Building a Web Corpus of Czech

Drahomíra „johanka“ Spoustová, Miroslav Spousta, Pavel Pecina

Institute of Formal and Applied Linguistics
Charles University Prague, Czech Republic
{johanka,spousta,pecina}@ufal.mff.cuni.cz

Abstract

Large corpora are essential to modern methods of computational linguistics and natural language processing. In this paper, we describe an ongoing project whose aim is to build a largest corpus of Czech texts. We are building the corpus from Czech Internet web pages, using (and, if needed, developing) advanced downloading, cleaning and automatic linguistic processing tools. Our concern is to keep the whole process language independent and thus applicable also for building web corpora of other languages. In the paper, we briefly describe the crawling, cleaning, and part-of-speech tagging procedures. Using a prototype corpus, we provide a comparison with a current corpora (in particular, SYN2005, part of the Czech National Corpora). We analyse part-of-speech tag distribution, OOV word ratio, average sentence length and Spearman rank correlation coefficient of the distance of ranks of 500 most frequent words. Our results show that our prototype corpus is now quite homogenous. The challenging task is to find a way to decrease the homogeneity of the text while keeping the high quality of the data.

1. Motivation

Large corpora are essential to modern methods of computational linguistics and natural language processing.

Since middle 90's, the Institute of the Czech National Corpus has been intensively gathering large Czech textual data. Up till now, the Czech National Corpus contains almost 500 million words of unannotated text in three subcorpora (published together as one big corpus SYN): SYN2000 and SYN2005 (CNC, 2005) with representative samples of 100 million words in each and SYN2006PUB containing 300 million words of news articles.

With the sharp increase of storage capabilities of current hardware, we have a long-awaited opportunity to process a huge number of electronically available documents. Some papers demonstrating the power of huge corpora of size of billion words have been already published (Ravichandran and Hovy, 2002), (Talukdan et al., 2006). Compiling corpora of such tremendous size by traditional human-assisted ways is daunting. Direct requisition of newspaper articles, book chapters, magazine essays from publishers is not always possible, not mentioning that only a relatively small data can be obtained this way.

A promising alternative to solve the data sparseness problem is to take advantage of the Internet (c.f. (Kilgarriff, 2001)). The world wide web contains a gigantic number of text documents written in many languages. For many languages, it is possible to compile a corpus of such web pages of multi-billion size (Halacsy et al., 2008).

Our Czech Web Corpus will contain at least 1 billion words of adequate quality text. The big success would be achievement of data ten times bigger than the Czech National Corpus (5 billion words). The estimation of Internet pages containing Czech texts is 50 million, according to several sources. If the average number of words obtained from one page is 100, we could then achieve the desired quantity. A corpus of this size would be comparable to biggest English corpora acquired in the similar way.

2. Crawling

The crawler is a tool to download web pages and other resources from the web. As such, it is the most important part of the web-corpus building chain. Fortunately, there are several mature and freely-available crawlers, mostly developed for small to medium scale search engines. They include Nutch[1], Heritrix[2], WIRE[3], Sherlock Holmes[4] or Egothor[5].

We investigated several crawlers, and results seemed to be encouraging at the beginning, but it turned out that some features are not crucial for a corpus-building process (saving web links data, robust timeout recovery, etc.) while others features are missing and should to be implemented (coarse text-quality measurement during the processing, revisiting policy, near-duplicity elimination).

Consistent with findings of (Kornai and Halacsy, 2008), we encountered the performance decrease of some crawlers (Heritrix) with larger crawls.

We are currently building a high-performance web crawler to download web pages efficiently, based on the components used in other open source crawlers. Our focus is on the near-duplicity elimination and web-page revisiting to enable long-term web crawling and text archiving. Before our crawler is finished, we use raw web data crawled by the Egothor engine (Galamboš, 2006).

3. Cleaning

Our system for web page cleaning, Victor, first described in (Marek et al., 2007) and then, in more detail, in (Spousta et al., 2008), is based on a sequence labeling algorithm with CRF++[6] implementation of Conditional Random Fields (Lafferty et al., 2001). It is aimed at cleaning arbitrary HTML pages by removing all text except headelines and main page content.

