Detecting Commas in Slovak Legal Texts

Róbert Sabo[1] and Štefan Beňuš[1,2]

[1]Institute of Informatics of Slovak Academy of Sciences
[2]Constantine the Philosopher University in Nitra
robert.sabo@savba.sk, sbenus@ukf.sk

Abstract. This paper reports on initial experiments with automatic comma recovery in legal texts. In deciding whether to insert a comma or not, we propose to use the value of the probability of a bigram of two words without a comma and a trigram of the words with the comma. The probability is determined by the language model trained on sentences with commas labeled as separate words. In the training database one sentence corresponds to one line. The thresholds of bigrams and trigrams probability were experimentally determined to achieve the best balance of precision and recall. The advantage of the proposed method is its high precision (95%) at a relatively satisfactory recall (49%). For judges as potential users of an ASR system with an automatic comma insertion function, precision is particularly important.

Keywords: automatic speech recognition; Slavic languages; judicial domain

1 Introduction

With the advance of the automatic speech recognition technology, ASR systems began to be deployed in many varied areas. One such area in Slovakia is the judicial domain. Currently, judges can dictate legal texts which are transcribed by an ASR engine with the accuracy of about 95% [1]. In this state of the art the overall quality of transcribed texts is more and more important. Correct text formatting or punctuation plays an important role in the everyday use of ASR systems.

In studies focused on punctuation recovering, authors typically detect commas and sentence ends [2,3]. Within the sentence ends some authors further distinguish between periods, question marks and possibly also exclamation marks [4,5]. In this paper we concentrate only on comma recovering. Our aim is to offer a text which should corresponds to the desired text as much as possible. The presence of the correct punctuation in texts generated by the ASR system deployed for dictating legal documents should shorten the time necessary for final corrections by the user.

Most previous punctuation prediction techniques, developed mostly by the speech processing community, exploit both lexical and prosodic cues. There is comparatively little work exploiting lexical features exclusively [6,7,8,9]. In this paper, we tackle the task of predicting punctuation symbols from a standard text processing perspective without relying on additional prosodic features such as pitch and pause duration.

The proposed method of commas recovering is trained and tested on the Slovak language. Slovak is similar to other Slavic languages in having a rich inflectional and derivational morphology which causes additional problems with language modeling

P. Sojka et al. (Eds.): TSD 2014, LNAI 8655, pp. 62–67, 2014.
This is preprint prepared by Proceedings editor for Springer International Publishing Switzerland.

such as the need for much larger vocabulary or greater difficulties in part-of-speech tagging. Slovak also has a relatively free word order, which degrades the performance of N-gram language models. The rules for writing commas in Slovak are stricter than in English. Commas separate:

- subordinate clauses from main clauses;
- all co-ordinate constituents unless they are connected by copulative conjunctions *a, i, alebo* (eng. ‘and’, ‘as well’, ‘or’);
- all independent constituents that are inserted into a sentence (parentheses, complements, explanations, etc.).

The rest of this paper is organized as follows. Section 2 briefly introduces the text corpus. Section 3 presents the evaluation metrics that were employed. Section 4 describes the proposed method of comma detection. In Section 5, we report experimental results and Section 6 presents our conclusions and future work.

2 Text Database

To train the language model we used a text corpus of legal text consisting of 17M word tokens. In this corpus 1,035k words were followed by a comma and 437k by sentence ends.

To minimize the perplexity of the trained language model we introduced named entities of three categories. We replaced the numbers by tags <num> and proper names (words beginning with a capital letter) by tags <pn>. Some words in the original legal texts, such as the names of defendants, witnesses, and companies, were anonymised by court employees for security reasons. These anonymised words were replaced by special tag <anon>. Names of some persons e.g. minutes clerk, judges and people who were not parties to court proceedings were omitted from the anonymisation process. These words, as well as all other words beginning with a capital letter, were replaced by tag <pn>.

Commas were separated by spaces and replaced by tag <com>. Sentences were divided into separate lines as it is required for n-gram training by SRILM toolkit [10]. The database was divided into the training and testing sets. The testing database comprises about 1 percent of the training set.

3 Evaluation Metrics

Precision, recall and F-measure are well known performance measures widely used in punctuation tasks. In our task, precision expresses the percentage of correctly inserted commas of all commas inserted by the system. It is important for this number to be as high as possible because users (the judges) prefer a lower number of correctly inserted commas over a higher number of wrongly inserted ones.

