could be transmitted several times or there could also exist empty records. This is also presented in Figure 1, where there exist several unwanted teletext marks for the sentence that has started several times and marked without punctuation. There are also some other mistakes and inconsistencies: an ordinal number and a multipart word *36 letni*, which could also be written elsewhere as *36-letni* or *36letni*, etc. Some of these problems were solved automatically, other manually during consistency checking. Apostrophes and quotation marks were deleted, parentheses were deleted with their contents included. Common abbreviations were expanded to approximate the spoken form. All other symbols, which were not expected in the text data, were additionally analyzed and text was corrected accordingly.

All the transformed text material was also automatically spell–checked during the text segmentation process. The

Figure 1: An example of text acquisition and segmentation.

spell–checking was performed based on tool *Aspell*[1] to find possible typing errors, to replace letters *c, s, z* with *č, š, ž*, where appropriate, and, in some cases, to standardize words that can be spelled in different ways.

[1] http://aspell.sourceforge.net

In the last phase of the text segmentation process we had to manually check all the translated texts. In this phase we had to solve problems that could not be processed automatically, especially unwanted text from other broadcasts and punctuation issues. This phase was the most time consuming part of the acquisition process, but it was essential for an arrangement of the corpus in a consistent way.

2.2. Text Normalization

In order to use this corpus for developing statistical language models for speech recognition the text material had to be further normalized.

The text normalization process included the following tasks and procedures: all punctuation marks were deleted, capitals were transformed into lower case, the character encoding was standardized and beginnings and ends of sentences were labeled as '<s>' and '</s>', respectively.

2.3. Manually–annotated Word Classes

Additionally, we manually derived some word classes, which we used them latter in the process of building language models. The word classes were defined manually and annotated automatically in the text corpora.

There were two main reasons for defining such word classes. The first was, that we wanted additionally standardize the text material in places where one could expect inconsistent annotations. The second reason was the fact that some words could be equally likely used in the text and could be therefore better modeled with word classes. Hence, we decided to derive classes which belongs to two major groups of words: proper names and numbers. The manually–annotated classes are the following:

- acronyms: included strings of two or more capitals with an optional hyphen and ending; this class consisted of 969 different words;
- proper names: in this class belong non–first words, starting with a capital and with more than one letter; this class consisted of 29 843 different words;
- cardinal numbers: are strings of digits with an optional full stop in between, or strings of digits in the end of the sentence (1 140 different words);
- ordinal numbers: are strings of digits with a full stop in the end, which do not mark the end of a sentence, or strings of digits with hyphens and an ending not longer than 4 letters (longer most possibly mean not an ending, but a word of its own where hyphen was meant as a dash); in this class 1 300 different words were included;
- decimal numbers: are strings of digits with a comma in between (268 different words);
- fraction: are strings of digits with a slash in between (22 different words);
- percents: are string of digits, followed by a percent sign or a string 'odstot' plus ending or a string 'procent' plus ending; it was found 419 different words of such kind;

- sport results: are strings of digits with one or more colons in between (420 different words).

Altogether these word classes covered 34 381 different words, which represented 33 % of all different words in the corpus.

An automatic procedure for applying these word classes on the text data was designed. This procedure just replaces the words in classes with their corresponding class labels. In a such way we obtained two kinds of texts: one with classes and another without classes. For building of all language models in our experiments we used the text data, where word classes were annotated.

3. Statistical Language Models

The acquired text data, as described in the previous section, were used in the very first experiments of constructing different language models (LM). Three different types of language models were built: a word-based 4-gram language model, a class-based model with statistically-derived class maps and an interpolated language model, combining the previous two.

For each of these types two kind of evaluation experiments were performed. We divided the text data from the corpus into thirds and tenths. In the first group of experiments we built all three language models each time on different two thirds of overall data serving as the training text and one third was used as the test set. In the same manner the second group of experiments were performed on ten language models where each time different nine tenths of overall data served as the training text and one tenth as the testing text. In a such way we performed several experiments on different test and train data, which guaranteed us more objective evaluation of the proposed language models.