[1] http://lucene.apache.org/nutch

[2] http://crawler.archive.org/

[3] http://www.cwr.cl/projects/WIRE/

[4] http://www.ucw.cz/holmes/

[5] http://egothor.org/

[6] http://crfpp.sourceforge.net/

The cleaning process consists of several steps:

1) Filtering invalid documents

Text from input documents is extracted and simple n-gram based classification is applied to filter out documents not in a target language (Czech in our case) as well as documents containing invalid characters (caused mainly by incorrect encoding specified in HTTP or HTML header).

2) Standardizing HTML code

The raw HTML input is passed through Tidy[7] in order to get a valid and parsable HTML tree. During development, we found only one significant problem with Tidy, namely interpreting JavaScript inside the `<script>` element, and employed a simple workaround for it in our system. Except for this particular problem which occurred only once in our training data, Tidy has proved to be a good choice.

3) Precleaning

Afterwards, the HTML code is parsed and parts that are guaranteed not to carry any useful text (e.g. scripts, style definitions, embedded objects, etc.) are removed from the HTML structure. The result is valid HTML code.

4) Text block identification

In this step, the precleaned HTML text is parsed again with a HTML parser and interpreted as a sequence of text *blocks* separated by one or more HTML tags. For example, the snippet `‘‘<p>Hello <b>world</b>!</p>’’` would be split into three blocks, `‘‘Hello’’`, `‘‘world’’`, and `‘‘!’’`. Each of the blocks is then a subject of the labeling task and cleaning.

5) Feature extraction

In this step, a feature vector is generated for each block. All features have a finite set of values[8]. The mapping of integers and real numbers into finite sets was chosen empirically and is specified in the configuration. Most features are generated separately by independent modules. This allows for adding other features and switching between them for different tasks.

6) Learning

Each *block* occurring in training data was manually assigned one of the following labels: *header*, *text* (*content blocks*) or *other* (*noisy blocks*).

The sequence of feature vectors including labels extracted for all blocks from the training data are then transformed into the actual features used for training the CRF model according to offset specification described in a template file.

7) Cleaning

Having estimated parameters of the CRF model, an arbitrary HTML file can be passed through steps 1–4, and its blocks can be labeled with the same set of labels as described above. These automatically assigned labels are then used to produce a cleaned output. Blocks labeled as *header* or *text* remain in the document, blocks labeled as *other* are deleted.

4. Automatic linguistic processing

We aim the Czech Web Corpus to be automatically linguistically processed at the same level as the Czech National Corpus (CNC, 2005) is, i.e. by the state-of-the-art morphological analysis and POS tagger (Spoustová et al., 2009).

In addition, we apply further processing: dependency parsing, named entity recognition, and (if necessary tools are available) also valency frame disambiguation. The prototype of the Czech Web Corpus is currently available through standard web query interface Bonito and we are investigating possibility to use tree-query engine and interface (such as TrEd) as the data available are very large compared to currently available ones.

4.1. Part-of-speech tagging

We will describe here in more detail our part-of-speech tagger.

The system it is based on Raab’s (Votrubec, 2006) implementation of (Collins, 2002), which has been fed at each iteration by a different dataset consisting of the supervised and unsupervised part: precisely, by a concatenation of the manually tagged training data (the WSJ portion of the PTB 3 for English, morphologically disambiguated data from PDT 2.0 for Czech) and a chunk of automatically tagged unsupervised data. The “parameters” of the training process (feature templates, the size of the unsupervised chunks added to the trainer at each iteration, number of iterations, the combination of taggers that should be used in the autotagging of the unsupervised chunk, etc.) have been determined empirically in a number of experiments on a development data set.

The final taggers have surpassed the current state-of-the-art taggers by significant margins (we have achieved 4.12 % relative error reduction for English and 4.86 % for Czech over the best previously published results, single or combined), using a single tagger. For more detail see (Spoustová et al., 2009).

5. Comparison with current corpora

Once we have a large corpus in our hands, we would like to compare it somehow to other resources available. Ideally, we would like to acquire data that are as similar to currently available corpora as possible.

First, we focus on word and sentence measures that may be easily extracted from the texts, such as mispelled-word ratio, average sentence length, or distribution of various word types (proper nouns, verbs, pronouns). If the differences of these measures are too big, we may conclude that texts included in the corpus different from those in reference corpus a lot.

For initial comparison experiments, we chopped a 50 million token portion of both the CNC SYN2005 and our prototype web corpus obtained using the process described above. We split both corpora into 1 million-token length parts and estimate mean and standard deviation for experiments where applicable.

