Using the web as corpus for linguistic research

Volk, M
Using the Web as Corpus for Linguistic Research

Martin Volk
University of Zurich
Seminar of Computational Linguistics
Winterthurerstr. 190
CH-8057 Zurich
volk@ifi.unizh.ch


1. Introduction

In the last decade the working methods in Computational Linguistics have changed drastically. Fifteen years back, most research focused on selected example sentences. Nowadays the access to and exploitation of large text corpora is commonplace. This shift is reflected in a renaissance of work in Corpus Linguistics and documented in a number of pertinent books in recent years, e.g. the introductions by (Biber et al. 1998) and (Kennedy 1998) and the more methodologically oriented works on statistics and programming in Corpus Linguistics by (Oakes 1998) and (Mason 2000).

The shift to corpus-based approaches has entailed a focus on naturally occurring language. While most research in the old tradition was based on constructed example sentences and self-inspection, the new paradigm uses sentences from machine-readable corpora. In parallel the empirical approach requires a quantitative evaluation of every method derived and every rule proposed.

Corpus Linguistics, in the sense of using natural language samples for linguistics, is much older than computer science. The dictionary makers of the 19th century can be considered Corpus Linguistics pioneers (e.g. James Murray for the Oxford English Dictionary or the Grimm brothers for the Deutsches Wörterbuch). But the advent of computers has changed the field completely.

Linguists started compiling collections of raw text for ease of searching. In a next step, the texts were semi-automatically annotated with lemmas and recently with syntactic structures. First, corpora were considered large when they exceeded one million words. Nowadays, large corpora comprise more than 100 million words. And the World Wide Web (WWW) can be seen as the largest corpus ever with more than one billion documents.

Professor Ōim and the Computational Linguistics group at the University of Tartu have participated in this field early on. They have compiled a Corpus of Estonian Written Texts and a Corpus of Old Estonian Texts. Later they have been active in the EuroWordNet project and contributed the Estonian side to this thesaurus project.

In addition to written text corpora some researchers have compiled spoken language corpora with audiotapes and transcriptions. There are also corpora with parallel videos and texts that allow the analysis of gestures. In this paper we will focus on text corpora for written language.
The current use of corpora falls into two large classes. On the one hand, they serve as the basis for intellectual analysis, as a repository of natural language data for the linguistic expert. On the other hand, they are used as training material for computational systems. A program may compute statistical tendencies from the data and derive or rank rules which can be applied to process and to structure new data.

Corpus Linguistics methods are actively used for lexicography, terminology, translation and language teaching. It is evident that these fields will profit from annotated corpora (rather than raw text corpora). Today's corpora can automatically be part-of-speech tagged and annotated with phrase information (NPs, PPs) and some semantic tags (e.g. local and temporal expressions). This requires standard tag sets and rules for annotation. As Geoffrey Sampson has put it: "The single most important property of any database for purposes of computer-assisted research is that similar instances be encoded in predictably similar ways."1

But corpora distributed on tape or CD-ROM have some disadvantages. They are limited in size, their availability is restricted by their means of distribution and they do no longer represent the current language by the time they are published. The use of the web as corpus avoids these problems. It is ubiquitously available and due to its dynamic nature represents up to date language use.

The aim of this paper is to show a number of examples of how the web has been used as a corpus for linguistics research and their applications in natural language processing. Based on this we list the advantages and limits of the web as corpus. We conclude with a proposal for a linguistic search engine to query the web.

2. Approaches of Using the Web for Corpus Linguistics

Using the web for Corpus Linguistics is a very recent trend. The number of approaches that are relevant to Computational Linguistics is still rather small. But already the web has been tried for tasks on various linguistic levels: lexicography, syntax, semantics and translation.

2.1. Lexicography

Almost from its start the web has been a place for the dissemination of lexical resources (word lists in various languages). There are a large number of interfaces to online dictionaries that are more or less carefully administered. But the more interesting aspect from a computational perspective comes from discovering and classifying new lexical material from the wealth of texts in the web. This includes finding and classifying new words or expressions and to gather additional information such as typical collocations, subcategorization requirements, or definitions.

(Jacquemin and Bush 2000) describe an approach of learning and classifying proper names from the web. This is a worthwhile undertaking since proper names are an open word class with new names being continually invented. Their system works in three steps. First, a harvester downloads web pages retrieved by a search engine following a query to a keyword pattern (e.g. universities such as; the following list of politicians). Second, shallow parsers are used to extract candidate names from enumerations, lists, tables and anchors. Third, a filtering module cleans the candidates from leading determiners or trailing unrelated words.

