The Slovak Categorized News Corpus

Daniel Hladek, Jan Stas, Jozef Juhar
Department of Electronics and Multimedia Communications
Technical University of Kosice, Slovak Republic
daniel.hladek@tuke.sk, jan.stas@tuke.sk, jozef.juhar@tuke.sk

Abstract

The presented corpus aims to be the first attempt to create a representative sample of the contemporary Slovak language from various domains with easy searching and automated processing. This first version of the corpus contains words and automatic morphological and named entity annotations and transcriptions of abbreviations and numerals. Integral part of the proposed paper is a word boundary and sentence boundary detection algorithm that utilizes characteristic features of the language.

Keywords: categorized corpus, Slovak language, annotation

1. The State of the Art

There are only few resources for studying the Slovak language. None of them are fully publicly available. The only existing formal research corpora are maintained by the Slovak National Corpus organization[1], but the corpora are publicly available only as a limited full-text search interface.

A set of informal corpora and processing tools have been prepared in our previous research. As a part of the Slovak Judicial Domain Dictation System applied research task (Rusko et al., 2011), a web-crawling, gathering and processing agent has been proposed (Hládek and Staš, 2010). Similar attempt was performed for Czech language in (Spoustová and Spousta, 2012).

The approximate size of the gathered and processed data in our previous system was 2 billions of tokens and enabled us to create a judicial-domain language model for a dictation application. The dictation system has been submitted and proven as useful, but there is still room for improvement in the language model part – detection of the named entities, adaptation of the language model for specific domain and utilization of the morphological features in the language model.

The biggest problem of the previously used data sets was their informal characteristics. The amount of data and its representation has been changing, number of crawled web pages increasing and the processing tools evolving. The gathered data were only very roughly sorted into domains. In order to overcome these problems a well-defined, processed and sorted set of textual data have to be constructed (as it is in (Brown et al., 1992)).

2. Method for the Corpus Construction

The proposed corpus utilizes previously developed tools for text gathering and processing. However, in this case the focus is not given on quantity, but on the quality of the gathered data.

A specialized web agent is used to explore the newspaper web site similar to the one presented in (Hládek and Staš, 2010). Content of each web-page is analyzed and saved in a database. Collected items are parsed and given to the appropriate form. The data processing step should preserve as much information as it is possible. Only those parts that are not interesting for the intended use of the corpus are removed.

[1] http://korpus.juls.savba.sk/index_en.html

3. Word and Sentence Boundary Detection

The main goal is to distinguish between types of tokens that are interesting for further processing by adding and removing spaces and unnecessary characters as it is required. The following types of tokens are recognized:

- words and acronyms,
- abbreviations,
- various number representations,
- URLs and e-mails,
- punctuation.

The regular expressions in the proposed tokenizer try to identify tokens that should be preserved, tokens that should be joined by removing spaces from the input stream of characters. After these tokens are correctly found in the input text, it is possible to identify the end of the sentence by the punctuation used.

Special attention should be paid to the tokens that contain a dot. The dot is probably the most ambiguous character in the Slovak language – it can be part of abbreviation, part of order numeral or can mean end of the sentence. Luckily, the Slovak convention is to use comma as a floating point mark.

Rules for recognition of words, acronyms, URLs, emails and punctuation are language independent. On the other hand, rules for recognition of abbreviations and numbers depend on language-specific context. After adapting these rules, the proposed tokenization algorithm is usable for another language.

3.1. State Machine Implementation

The problem of the existing rule-based approaches for tokenization and sentence boundary detection is their slow speed and high complexity. Classical regular expressions are very hard to read and complicated to find potential errors.

In order to overcome these problems, each proposed rule of the formal grammar is compiled into a single state-machine, using a specialized parser generator Ragel (Thurston, 2006). Rules in the Ragel language are better readable than rules in classical regular expression engines and the whole system is later translated to a single state machine in the C language.
The result is a word and sentence boundary detector, written in C that has no external dependencies and is usable as a library in other systems. Similar rule-based system for lexical and morphological analysis has been developed for Croatian language in (Ćavar et al., 2009).