[7] http://tidy.sourceforge.net/

[8] This is a limitation of the CRF tool used.

First, we show (Figure 1) there is not big difference in Part-of-Speech tag proportions of SYN2005 and WEB data.

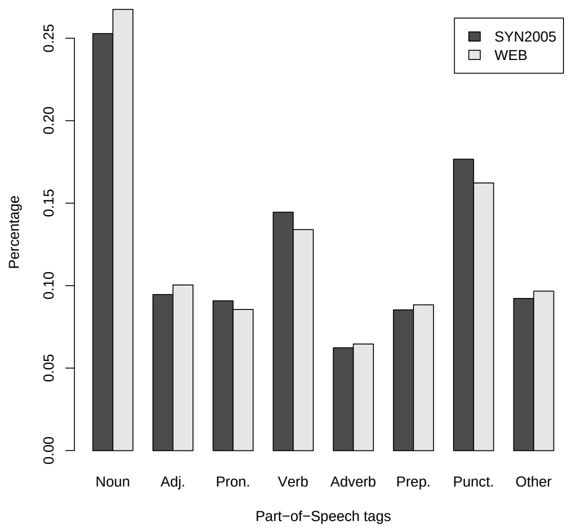

Figure 1: Part-of-Speech tags percentage in SYN2005 and WEB corpus. *Others* category include numerals, interjections, particles, etc.

According to Figure 2 it turns out that the web corpus contains in general longer sentences than the reference SYN2005 corpus. This may be caused by the cleaning algorithm as it was trained to extract ”nice” (and usually longer) sentences. We would like to focus on adding more sight into this issue and try to determine whether this difference is really caused by the cleaning algorithm, or is it a general feature of the web data.

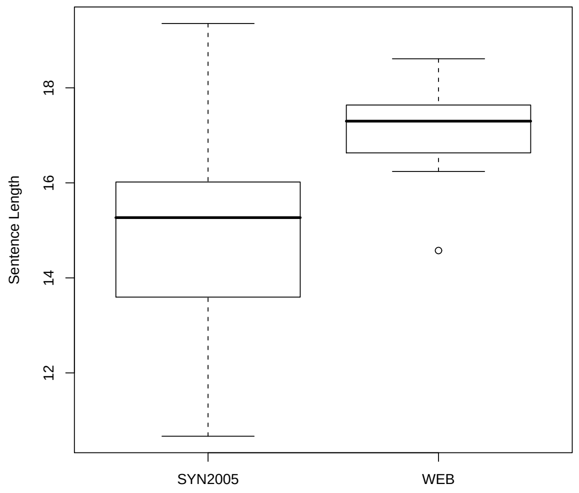

Figure 2: Average sentence length comparison of SYN2005 and WEB corpus.

In Figure 3 we can see that web corpus contains more out-of-vocabulary words than the SYN2005 corpus. On the basis of analysis of the most frequent OOV words, we may conclude that one of the main reasons for this difference is missing word diacritics. Many of Internet users still write Czech words without diacritics (i.e. ”c” instead of č, ”a” instead of á etc.). There are several ways how we can deal with this phenomena – we can ignore it (keep words unchanged), drop such words or try to automatically reconstruct them.

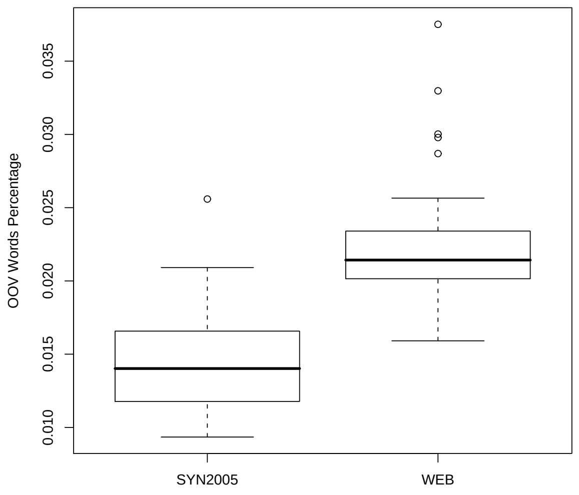

Figure 3: Box-plots of out-of-vocabulary words percentage for SYN2005 and WEB corpus.