The example sentence (1) will lead to the acquisition of the university names University of Science and Technology and University of East Anglia with acronyms for both names and the location of the former.

1 Quote from a talk at Corpus Linguistics 2001 in Lancaster: "Thoughts on Twenty Years of Drawing Trees".
(1) While some universities, such as the University of Science and Technology at Manchester (UMIST) and the University of East Anglia (UEA), already charge students using the internet ...

Names are collected for organizations (companies, universities, financial institutions etc.), persons (politicians, actors, athletes etc.), and locations (countries, cities, mountains etc.). The precision of this acquisition and classification process is reported as 73.6% if the names are only sorted into wide semantic classes. For fine-grained classes it is 62.8%. These are remarkable results that demonstrate the principal feasibility of the approach.

2.2. Syntax

There are a number of web sites that allow the parsing of sentences at different levels of sophistication (see www.ifi.unizh.ch/cl/volk/InteractiveTools.html for a collection of links to such systems for English and German). Most are demos that work only for single sentences or on short texts. But again the more interesting question is: How can the vast textual resources of the web be exploited to improve parsing? In this section we summarize some of our own research on using the web (i.e. frequencies obtained from search engines) to resolve PP attachment ambiguities (see also Volk 2000 and Volk 2001).

2.2.1. Disambiguating PP Attachment

Any computer system for natural language processing has to struggle with the problem of ambiguities. If the system is meant to extract precise information from a text, these ambiguities must be resolved. One of the most frequent ambiguities arises from the attachment of prepositional phrases (PPs).

An English sentence consisting of the sequence verb + NP + PP is a priori ambiguous. (The same holds true for any German sentence.) The PP in example sentence 2 is a noun attribute and needs to be attached to the noun, but the PP in 3 is an adverbial and thus part of the verb phrase.

(2) Peter reads a book about computers.
(3) Peter reads a book in the subway.

If the subcategorization requirements of the verb or of the competing noun are known, the ambiguity can sometimes be resolved. But often there are no clear requirements. Therefore, there has been a growing interest in using statistical methods that reflect attachment tendencies. The first idea was to compare the cooccurrence frequencies of the pair verb + preposition and of the pair noun + preposition.

But, subsequent research has shown that it is advantageous to include the core noun of the PP in the cooccurrence counts. This means that the cooccurrence frequency is computed over the triples V+P+N2 and N1+P+N2 with N1 being the possible reference noun, P being the head and N2 the core noun of the PP. Of course, the frequencies need to be seen in relation to the overall frequencies of the verb and the noun occurring independently of the preposition. For example sentence 2 we would need the triple frequencies for (read, about, computer) and (book, about, computer) as well as the unigram frequencies for read and book.

Obviously it is very difficult to obtain reliable frequency counts for such triples. We therefore used the largest corpus available, the WWW. With the help of a WWW search engine we obtained frequency values ('number of pages found') and used them to compute cooccurrence values. Based on the WWW frequencies we computed the cooccurrence values with the following formula (X can be either the verb V or the head noun N1):
cooc(X,P,N2) = freq(X,P,N2) / freq(X)

For example, if some noun N1 occurs 100 times in a corpus and this noun co-occurs with the PP (defined by P and N2) 20 times, then the cooccurrence value cooc(N1,P,N2) will be 20 / 100 = 0.2. The value cooc(V,P,N2) is computed in the same way. The PP attachment decision is then based on the higher cooccurrence value. When using the web, the challenge lies in finding the best query formulation with standard search engines.

**Query formulation**

WWW search engines are not prepared for linguistic queries, but for general knowledge queries. For the PP disambiguation task we need cooccurrence frequencies for full verbs + PPs as well as for nouns + PPs. From a linguistic point of view we will have to query for

1. a noun N1 occurring in the same phrase as the PP that contains the preposition P and the core noun N2. The immediate sequence of N1 and the PP is the typical case for a PP attached to a noun but there are numerous variations with intervening genitive attributes or other PPs.
2. a full verb V occurring in the same clause as the PP that contains the preposition P and the core noun N2. Unlike in English, the German verb may occur in front of the PP or behind the PP depending on the type of clause.

Since we cannot query with these linguistic operators ('in the same phrase', 'in the same clause'), we have approximated these cooccurrence constraints with the available operators. We used the NEAR operator (V NEAR P NEAR N2). In AltaVista this operator restricts the search to documents in which its argument words co-occur within 10 words.