3.2. Tokenization Algorithm

Input of the algorithm is:

1. list of regular expressions describing recognized tokens,
2. list of words that should be written using capital word in the beginning of the sentence,
3. list of abbreviations,
4. unprocessed text.

The tokenization and sentence boundary detection algorithm can be described as follows:

1. List of recognized tokens is searched. The longest matching token is selected.
2. If recognized token is a dot, colon, empty line, exclamation mark or question mark, the end of sentence is found.
3. If no token is found, the first character is discarder and the search process continues.
4. If some other token is found, it is added to the sentence, characters are discarded from the input and the search process continues. If the token is the first in the sentence and it is not in the list of exceptions then it is lowercased.
5. If there are no more characters in the input string, the search process finishes.

Output of the algorithm is a list of the recognized tokens and sentences. The language-dependent parts of the rule-base of the tokenizer are list of abbreviations, list of uppercase words and a list of helper words for recognition of dates.

4. Token Annotations

The corpus is intended for use in several natural language processing tasks. After identification of words and sentences of the text, each identified token has annotations assigned. Token annotations are stored together with tokens in the document files. Example of annotated sentences is in the table 4. Thanks to the very simple format of annotations it is possible to parse document file easily and utilize only meta-information that is useful for the given task.

entity	tag
Integer numbers	<INT>
Floating point numbers	<FLOAT>
Names of months	<MONTH>
Male Names	<MM>
Male Surnames	<MP>
Female Names	<ZM>
Female Surnames	<ZP>
Slovak cities and villages	<OBEC>
Slovak street names	<ULICA>
Organization names	<ORG>
Names of countries	<COUNTRY>
Other geographical locations	<LOCATION>
Stop-words	<STOP>

Table 1: Recognized named entities

word	ukrajinským	športovcom
tag	AAms7x	SSms7
meaning	Ukrainian	sportsmen
Part-Of-Speech	AA - adjective	SS - substantive
Person	m - male	m - male
Number	s - singular	s - singular
Case	7 - instrumental	7 - instrumental

Table 2: Example of morphological tags

The most important part of the annotation is morphological tagging. Other information, such as named entity tags and transcriptions to the spoken form might be useful for better understanding of the token function in the document.

4.1. Named Entity Recognition

The named entity recognition uses a very simple rule-based approach. Each recognized named entity is recognized using a certain rule that can be a regular expression or a dictionary item. Named entities that are covered by dictionaries or rules are in the tab. 1.
The Slovak language has very similar features to the Czech and the future research in the named entity recognition will be focused on a statistical approach as it is in (Straková et al., 2013; Konkol and Konopík, 2013).

4.2. Part of Speech Tagging

The most important part of the annotation process is the morphological annotator Dagger (Hládek et al., 2012). This classifier uses second-order hidden Markov model and Viterbi algorithm and can utilize grammatical features for smoothing of the observation and transition matrix for improvement classification accuracy.
The model has been trained on trigram counts from the Slovak National Corpus (Horák et al., 2004) and uses their tag set containing 3500 distinct tags. The search space of the classifier is restricted by a lexicon that contains a list of possible tags for each known word. Observation probabilities are smoothed using custom algorithm that takes morphological features of words into account. The classifier is 86% correct (Hládek et al., 2012). Manual correction of the

token	transcription	meaning
`napr.`	napríklad	for example
`3.`	tretie	third
`miesto`	not applicable	place

Table 3: Example of token transcription

annotation will follow later. Example of possible morphological tags are in the table 2.
Detailed documentation for all possible tags and flags is available in the pages of the Slovak National Corpus[2].

4.3. Lemmatization

The classification system is almost the same as in the case of morphological tags - second-order HMM model smoothed using additional information about word suffices, presented in (Hládek et al., 2012). A similar approach to lemmatization for Hungarian language is described in (Orosz and Novák, 2013). Each token in the corpus has assigned the most probable lemma. In the case when it was not possible to assign a lemma, a not-available sign is used (%).

