Linguistic Corpus Search

Christian Biemann[*], Uwe Quasthoff[*], Christian Wolff[†]

[*]Leipzig University
Computer Science Institute
Natural Language Processing Dept.
Augustusplatz 10/11
04109 Leipzig, Germany
{biem, quasthoff}@informatik.uni-leipzig.de

[†]Regensburg University
Institute for Media, Information and Cultural Studies
Media Computing Dept.
Universitätsstr. 31
93040 Regensburg, Germany
christian.wolff@sprachlit.uni-regensburg.de

Abstract

Searching corpora with linguistic questions requires both additional information encoded in the corpus and efficiency as in "traditional" search engines. We describe a search engine-like approach to querying plain as well as part-of-speech-tagged monolingual corpora. This approach makes use of a 'minimalist' query language which nevertheless allows powerful searches by optionally ignoring positional as well as inflectional features in the corpus sentences. Many queries can be formulated without detailed training via a simple web-based front-end. Relevant applications of this search tool in knowledge extraction are discussed as well.

1. Introduction

Searching for multiword structures in corpora is important for multiple reasons and has many applications from purely linguistic questions concerning phrases to proper name detection in Information Retrieval. We describe an (almost) language independent approach for corpus preprocessing and querying. The query language is designed for users who are expert linguists but may not be willing to learn formal languages like regular for formulating linguistic problems. The fact that users rarely employ advanced search features has been shown in many studies on search engine effectiveness (see Jansen et al. 2000).

Making use of pre-calculated collocations the search engine is able to return sentences showing search terms in a typical context. This corpus search engine will be available via a web-based front-end for corpora of the size from one to ten million sentences for about ten different languages within 2004.

2. Related Work

Over the last dozen years, the work of corpus linguists has substantially changed. In the early Nineties of the last century, linguists concentrated on finding and analyzing single sample sentences of a specific phenomenon. Today, accessing large corpora automatically in order to find not only samples, but also frequency information is commonplace, shifting the main interest from the inspection and analysis of theoretically possible constructions to the examination of naturally occurring language (see Volk 2002).

The linguist's need for corpus search engines has recently resulted in a variety of search tools that allow finding special constructions or phrases rather than content. While search engines are now at a point where people really like to use them because of speed, well-grown ranking mechanisms and sufficient coverage in web crawling, they fail on searches on linguistic problems.

For the German Language, the *COSMAS Search Engine* from the Institut für Deutsche Sprache (IDS; Mannheim, Germany, see *http://www.ids-mannheim.de/kt/corpora.html*) has to be mentioned as a prominent example for this kind of tool. *COSMAS* offers a Web Client and a multitude of search options, ranging from single word queries to complex connections of a diversity of operators allowing for search on syntactical structures. However, the query language, be it the textual or the graphical option, is quite complicated and requires extensive knowledge about word classes and logical operators. While the former is very familiar to linguists, the latter quite often is not. The *IMS Corpus Workbench* (see Christ 1994, Christ & Schulze 1996) suffers from the same problem: though powerful and available for many corpora in several languages, the query language requires some programming skills and the syntax does not allow for trial-and-error by starting with a single word query.

Experiments on improving search engines by the means of linguistic information (see Bruder et al. 2001) have not found their way into real world application, in fact it happens the other way around: more recent approaches try to employ the Web as data source on behalf of the Web providing the largest textual database in the world, as described in Kilgariff 2003. Implementations rely on search engines as providers of raw material on which the linguistic search engine builds upon. At the *WebCorp Initiative* (see *http://www.webcorp.org.uk/*) the underlying search engine can be chosen and parameterized by (top level) domain endings, as well as static corpora of different domains can be used. The query language is easy and comes with some optional regular expression syntax. Outstanding features are calculations of collocation frequencies within a configurable window of the query match as well as the listing of target words when using wildcards in queries. A minor drawback is that word separation is confused by characters others than those of the 26-letter alphabet, making it unusable for most of the world's languages. A big difference to the previously named approaches is the absence of any kind of tagging, making the search for specific constructions a surge.