The precision and recall are defined in (1) and (2) below.

$$P = \frac{C}{C + F} \tag{1}$$

$$R = \frac{C}{C + M} \tag{2}$$

In these equations C is the number of correctly inserted commas, F is the number of falsely inserted commas and M is the number of missing commas. To express the system performance by a single number, it is possible to use a harmonic mean of P and R called the F-measure (3).

$$F = \frac{2PR}{P + R} \tag{3}$$

4 Proposed Method

Our proposed method is based on comparing the probability of bigrams and trigrams obtained from the language model trained with the SRILM toolkit [10] and saved in the ARPA format. In the language model, for each line the logarithm (base 10) of conditional probability of each N-gram [11] is given; in this work it is labeled as p2 for bigrams and p3 for trigrams. The training database was formatted as a file containing one sentence per line.

In the first step we find the probability p3 of the trigrams (a pair of words separated by a comma). If the trigram (word1 <com> word2) is more probable than a defined threshold, the second step follows. In this step, the probability p2 of the bigram of these words without a comma (word1 word2) is evaluated. If the bigram (word1 word2) is less probable than a selected threshold, the script places a comma between word1 and word2. The thresholds have been examined for different combinations of p3 and p2 values shown in Figures 1 and 2 and discussed in the following section.

5 Experimental Results

The testing database size was about 1 percent of the training set and contained 14,7k words. The quality of the testing database is affected by the fact that the original text material contains a number of errors and ambiguities in comma inserting, which significantly influenced the resulting precision and recall values. The results were degraded by 1–2% due to these errors. The most frequent error was the absence of a comma. The ambiguities typical for the Slovak language involve commas in front of conjunctions *a, i, aj, ani, alebo, či* in graduative, adversative or disjunctive semantic relationships between the connected clauses. In these cases commas preceding the conjunctions are optional. The best results for precision, recall and the F-measure are shown in Table 1. The table shows two different settings of probability thresholds *p3* for trigrams and *p2* for bigrams. The first line shows the best possible precision and the second line the best possible recall.

Figures 1 and 2 illustrate precision and recall values for different thresholds for bigram (p2) and trigram (p3) probabilities. The precision values are consistently high when $p3 > -2.2$ and $p2 < -2.9$. In contrast, recall achieves the best results by $p3$ between -1.8 and -2.4 and by $p2$ between -2.9 and -3.3.

Table 1. Precisions, recalls and F-measures for detecting commas using different models.

	Precision	**Recall**	**F-measure**
Best precision	97.33	44.12	60.72
Best recall	95.31	49.64	65.28

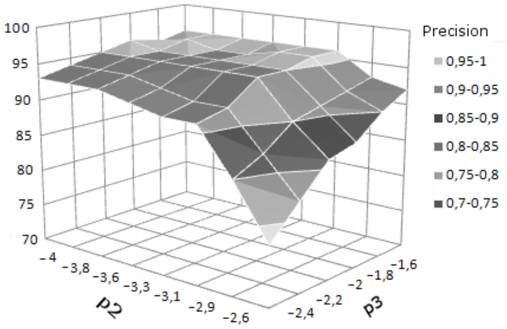

Fig. 1. Precision values (vertical axis) for different thresholds for bigram (p2) and trigram (p3) probabilities.

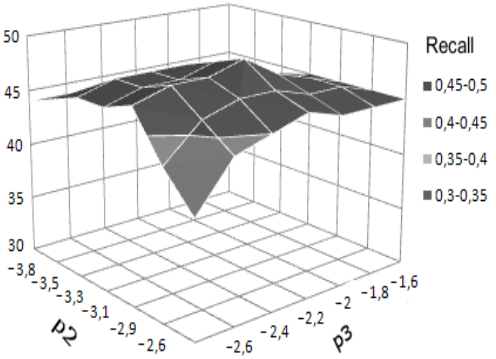

Fig. 2. Recall values (vertical axis) for different thresholds for bigram (p2) and trigram (p3) probabilities.

The best precision is achieved when $p3 = -2$ and $p2 = -3.1$. The best recall is achieved by $p3 = -2.2$ and $p2 = -2.9$. We obtained the best F-measure with $p3 = -2.2$ and $p2 = -2.9$.

We also performed similar tests on the test database without the information about sentence ends. The results are shown in Table 2.