4.4. Transcription to a Spoken Form

If it is applicable, a transcription of a named entity to the spoken form is provided. Again, it is performed using a rule-based system that is capable of utilizing morphological information to transcribe token with a correct grammatical form. Example of transcription of some named entities is in the table 3.
Transcription to a verbal form is helpful for normalization of the token meaning and for training a language model. More information about the problem can be found in the work by (Sak et al., 2013).

5. Format of the Corpus

Contents of the corpus are stored in a form that should be easy to process using any common programming language or tool. There are many possible ways to do so and none of them cannot work for all use-cases.
Probably the most popular form of organization of structured data is XML as it is in (Böhmová et al., 2003). Its biggest advantage is a precise description of the stored information and relatively easy parsing. On the other hand, it is also very verbose, requires working with external parsing library and it is unable to read improperly formatted data.
From this reason our own way of data organization has been designed. It has more informal structure of data that can be processed using only very basic tools and can be easily viewed and queried using only shell commands or common text editor. It does not require working with additional library nor any knowledge about XML format.
The starting point for processing text files is a search index (example is in table 5). The index is a separate file where meta information about document is stored. Each line in the index file contains following information, separated by a tabulator:

- Name of the file, including name of the domain,
- title of the document,
- date of publication,
- author of the document.

The index file allows very easy selection of documents, using just standard tool as Grep or simple Perl script. It is possible to select documents from specific authors or search documents for a specific keyword using just regular expressions.
The first column of the index file points to a contents of the document, stored in a certain folder. Each document is given an unique name. Folders are divided according to the sections as it is in the table 6.
Contents of the file with identified tokens and sentences is a text that can be used as an input for natural language processing system. The representation format is chosen to be as simple as it is possible – one sentence per line. Example of annotated sentences is in the table 4.
Each token in the sentence is separated by a space. Special characters are used to include annotation for each token in the sentence. One token can have more annotations, separated by horizontal line `|`. Spaces in the annotations are replaced by an underscore _. Missing or not applicable annotation is marked by `%` mark.

Label	# tokens	# sentences	# documents
Politics	472 305	32 258	1 655
Sport	339 791	24 699	1 042
Culture	13 373	757	22
Economy	376 036	22 893	1 022
Health	231 275	13 067	560
World	149 042	8 388	481
Together	1 581 822	102 062	4 782

Table 6: Contents of the Corpus

6. Contents of the Corpus

All articles are written by a professional reporters and should fulfill a characteristic style of the particular newspaper. The texts should cover the ”correct” Slovak grammar with proper expressions and vocabulary. On the other hand, the newspaper articles in this corpus does not cover the Slovak language in general. Colloquial forms of the language are present only as parts of interviews with people. Improper, vulgar or expressive language is committed at all. Artistic form of the language is not present.
The size of the corpus has been chosen to be as small as it is possible. The reason is that it should be possible to check the text manually. One and half million of tokens seems to be a feasible amount. Contents of the corpus is summarized in the table 6.

[2]http://korpus.juls.savba.sk/attachments/morpho/tagset-www.pdf

autori|%|SSmp1|% poukazujú|%|VKepc+|% na|%|Eu4|% to|%PFns4|% ,|<PUN>|Z|čiarka že|%|O|% britský|%|%|% peňažný|%|AAis1x|% účet|%|SSis1|% bol|%|VLescm+|% naposledy|%|T|% v|%|Eu6|% prebytku|%|SSis6|% v|%|Eu6|% roku|%|SSis6|% 1983|<INT>|NUms1|tisíc_deväťsto_osemdesiat_tri .|<PUN>|Z|bodka+
odvtedy|%|PD|% si|%|R|% krajina|%|SSfs1|% musí|%|VKesc+|% požičiavať|%|VIe+|% od|%|Eu2|% Číny|<COUNTRY>|SSfs2|% a|%|O|% ďalších|%|AAfp2x|% a|%|O|% predáva|%|VKesc+|% majetok|%|SSis4|% cudzincom|%|SSmp3|% .|<PUN>|Z|bodka

Table 4: Preview of the corpus file, tokens are separated by a space, sentences are separated by a new line.