In each experiment we had to build a new language model from different training texts and test it on corresponding test data. The average statistics of the training and testing texts, which were used in experiments, are shown in Table 2. The average statistics based on the vocabularies of each datasets reveal the expected proportion of sentences and words according to the proportion of data in each group of experiments (2/3 train, 9/10 train).

		2/3 train	9/10 train
sentences	test	46 436	13 931
	train	92 871	125 376
word tokens	test	766 488	229 947
	train	1 504 506	2 038 571
word types	test	51 019	28 275
	train	40 314	46 722
OOV rate	test	2.99%	2.44%
	train	1.75%	1.41%

Table 2: The average statistics of data using in test and train set texts.

Due to the different training data in each experiment, the special attention was needed to model out–of–vocabulary (OOV) words. The basic rule here was to drop out the words with frequency of 1 in all training texts and map them to the unknown word class. This class was then used for modeling OOV words in different types of language models. Average OOV rates for different train and test datasets are also shown in Table 2. All other words, which were not marked as OOV words, were then used as a basic vocabulary in the training of the language models.

All tested language models were built in a standard way using the HTK Toolkit (Young et al., 2005). So, in the following subsections the main ideas and approaches in constructing of all three language models will be presented.

3.1. Word-based Language Models

We built a standard word–based 4-gram language model, which was used as a baseline language model in all of our experiments.

The main idea in n–gram language modeling is, that the probabilities of word sequences are estimated based on frequencies of words and word sequences obtained from training text material. In a word 4-gram language model the probabilities of words sequences are approximated from the conditional probabilities based on sequences of last 4 words:

$$P(w_1, \ldots, w_n) \simeq \prod_{i=1}^{n} P(w_i \mid w_{i-3}, w_{i-2}, w_{i-1}). \quad (1)$$

In our case the probabilities of 4-gram language models were estimated using Good-Turing discounting with Katz back-off smoothing (Katz, 1987). This is a standard procedure in building a n–gram language model with HTK Toolkit (Young et al., 2005).

3.2. Class-based Language Models

Class–based language models work in the same manner than word–based models, but, instead of words, word classes are derived and used for estimation of probabilities. The probabilities of word sequences can be therefore estimated in the case of 4-gram as

$$P(w_1, \ldots, w_n) \simeq \prod_{i=1}^{n} P(w_i \mid G(w_i)) \times P(G(w_i) \mid G(w_{i-1}), G(w_{i-2}, G(w_{i-3})), \quad (2)$$

where w_i represents words and $G(w)$ word classes. The probability of word sequences $w_1, \ldots, w_n$ is here approximated by a product of conditional probability that a word w_i belong to a word class $G(w_i)$ and a conditional probability that a word class $G(w_i)$ followed a sequence of word classes $G(w_{i-1}), G(w_{i-2}), G(w_{i-3})$. The estimated probability is therefore a generalization of the form in equation (1), where probabilities are estimated from word–based sequences.

The main issue here is how to obtain word classes. General idea is that we map words with similar syntactic or semantic behavior into the same class or category. A member word of such class is considered as equally likely to appear in contexts of any other member of the same class. Classes can be obtained manually or statistically.

In our case we derived classes statistically. For deriving word classes from the training texts of the corpus a word

exchange algorithm (Young et al., 2005) was applied. The procedure is shown in Figure 2.

1. **Initialize**: $\forall w \in \mathbb{W} : G(w) = 1$
   Set up the class map so that all words are in the first class and all other classes are empty (*or* initialize using some other scheme)
2. **Iterate**: $\forall i \in \{1, \ldots, n\} \wedge \neg s$
   For a given number of iterations $1 \ldots n$ or until some stop criterion s is fulfilled