Let us demonstrate the method for example sentence 4 which we took from a computer magazine text. It contains the PP *unter dem Dach* following the noun *Aktivitäten*. The PP could be attached to this noun or to the verb *bündeln*.

(4) *Die deutsche Hewlett-Packard wird mit Beginn des neuen Geschäftsjahres ihre Aktivitäten unter dem Dach einer Holding bündeln.*

[Hewlett-Packard Germany will bundle its activities under the roof of a holding at the beginning of the new business year.]

We queried AltaVista for the following triple and unigram frequencies which led to the cooccurrence values in column 4. Since the value for the verb triple is higher than for the noun triple, the method will correctly predict verb attachment for the PP.

<table>
<thead>
<tr>
<th>(X,P,N2)</th>
<th>freq(X NEAR P NEAR N2)</th>
<th>freq(X)</th>
<th>cooc(X,P,N2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Aktivitäten, unter, Dach)</td>
<td>58</td>
<td>246,238</td>
<td>0.00024</td>
</tr>
<tr>
<td>(bündeln, unter, Dach)</td>
<td>47</td>
<td>14,674</td>
<td>0.00320</td>
</tr>
</tbody>
</table>

For the evaluation of the method we have extracted 4383 test cases with an ambiguously positioned PP from a German treebank. 61% of these test cases have been manually judged as noun attachments and 39% as verb attachments. With the triple cooccurrence values computed from web queries we were able to decide 63% of the test cases, and we observed an attachment accuracy (percentage of correctly disambiguated cases) of 75% over the decidable cases.
This result is based on using the word forms as they appear in the sentences, i.e. possibly inflected verbs and nouns some of which are rare forms and lead to zero frequencies for the triples. Since a coverage of 63% is rather low, we experimented with additional queries for base forms. We combined the frequency counts for base forms and surface forms and in this way increased the coverage to 71% (the accuracy stayed at 75%).

The good news is that such simple queries to standard search engines can be used to disambiguate PP attachments. The result of 75% correct attachments is 14% better than pure guessing (of noun attachments) which would lead to 61% correct attachments. The negative side is, that these accuracy and coverage values are significantly lower than our results from using a medium size (6 million words) local accessible corpus with shallow corpus annotation. Frequencies derived from that corpus led to 82% accuracy for a coverage of 90%. Obviously, using the NEAR operator introduces too much noise into the frequency counts.

Many of the unresolved test cases involved proper names (person names, company names, product names) as either N1 or N2. Triples with names are likely to result in zero frequencies for WWW queries. One way of avoiding this bottleneck is proper name classification and querying for well-known (i.e. frequently used) representatives of the classes. As an example, we might exchange the (N1,P,N2) triple *Computer von Robertson Stephens & Co.* by *Computer von IBM*, query for the latter and use the cooccurrence value for the former. Of course, it would be even better if we could query the search engine for *Computer von <company>* which matched any company name. But such operators are currently not available in web search engines.

One option to escape this dilemma is the implementation of a linguistic search engine that would index the web in the same manner as AltaVista or Google but offer linguistic operators for query formulation. Obviously, any constraint to increase the query precision will reduce the frequency counts and may thus lead to sparse data. The linguistic search engine will therefore have to allow for semantic word classes to counterbalance this problem. We will get back to this in section 3.

Another option is to automatically process (a number of) the web pages that are retrieved by querying a standard WWW search engine. For the purpose of PP attachment, one could think of the following procedure.

1. One queries the search engine for all German documents that contain the noun N1 (or the verb V), possibly restricted to a subject domain.
2. A fixed number of the retrieved pages are automatically loaded. Let us assume the thousand top-ranked pages are loaded via the URLs provided by the search engine.
3. From these documents all sentences that contain the search word are extracted (which requires sentence boundary recognition).
4. The extracted sentences are compiled and subjected to corpus processing (with proper name recognition, part-of-speech tagging, lemmatization etc.) leading to an annotated corpus.
5. The annotated corpus can then be used for the computation of unigram, bigram and triple frequencies.

### 2.3. Semantics

Gathering semantic information from the web is an ambitious task. But it is also the most rewarding with regard to practical applications. If the semantic units in a web site can be
classified, then retrieval will be much more versatile and precise. In some sense the proper
name classification described in section 2.1 can be seen as basic work in semantics.