A very recent implementation is the *Linguist's Search Engine* (see *http://lse.umiacs.umd.edu*). This application provides the user with an easy-to-learn query language by performing *queries by example*, without losing expressive power by operating on fully parsed corpora. The query is transformed into a parse tree and the user may loosen some constraints on it to add some generality. The search is performed on a relatively small corpus of currently approx. 3 million sentences. While it is possible to obtain personal corpora automatically by web searches, parsing

these results in long waiting times and the whole project works for English only at the moment.

3. Corpus Preprocessing

Our attempt is to provide an easy and intuitive way of exploring corpora for many languages. Leaving out parse trees for the moment due to low coverage with respect to rules and languages, we concentrate on the use of words, affixes, and part-of-speech (POS) tags if available. The use of fixed-size corpora for many languages makes it possible to compare frequencies of constructions and phenomena. A definition of user corpora will be available through keyword-based web search processing in some later step.

The corpus search described in this paper operates on monolingual sentence-separated corpora. In that way we can assure that search queries do not cross sentence boundaries. Hence, we get sentences as results for the search queries. If possible, the corpus for a given language is tagged using a POS tagger. For such a tagged corpus, tags may be included in the queries. At this stage, the corpus may be viewed as a text file with one sentence at each line. In the case of a tagged corpus, each word is followed by its POS tag. To be able deal with very large corpora as well we actually use a relational database with an additional index structure.

For later reference, we give here a tagged English and an untagged German sentence. The tags are separated by the sign '|':

- `Officials|NN2 still|RR have|VH0 not|XX identified|VVN the|AT owner|NN1 of|IO the|AT house|NNL1 ,|YC he|PPHS1 said|VVD |.`
- `Jetzt hat er den vierten Krieg vom Zaun gebrochen.`

Tagging is done using the Susanne tag set for English corpora (see Sampson 1995) and the Stuttgart-Tübingen tag set for German texts (STTS; see Brants 2000 and *http://www.sfs.nphil.uni-tuebingen.de/Elwis/stts/ Wortlisten/WortFormen.html*).

To find sentences with typical usage for a searched pattern we calculate co-occurrences as described in Biemann et al 2004. If inflection information is available, we additionally connect inflected forms with their corresponding base forms. This, for instance, allows the search for all inflected forms of a pattern.

In a longer perspective, we also want to deal with corpora of higher annotation level, i.e. treebanks or the output of a chunk parser.

4. Corpus Search for Single Words

In this section we describe how to search for word with the optional usage of tags. If the wildcard '*' is used in the search string within a word, in a first step we look for all words satisfying the given pattern. Next, all these words are searched in the corpus. The wildcard '*' for parts of words has the following properties:

- It abbreviates zero, one or more letters in a word or in a tag, not the sign '|' or white spaces. For example, "`house*|NN*`" will match both "`house|NNL1`" and "`houses|NNL2`", but not "`houses|VVZ] the|AT camp|NNL1`"
- If, in the case of a tagged corpus, either the searched word or its tag are unspecified, the corresponding part in the search pattern can be ignored.

Hence, "`of|*`" can be abbreviated to "`of`", "`*|VVAD`" to "VVAD" and "`*|*`" to "`*`".So if the user is not familiar with tags, he can fully ignore tags in his queries.

5. Corpus Search for Phrases

For phrase searches we use the same notation as above: The wildcard '*' is used for at most one word. The words matching the query will be searched in the ordering given in the query without any additional words filled in. Searching for "`ein* * vom Zaun brechen`", we will find phrases like "`einen Streit vom Zaun brechen`", "`einen Krieg vom Zaun brechen`" and so on. The ordering of the results is described in the section on ranking below.

5.1. Global Search Flags: Inflection and Word Order

When counting frequencies of phrases, it is not sufficient to perform string matching with the given phrase on the corpus. Because of inflection and word order many occurrences of the phrase are missed by such an naïve approach.