   Iterate: $\forall w \in \mathbb{W}$
   For each word w

   (a) **Iterate**: $\forall c \in \mathbb{G}$
   For each class c

   i. **Move** word w to class c, remembering its previous class

   ii. **Calculate** the change in F_{M_C} for this move (F_{M_C} is defined below)

   iii. **Move** word w back to its previous class

   (b) **Move** word w to the class which increased F_{M_C} by the most, or do not move it if no move increased F_{M_C}

$$F_{M_C} = \sum_{g,h \in \mathbb{G}} C(g,h) \cdot \log C(g,h) - 2 \sum_{g \in \mathbb{G}} C(g) \cdot \log C(g)$$

where (g, h) is a class sequence $'hg'$ and $C(g)$ is the number of times that the class g occurs in the training text list of words

Figure 2: The clustering procedure for obtaining class maps.

This clustering algorithm optimizes the classification function $G(w) = g$, which maps a word w from the set of all words $\mathbb{W}$ to a word class g from the set of all classes $\mathbb{G}$. The map is set on the basis of the differences in entropy of the bigram and unigram word class probabilities, defined with function F_{M_C}. The basic procedure is following. In every iteration each word is mapped into that class (the class maps are deterministic – each word can only be in one class), which increase the overall entropy in function F_{M_C}. This procedure actually performs an exhaustive searching of optimum mapping between words and word classes. Due to computational complexity of the algorithm one should carefully set the actual number of classes and the number of iterations for finding the optimal class maps. One should choose the number of classes significantly lower than there are words in a training-data vocabulary, otherwise the class–based language model approximates to the word–based model. Our language models were built using 2 iterations and 600 classes, as it was recommended for our vocabulary size in (Young et al., 2005).

Word classes reduce the number of parameters and give more reliable estimations of rare words, which is useful in situations, where train and test conditions do not match. Also note, that one should expect higher perplexities, when using such models on training data, in comparison to word–based models. This can be explained by the fact that word–based models better estimate the expected word sequences in training text, while class–based models generalize the estimation of the word sequences with word classes.

3.3. Interpolated Language Models

Interpolated language models are generated by combining word–based and class–based language models. There exist number of combining techniques how to join two language models together. In our case, word–based and class–based language models were joined together by linear interpolation (Young et al., 2005).

We had to additionally set the interpolation weights for both models in a way to favor one model against the other. The interpolation weights were chosen to maximize the overall perplexity of an interpolated model on test data. The results are shown in Figure 3.

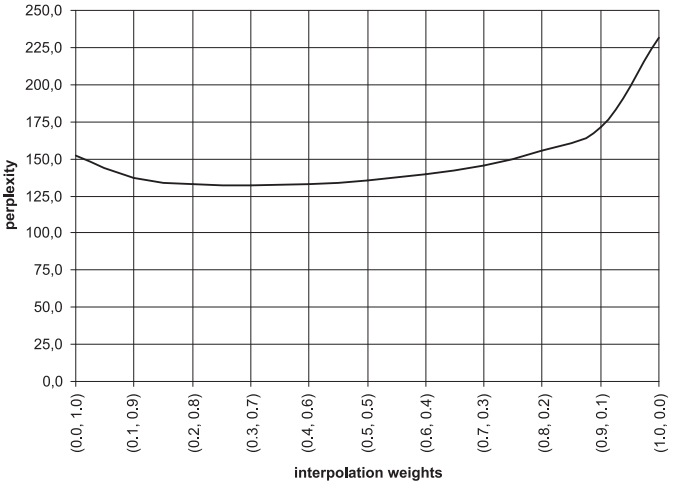

Figure 3: Setting interpolation weights for class–based and word–based language models in joined language models on test data. The first weight in a pair of weights in round brackets corresponds to class–based language model, the second weight belongs to word–based language model.

We tested different combinations of weights in interval from 0 to 1. As can be seen from the results in Figure 3, we obtained the best perplexity by applying linear interpolation of both models with interpolation weights of 0.7 and 0.3 for word–based and class–based models respectively.