(Agirre et al. 2000) go beyond this and describe an approach to enriching the WordNet
ontology using the WWW. They show that it is possible to automatically create lists of words
that are topically related to a WordNet concept. If a word has multiple senses in WordNet, it
will be accompanied by synonyms and other related words for every sense. Agirre et al. query
the web for documents exemplifying every sense by using these co-words. The query is
composed by using the disjunction of the word in question and its co-words and by the
exclusion of all co-words of the competing senses (via the NOT operator). The documents
thus retrieved are locally processed and searched for terms that appear more frequently than
expected using the $X^2$ function. They evaluated the resulting topic signatures (lists of related
words) by successfully employing them in word sense disambiguation.

Moreover they use the topic signatures to cluster word senses. For example, they were able to
determine that some WordNet senses of *boy* are closer to each other than others. Their method
could be used to reduce WordNet senses to a desired grain size.

2.4. Translation

Translators have found the web a most useful tool for looking up how a certain word or
phrase is used. Since queries to standard search engines allow for restrictions to a particular
language and, via the URL domain, also to a particular country, it has become easy to obtain
usage information which was buried in books and papers (or local databases at best) prior to
the advent of the web. In addition to simply browsing through usage examples one may
exploit the frequency information. We will summarize two other examples of how a translator
may profit from the web:

2.4.1. Translating Compound Nouns

(Grefenstette 1999) has shown that WWW frequencies can be used to find the correct
translation of German compounds if the possible translations of their parts are known. He
extracted German compounds from a machine-readable German-English dictionary. Every
compound had to be decomposable into two German words found in the dictionary and its
English translation had to consist of two words. Based on the compound segments more than
one translation was possible. For example, the German noun *Aktienkurs* (share price) can be
segmented into *Aktie* (share, stock) and *Kurs* (course, price, rate) both of which have multiple
possible translations. By generating all possible translations (*share course, share price, share
rate, stock course, ...*) and submitting them to AltaVista queries, Grefenstette obtains WWW
frequencies for all possible translations. He tested the hypothesis that the most frequent
translation is the correct one.

He extracted 724 German compounds according to the above criteria and found that his
method predicted the correct translation for 631 of these compounds (87%). This is an
impressive result given the simplicity of the method.

2.4.2. Parallel Texts in the Web

Translation memory systems have become an indispensable tool for translators in recent
years. They store parallel texts in a database and can retrieve a unit (typically a sentence) with
its previous translation equivalents when it needs to be translated again. Such systems come
to their full use when a database of the correct subject domain and text type is already stored.
They are of no help when few or no entries have been made.
But often previous translations exist and are published in the web. The task is to find these text pairs, judge their translation quality, download and align them, and store them into a translation memory. Furthermore, parallel texts can be used for statistical machine translation. (Resnik 1999) therefore developed a method to automatically find parallel texts in the web. In a first step he used a query to AltaVista by asking for parent pages containing the string "English" within a fixed distance of "German" in anchor text. This generated many good pairs of pages such as those reading "Click here for English version" and "Click here for German version", but of course also many bad pairs.

Therefore he added a filtering step that compares the structural properties of the candidate documents. He exploited the fact that web pages in parallel translations are very similarly structured in terms of HTML mark-up and length of text. A statistical language identification system determines whether the found documents are in the suspected language. 179 automatically found pairs were subjected to human judgement. Resnik reports that 92% of the pairs considered as good by his system were also judged good by the two human experts.

In a second experiment he increased the recall by not only looking for parent pages but also for sibling pages, i.e. pages with a link to their translated counterpart. For English-French he thus obtained more than 16,000 pairs. Further research is required to see how many of these pairs are suitable for entries in a translation memory database.

### 2.5. Diachronic Change

In addition to accessing today's language, the web may also be used to observe language change over time. Some search engines allow restricting a query to documents of a certain time span. And although the web is young and old documents are often removed, there are first examples in which language change is documented.

Let us look at the two competing words for *mobile phone* in Swiss German. When the Swiss telephone company Swisscom first launched mobile phones, they called them *Natel*. At about the same time mobile phones in Germany were introduced as *Handy*. Ever since, these two words are competing for prominence in Switzerland. And our hypothesis was that *Handy* has become more frequently used because of the open telecom market.

We therefore checked the frequency of occurrence in the web before and after January 1st 2000. We used the Hotbot search engine since it allows this kind of time span search. We queried for all inflected forms of the competing words and restricted the search to German documents in the.ch domain. It turned out that before the year 2000 the number of documents found for *Natel* was about twice the number for *Handy*. For the period after January 2000 the frequency for both is about the same. This is clear evidence that the use of *Handy* is catching up, and it will be interesting to follow whether it will completely wipe out *Natel* in the future.