In the following we give four sample sentences containing the German phrase "`einen Streit vom Zaun brechen`" (to pick up a quarrel; literally: to break an argument from the fence):

(1) `Natürlich wollte er keinen Streit mit dem Kanzler vom Zaun brechen.`
(2) `Sie brechen immer wieder einen Streit vom Zaun.`
(3) `Er brach einen heftigen Streit vom Zaun.`
(4) `Da wurde ein Streit – noch dazu ein sinnloser - vom Zaun gebrochen!`

This illustrates that both – inflection and word order – have to be taken into consideration when looking for sentences containing a phrase. To deal with this, we introduced two global search flags, *ignore word order* and *unify inflection*. If *ignore word order* is checked, all sentences containing the search words are returned, resulting in the retrieval of example (2). The additional setting of *unify inflection* results is the retrieval of examples (2), (3), and (4). To match with example (1), the pattern has either to be extended to "`*einen Streit vom Zaun brechen`" or POS information like "`*|DET Streit vom Zaun brechen`" has to be used.

5.2. Ranking

Using patterns like above, there is no natural sort order for the search results. On the other hand, human users consider some sentences as more typical than others. To model typical usage, we prefer sentences

(1) containing additional collocations as typical objects,
(2) sentences containing the search patterns in the given order (in the case of variable word order), and

(3) sentences *not* containing subordinate clause separators such as "`,`", "`;`" and "`-`".

The criteria given here will be reflected in any result set if the corpus is sorted according to them in a preprocessing step. Hence, all search results are ranked automatically if the results are extracted from the corpus in their sorted order.

5.3. Using Corpus Search for Knowledge Extraction

Having an extraction tool for patterns based on POS tags and anchor words at hand, it is possible to use it for knowledge extraction purposes, e.g. the semiautomatic building of fact databases or ontologies from corpora.

Kim et al. 2004 describe a system that extracts ontological triplets from the web for databases of galleries on artists using patterns and inductive logic programming (ILP) methods on the extracted results. Here, their patterns are hand made and the linguistic preprocessing includes syntactic parsing and named entity recognition. We believe that the sheer mass of occurrences of a fact will give hints which occurrences to believe and which to discard.

The following sentences were extracted from the English corpus using the queries " `|ADJ |N like |N and |N`" and " `|N like |N and |N`", illustrating the extraction of the hyponymy relation (hyperonyms are show in bold italic face, hyponym candidates in bold face).

- But […] lately he was talking about defense of the forest and union of ***jungle peoples*** like **tappers** and **Indians.**
- "Not only will these changes reduce the hazards from backups and passing, but they will also save energy and reduce ***environmental problems*** like **noise** and **exhaust**," Schulte said.
- „Where ***diseases*** like **AIDS** and **Hepatitis-B** are concerned, our health care professionals are truly on the front line," McLaughlin said. "They're working to contain the problem."
- Zirkle and his staff put together a curriculum that offers associate degrees in ***business areas*** like **accounting** and **computer science** and in the **humanities**.

5.4. Efficiency

Due to their experiences with web search engines, users expect immediate results. Unfortunately, the query language described above may lead to complex queries which require some processing time. Therefore, we distinguish between *rapid answer mode* and *slow answer mode*. In rapid answer mode the result is given within about a second and will be displayed in the web front-end. As in web search, this result contains only the first (lets say) 50 results. More results are available with a new query. In the slow answer mode, a larger or even full result set will be calculated offline and provided by e-mail. Each query will be tested for execution in rapid mode. If the rapid mode fails, the system automatically switches to slow mode.

For the rapid answer mode we use the following index structure. While web search engines usually only have a full text index with words as smallest index term, we use a *4+gram index*: This 4+gram index lists the occurrences of any n-gram of letters for n≥4, not containing white spaces. This type of index can effectively being used for queries containing wildcards but at least four consecutive letters. At the moment, there is no index for POS tags or the concatenation of words and POS tags. The searching algorithm starts with the whole corpus as potential answer set and processes each query in the following two steps:

Step1: If the query contains at least four consecutive letters as part of a word, the 4+gram index is used to restrict the potential answer set to those sentences containing the given 4+gram(s).