4. Evaluation experiments

The perplexities of language models were estimated. In Table 3, the average perplexities of the three language models trained on two thirds of overall data and the ten language models trained on nine tenths of overall data for each type of language models built for 2-gram, 3-gram and 4-gram, evaluated on both testing and training texts parts, are given.

		word-based LM		class-based LM		interpolated LM	
		test	train	test	train	test	train
2-gram	2/3 train	233.6	84.5	322.7	272.3	213.2	93.1
	9/10 train	228.7	87.7	324.7	283.9	209.0	96.7
3-gram	2/3 train	171.2	20.0	255.4	117.5	150.7	23.0
	9/10 train	161.3	19.9	251.3	125.5	142.0	22.9
4-gram	2/3 train	163.8	13.1	239.7	59.3	142.8	14.7
	9/10 train	151.9	12.2	232.0	60.4	132.3	13.7

Table 3: The average perplexities of tested language models.

Results on the particular test an training text partition did not differ significantly between each other for the two thirds – one third and nine tenths – one tenth text partitions. Big differences between the estimated perplexities acquired from the training and test part of the corpus could be noticed. However estimated perplexities from the test parts are more descriptive. Interpolated n–grams models consistently gives better lower estimated perplexities results on the test sets compared to the corresponding n–grams word models. We also achieved as expected better results in the nine tenth train – one tenth test evaluation scenario. In this case larger amount of training data and smaller test part with consequently smaller OOV word rate was used (Table 2). As it was expected, the best model appeared to be the interpolated 4–gram model, trained on nine tenths of the overall data.

The results cannot be directly compared to the other recently reported results for Slovenian language since different text corpora with different vocabularies were used. However, a novel method for highly inflected languages used in language modeling of Slovenian should be mentioned: (Sepesy Maučec et al., 2004) reports of perplexity improvement from 360 to 248 and OOV rate improvement from 6.03% to 0.97% when cutting off the grammatical information from words in the so-called stem-ending language model, trained on a 59M-word corpus of newspaper Večer with vocabulary size of 64 000.

5. Conclusion

The acquisition of the Slovenian broadcast news text corpus was described. Using this corpus, three different types of statistical language models were built. We applied two different evaluation scenarios using two thirds and nine tenths of the corpus for training purposes. The language model with the lowest estimated perplexity on the test set was the interpolated 4-gram model, trained on nine tenths of overall data.

The statistical language models will be applied for automatic transcription of Slovenian broadcast news as one of the components of large vocabulary continuous speech recognizer. Future work might also include acquisition of a larger corpus, different extended annotations and application and experiments with other different approaches for language modeling.

6. Acknowledgement

The authors would like to thank the public broadcasting company RTV Slovenija for their permission to freely use the broadcast news data for scientific purposes.

7. References

Beutler, R., Kaufmann, T., & Pfister, B. (2005). Integrating a non-probabilistic grammar into large vocabulary continuous speech recognition. In *Proceedings of the IEEE ASRU 2005 Workshop*, pp. 104–109, San Juan, PR. IEEE.

Chelba, C. (2000). *Exploiting Syntactic Structure for Natural Language Modeling*. PhD thesis, Johns Hopkins University, Baltimore, MD.

Katz, S. M. (1987). Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 35(3):400–401.

Manning, C. D. & Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*. The MIT Press, Cambridge, MA.

Sepesy Maučec, M. & Kačič, Z. (2000). Topic-sensitive language modelling. In *Proceedings of the Third International Workshop on Text, Speech and Dialogue*, pp. 235–258, Brno. Springer-Verlag.

Sepesy Maučec, M., Kačič, Z., & Horvat, B. (2004). Modelling highly inflected languages. *Information Sciences*, 166:249–269.

Young, S., Evermann, G., Gales, M., Hain, T., Kershaw, D., Moore, G., Odell, J., Ollason, D., Povey, D., Valtchev, V., & Woodland, P. (2005). *The HTK Book*. Cambridge University Engineering Department, Cambridge.

Zheng, D., Yu, H., Zhao, T., Li, S., & Peng, Y. (2005). Research on an ontology-based language model. In *Proceedings of the International Conference on Chinese Computing 2005*, Singapore. COLIPS.