Table 2. Precisions, recalls and F-measures for detecting commas using different models [%] without information on sentence ends.

	Precision	Recall	F-measure
Best precision	96.75	42.84	59.38
Best recall	95.14	47.64	63.48

The minimal degradation of precision and recall allows the proposed method to be applied in the tasks in which the sentence ends are not known.

The results are comparable to a similar system for the closely related Czech language [2]. The main contribution of our approach is a high precision and a satisfactory recall of comma detection.

The advantage of our domain is that legal texts contain specific vocabulary and stylistic features. In comparison with broadcast news texts, legal texts contain many repetitive words that occur in similar collocations, which facilitates training the language model with low perplexity.

6 Conclusions and Future Work

Results for detecting commas in Slovak legal texts were presented. In deciding whether to insert a comma or not we follow the value of the probability of a bigram of two words without a comma and a trigram of the words with a comma. We perform two tests. First, we test on the database with information on sentence ends and then without information on sentence ends. Both tests of the presented method show high precision (approximately 95%). Recall value was higher when testing the database with information on sentence ends (49%) than with the database without sentence ends (47%).

These results are comparable with a similar system for the Czech language [2] that is a highly inflectional and derivational language similar to Slovak. The use of acoustic features such as prosody or pauses could improve the results. It is plausible that implementing part of speech tagging would also improve the results of comma detection.

Our paper approached the problem of comma detection only from a standard text processing perspective without relying on additional prosodic features such as pitch and pause duration. We take it as a first, but very important, step towards solving the comma detection problem in Slovak. We expect that including prosodic cues may improve the results but at the same time, their influence might not be very significant in this domain, which is supported also in other studies [2,4].

Acknowledgement. This publication is the result of the project implementation: Technology research for the management of business processes in heterogeneous distributed systems in real time with the support of multimodal communication, ITMS 26240220064 supported by the Research & Development Operational Program funded by the ERDF.

References

1. Rusko, M., Juhár, J., Trnka, M., Staš, J., Darjaa, S. et al.: Slovak automatic transcription and dictation system for the judicial domain. In: Human Language Technologies as a Challenge for Computer Science and Linguistics: 5th Language & Technology Conference. pp. 365–369. Fundacja Uniwersytetu Im. A. Miczkiewicza, Poznań (2011)
2. Kolář, J., Švec, J., Psutka, J.: Automatic punctuation annotation in Czech broadcast news speech. SPECOM´2004, pp. 319–325. Saint-Petersburg (2004)
3. Batista, F., Caseiro, D., Mamede, N., Trancoso, I.: Recovering capitalization and punctuation marks for automatic speech recognition: Case study for the Portuguese broadcast news. Speech Communication, vol. 50, no. 10, 847–862 (2008)
4. Huang, J., Zweig, G.: Maximum entropy model for punctuation annotation from speech. In: Proceedings of International Conference on Spoken Language Processing, pp. 917–920. Denver (2002)
5. Christensen, H., Gotoh, Y., Renals, S.,. Punctuation annotation using statistical prosody models. In: Proc. ISCA Workshop on Prosody in Speech Recognition and Understanding, pp. 35–40. (2001)
6. Wei, L. and Hwee, T. N.: Better punctuation prediction with dynamic conditional random fields. In: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, pp. 177–186. Cambridge (2010)
7. Gravano, A., Jansche, M., Bacchiani, M.: Restoring punctuation and capitalization in transcribed speech. In: Proceedings of The International Conference on Acoustics, Speech, and Signal Processing, pp. 4741–4744. Dallas (2009)
8. Stolcke, A., Shriberg, E., Bates, R., Ostendorf, M., Hakkani, D., Plauche, M., Tur, G., and Lu, Y. Automatic detection of sentence boundaries and disfluencies based on recognized words. In: Proc. of ICSLP 1998. (1998)
9. Jakubíček, M., Horák, A.: Punctuation Detection with Full Syntactic Parsing. In: Research in Computing Science, Special issue: Natural Language Processing and its Applications, vol. 46, pp. 335–343. Mexiko: Instituto Politécnico Nacional (2010)
10. Stolcke, A.: SRILM – An Extensible Language Modeling Toolkit. In: Proc. of ICSLP 2002, pp. 901–904. Denver (2002)
11. `http://www.speech.sri.com/projects/srilm/manpages/ngram-format.5.html`