In fact, the corpus comparison is quite difficult and challenging task itself. (Kilgarriff, 2001) explores several different measures of corpus similarity (and homogeneity), such as perplexity and cross-entropy of the language models, χ^2 statistics or Spearman rank correlation coefficient. Using the ”Known-Similarity Corpora”, he finds, that for the purpose of corpora similarity comparison, χ^2 and Spearman rank methods work significantly better than the cross-entropy based ones.

For our data sets, we compute Spearman rank correlation coefficient of the distance of ranks of 500 most frequent words. The difference is small for text where common word patterns are similar. As the measure is independent of the corpora size, we can directly compare both homogeneity (intra-corpus) and similarity (inter-corpus) results.

Table 1 shows that both SYN2005 and the prototype web corpus is quite homogeneous (Spearman coefficient approaches 1). Higher average homogeneity (with lower variance) of the Web data is probably caused by the fact, that some genres (e.g. fiction and poetry), although present on the web, are shaded by tons of news and product description texts. The challenging task is to find a way to decrease the homogeneity of the text while keeping the high quality of the data.

6. Conclusion

Our paper presented the on-going project of the Czech Web Corpus and corresponding tools. We have presented the

SYN2005	WEB	SYN2005 vs WEB
0.938 (0.023)	0.986 (0.004)	0.741

Table 1: Spearman rank correlation coefficient as a measure of homogeneity (SYN2005 and WEB) and inter-corpus similarity (SYN2005 vs WEB). Homogeneity is measured using 10 random partitions of the corpus divided into two halves and the results are average and standard deviation (in brackets).

main parts of the projects (crawling, cleaning, linguistic processing) and compared the prototype 50 million token web corpus with the existing Czech National Corpus using various statistics.

Acknowledgments

The research described here was supported by the projects *GA405/09/0278* of the Grant Agency of the Czech Republic and *LC536* of *Ministry of Education, Youth and Sports* of the Czech Republic.

7. References

CNC, 2005. *Czech National Corpus – SYN2005*. Institute of Czech National Corpus, Faculty of Arts, Charles University, Prague, Czech Republic.

Michael Collins. 2002. Discriminative Training Methods for Hidden Markov Models: Theory and Experiments with Perceptron Algorithms. In *EMNLP '02: Proceedings of the ACL-02 conference on Empirical methods in natural language processing*, volume 10, pages 1–8, Philadelphia, PA.

Leo Galamboš. 2006. Egothor, full-featured text search engine written entirely in java. http://www.egothor.org/.

Peter Halacsy, Andrals Kornai, Peter Nemeth, and Daniel Varga. 2008. Parallel creation of gigaword corpora for medium density languages — an interim report. In *Proceedings of Language Resource and Evaluation Conference (LREC08)*, Marrakech, Morocco.

Adam Kilgarriff. 2001. Comparing corpora. *International Journal of Corpus Linguistics*, 6(1):97–133.

Andras Kornai and Peter Halacsy. 2008. Google for the linguist on a budget. In *Proceedings of 4th Web as Corpus Workshop, LREC 2008*, Marrakech, Morocco.

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. 18th International Conf. on Machine Learning*. Morgan Kaufmann, San Francisco, CA, USA.

Michal Marek, Pavel Pecina, and Miroslav Spousta. 2007. Web page cleaning with conditional random fields. In *Proceedings of the Web as Corpus Workshop (WAC3), Cleaneval Session*, Louvain-la-Neuve, Belgium.

D. Ravichandran and E.H. Hovy. 2002. Learning surface text patterns for a question answering system. In *Proceedings of the 40th ACL conference*, Philadelphia, PA.

Miroslav Spousta, Michal Marek, and Pavel Pecina. 2008. Victor: the web-page cleaning tool. In *Proceedings of the Web as Corpus Workshop (WAC-4)*, Marrakech, Morocco.

Drahomíra "johanka" Spoustová, Jan Hajič, Jan Raab, and Miroslav Spousta. 2009. Semi-supervised training for the averaged perceptron POS tagger. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 763–771, Athens, Greece, March. Association for Computational Linguistics.

Talukdan, Brants, Lieberman, and Pereira. 2006. A context pattern induction method for ne extraction. In *ACL*.

Jan Votrubec. 2006. Morphological Tagging Based on Averaged Perceptron. In *WDS'06 Proceedings of Contributed Papers*, pages 191–195, Prague, Czech Republic. Matfyzpress, Charles University.