Step2: In a post processing step each sentence in the potential answer set is tested whether it fulfills the query. If 50 results (or all, if less) are found within a second, the result is presented. Otherwise the system switches to slow answer mode.

6. Availability

At the moment, we maintain untagged corpora of substantial size (from 1 to 50 Million sentences) for about 20 natural languages (cf. Biemann et al. 2004 and *http://wortschatz.uni-leipzig.de*). A variety of trainable, language-independent taggers are currently in use for the analysis of these corpora, e.g. Brill's Tagger (Brill 1992) or Brants' TNT (Brants 2000). We use the pre-trained parameters of the latter to tag German and English. The use of Schmid's TreeTagger (cf. Schmid 1996) including the parameter files for English, German, French, and Italian is in preparation. For higher levels of annotation, we use standard rules for the lemmatization of English and an example-based trainable base form reduction for German. Through the modularity and the language-independence of the linguistic search engine, it can be applied to any new language and annotation level available from other sources. A screenshot of the forthcoming web site (see *http://www.wortschatz.uni-leipzig.de/corpussearch*) can be found at the end of this paper (fig. 1).

7. Conclusion

The prototype of a corpus search engine discussed in this paper offers powerful linguistic search operations without the cognitive load of a complex formal search language. As the proposed set of tools is available for various monolingual corpora, we plan to run practical tests of the search engine in close cooperation with corpus linguists in the near future. Additional types of search operations will be added to the search environment as required by linguists' needs.

8. References

Biemann, Ch.; Bordag, S.; Heyer, G.; Quasthoff, U.; Wolff, Ch. (2004): "Language-independent Methods for Compiling Monolingual Lexical Data." In: Proceedings of The Fifth International Conference on Intelligent Text Processing and Computational Linguistics (CICling-2004), Seoul, South Korea, February 2004, pp.

217-228 [= Lecture Notes in Computer Science Vol. 2945, Berlin et al.: Springer].
Brants T. (2000): "TnT - A Statistical Part-of-Speech Tagger." In: Proceedings of the Sixth Applied Natural Language Processing Conference ANLP-2000, Seattle, pp. 224-231.
Brill, E. (1992): "A Simple Rule- Part of Speech Tagger." In: Proceedings of the Third Conference on Applied Computational Linguistics, Trento, Italy, 1992.
Bruder, I.; Düsterhöft, A.; Becker, M.; Bedersdorfer, J.; Neumann, G. (2001): "GETESS: Constructing a Linguistic Search Index for an Internet Search Engine." In: Bouzeghoub, M.; Kedad, Z.; Métais, E. (Eds.): Natural Language Processing and Information Systems. Proc. 5th International Conference on Applications of Natural Language to Information Systems (NLDB 2000), Versailles, June 2000, pp. 227-238 [= Lecture Notes in Computer Science Vol. 1959, Berlin et al.: Springer].
Christ, O. (1994): "A Modular and Flexible Architecture for an Integrated Corpus Query System." In: Proceedings of COMPLEX'94: 3rd Conference on Computational Lexicography and Text Research, Budapest, pp. 23-32.
Christ, O.; Schulze, B.M. (1996): "Ein flexibles und modulares Anfragesystem für Textcorpora". In: Feldweg, H.; Hinrichs, E. W. (eds.) (1996). Lexikon und Text: wiederverwendbare Methoden und Ressourcen zur linguistischen Erschließung des Deutschen, Tübingen: Niemeyer, pp. 121-134 [= *Lexicographica Series maior*, *Vol. 73*].
Heyer, G.; Quasthoff, U.; Wolff, Ch. (2002): "Knowledge Extraction from Text: Using Filters on Collocation Sets." In: Proc. Third International Conference On Language Resources, LREC-02, Las Palmas, Spain, May 2002, Vol. III, pp. 241-246.
Jansen, B. J.; Spink, A.; Saracevic, T. (2000): "Real Life, Real Users, and Real Needs: A Study and Analysis of User Queries on the Web." In: Information Processing & Management 36(2) (2000), pp. 207-227.
Kehoe, A.; Renouf, A. (2002): "WebCorp: Applying the Web to Linguistics and Linguistics to the Web." In: Proceedings World Wide Web 2002 Conference (WWW2002), Honolulu, Hawaii, May 2002.
Kilgarriff, A. (2003): "Linguistic Search Engine." In: Proceedings of The Shallow Processing of Large Corpora Workshop (SProLaC 2003), Corpus Linguistics 2003, Lancaster, March 2003, pp.53-58 [http://www.bultreebank.org/SProLaC/paper06.pdf].
Kim, S.; Lewis, P.; Martinez, K. (2004): "The Impact of Enriched Linguistic Annotation on the Performance of Extracting Relation Triples." In: Proceedings of The Fifth International Conference on Intelligent Text Processing and Computational Linguistics (CICling-2004), Seoul, South Korea, February 2004, pp. 547-558 [= Lecture Notes in Computer Science Vol. 2945, Berlin et al.: Springer].
Sampson, G. (1995). English for the Computer. The SUSANNE Corpus and Analytic Scheme. OUP 1995.
Schmid, H. (1996): "Improvements in Part-of-Speech Tagging with an Application to German." Proc. EACL SIGDAT Workshop, Dublin, March 1995. In: Feldweg, H.; Hinrichs, E. W. (eds.) (1996). Lexikon und Text: wiederverwendbare Methoden und Ressourcen zur linguistischen Erschließung des Deutschen, Tübingen: Niemeyer, pp. 47-50 [= *Lexicographica Series maior*, *Vol. 73*].
Volk, M. (2002): "Using the Web as Corpus for Linguistic Research." In: Pajusalu, R.; Hennoste, T. (eds.): Tähendusepüüdja. Catcher of the Meaning. A Festschrift for Professor Haldur Õim. Tartu/Estonia: University of Tartu [= Papers in Estonian Cognitive Linguistics / Publications of the Department of General Linguistics, Vol. 3].