File Name	Document Title	Publication Date	Author
corpus/domova/000962txt	Mišenku, Mella a Borka zbavili obvinení v kauze lúpeže	17. 8. 2013	sme
corpus/domova/000963txt	Tretí lekár nesúhlasí s posudkom v kauze Malinová	11. 9. 2009	Monika Tódová
corpus/domova/000964txt	Nástupom Fica je víťaz prvého kola jasný	18. 12. 2013	Matúš Burčík

Table 5: Preview of the index file, columns are separated by a tab.

7. Conclusion

The presented corpus of the Slovak newspaper articles should represent a contemporary formal language. Its main purpose is to serve as a testbed for several language processing tasks, such as:

- Automatic excerpt extraction,
- language model evaluation,
- document categorization,
- language model adaptation,
- linguistic research.

As there are only a few Slovak language resources available, it also should serve as a foundation for further development. A constrained and well-defined set of textual data is the first step before creation of manually annotated corpus. In the close future, the corpus should be subject of manual correction of the morphological forms and the named entities.

8. Acknowledgements

The research presented in this paper was supported by Research and Development Operational Program funded by the ERDF under the project numbers ITMS-26220220182 (50%) and ITMS-26220220141 (50%).

9. References

Böhmová, A., Hajič, J., Hajičová, E., and Hladká, B. (2003). The Prague dependency treebank. In *Treebanks*, pages 103–127. Springer.

Brown, P. F., Pietra, V. J. D., Mercer, R. L., Pietra, S. A. D., and Lai, J. C. (1992). An estimate of an upper bound for the entropy of English. *Computational Linguistics*, 18(1):31–40.

Ćavar, D., Jazbec, I.-P., and Stojanov, T. (2009). Cromomorphological analysis for standard Croatian and its synchronic and diachronic dialects and variants. *Finite-State Methods and Natural Language Processing. Frontiers in Artificial Intelligence and Applications*, 19:183–190.

Hládek, D. and Staš, J. (2010). Text gathering and processing agent for language modeling corpus. *Proceedings of the 12th International Conference on Research in Telecommunication Technologies, RTT*, pages 200–203.

Hládek, D., Staš, J., and Juhár, J. (2012). Dagger: The Slovak morphological classifier. In *ELMAR, 2012 Proceedings*, pages 195–198. IEEE.

Horák, A., Gianitsová, L., Šimková, M., Šmotlák, M., and Garabík, R. (2004). Slovak national corpus. In *Text, Speech and Dialogue*, pages 89–93. Springer.

Konkol, M. and Konopík, M. (2013). CRF-based Czech named entity recognizer and consolidation of Czech NER research. *Lecture Notes in Computer Science*, 8082 LNAI:153–160.

Orosz, G. and Novák, A. (2013). Purepos 2.0: A hybrid tool for morphological disambiguation. *International Conference Recent Advances in Natural Language Processing, RANLP*, pages 539–545.

Rusko, M., Juhár, J., Trnka, M., Staš, J., Darjaa, S., Hládek, D., Cerňak, M., Papco, M., Sabo, R., Pleva, M., et al. (2011). Slovak automatic transcription and dictation system for the judicial domain. *Human Language Technologies as a Challenge for Computer Science and Linguistics: 5th Language & Technology Conference*, pages 365–369.

Sak, H., Beaufays, F., Nakajima, K., and Allauzen, C. (2013). Language model verbalization for automatic speech recognition. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, pages 8262–8266.

Spoustová, J. and Spousta, M. (2012). A high-quality web corpus of Czech. In *LREC*, pages 311–315.

Straková, J., Straka, M., and Hajič, J. (2013). A new state-of-the-art Czech named entity recognizer. *Lecture Notes in Computer Science*, 8082 LNAI:68–75.

Thurston, A. (2006). Parsing computer languages with an automaton compiled from a single regular expression. *Implementation and Application of Automata*, pages 285–286.