9. Appendix: Prototype Screenshot

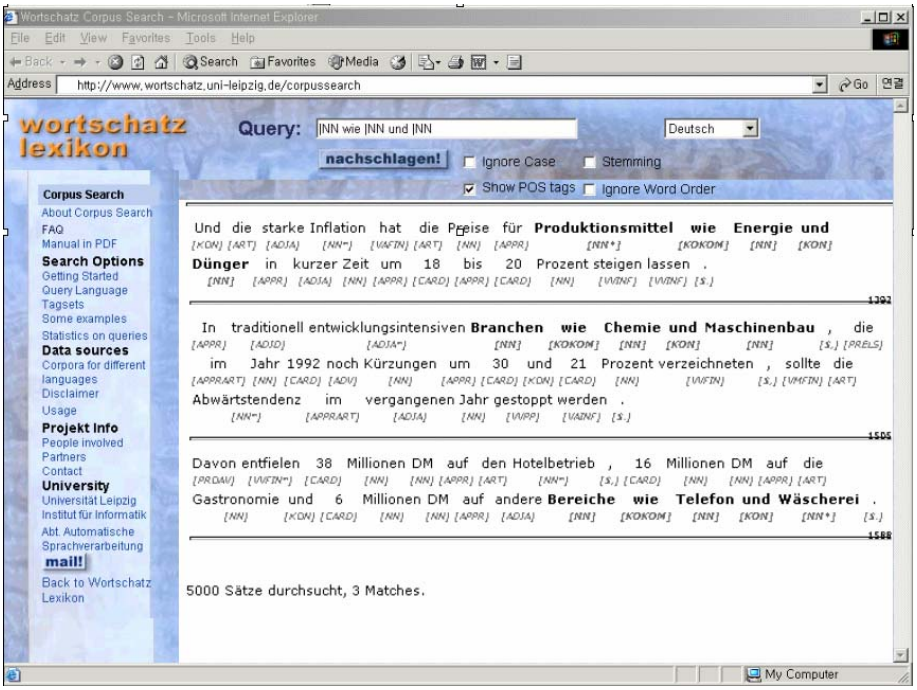

Figure 1: Screenshot of the Wortschatz Corpus Search Web Front-end (prototype preview)