### 3. Towards a Corpus Query Tool for the Web

Current access to the web is limited in that we can only retrieve information through the bottleneck of search engines. We thus have to live with the operators and options they offer. But these search engines are not tuned to the needs of linguists. For instance, it is not possible to query for documents that contain the English word *can* as a noun. Therefore we call for the development a linguistic search engine that is designed after the most powerful corpus query tools. Of course their power depends on what kind of linguistic processing they can apply.
We checked the query languages of the Cosmas query tool at the "Institut für deutsche Sprache"\(^2\) and the corpus query language for the British National Corpus (BNC). The Cosmas query tool knows Boolean operators, distance operators, wildcards and form operators (“all inflected forms” and “ignore capitalization”). In particular the distance operators for words, sentences and paragraphs are positive. But in comparison the query language for the BNC is much more powerful. Since the texts are part-of-speech tagged and marked up with SGML, this information can be accessed through the queries.

Let us go through the requirements for an ideal corpus query tool and the operators that it should comprise. The operators typically work on words but sometimes it is desirable to access smaller units (letters, syllables, morphemes) or larger units (phrases, clauses, sentences).

1. **Boolean operators (AND, OR, NOT):** combining search units by logical operators. These operators are the most basic and they are available in most search engines (on the word level). Combining Boolean operators with some of the other operators may slow down retrieval time and is therefore often restricted.

2. **Regular expressions:** widening the search by substituting single search units (letters, words) or groups thereof by special symbols. The most common are the Kleene star (*) substituting a sequence of units and the question mark (?) substituting exactly one unit.

3. **Distance operators:** restricting the search to a specific distance between search units (word, phrase, clause, sentence, paragraph, chapter, document; e.g. find `bank` within 2 sentences of `money`) often combined with a direction (to the left or right; e.g. find `with` within 2 words to the left of `love`).

4. **Environment operators:** restricting the search to a specific section of a document (e.g. header, abstract, quote, list item, bibliographic list; e.g. find `Bill Gates` in headers).

5. **Form operators:** widening the search by abstracting from a specific form. They include the capitalization operator (ignore upper and lower case) and the inflection operator (use all inflected forms).

6. **Syntactic class operators:** restricting the search to a certain part-of-speech (e.g. `with` followed by any noun), or phrase (e.g. an accusative NP followed by a PP).

7. **Subject domain operators:** restricting the search to a specific subject domain (e.g. linguistics, computer science, chemistry etc.).

8. **Text type operators:** restricting the search to a specific text type (e.g. newspaper articles, technical manuals, research reports).

9. **Semantic class operators:** restricting the search to e.g. proper name classes (person, location, organization, product), temporal and local phrases; causal relations, or certain word senses (e.g. find `bank` in the sense of `financial institution`); synonym and hypernym searches; definitory contexts.

A search engine stores information about a document at the time of indexing (e.g. its location, date and language). This speeds up online retrieval but requires offline processing of the documents. And some of the operators rely on information that is difficult and costly to compute over large amounts of text.

Therefore it might be advisable to use offline processing as suggested by (Corley et al. 2001). They describe the Gsearch system which allows finding syntactic structure in unparsed

\(^2\) See [http://corpora.ids-mannheim.de/~cosmas/](http://corpora.ids-mannheim.de/~cosmas/) to query the corpora at the "Institut für deutsche Sprache".
corpora. The idea is that the Gsearch user provides the system with contextfree grammar rules that describe constituents for the syntactic phenomenon under investigation. Gsearch's bottom up chart parser processes every sentence and checks whether it can apply the grammar rules and whether they lead to the search goal. For example, a grammar might specify rules for NPs and PPs. Then the search goal might be to find sentences which contain a sequence of a verb, an NP and a PP for the investigation of PP attachments. Gsearch is not intended for accurate unsupervised parsing but as a tool for corpus linguists. Since its grammar is exchangeable, it can be adapted to the specific needs of the linguist.

4. Conclusion

We have shown that the web offers a multitude of opportunities for corpus research. Access to this corpus is ubiquitous and its size exceeds all previous corpora. Currently the access is limited by search engines that are not tuned to linguistic needs. We therefore propose to use these search engines only for the preselection of documents and add linguistic postprocessing. A better but much more complex solution is the implementation of a special purpose linguistic search engine.

(Kilgarriff 2001) has sketched an intermediate solution of organizing a controlled corpus distributed over various web servers. This would escape some of the problems that we currently encounter in web-based studies. The web is a very heterogeneous collection of documents. Many documents do not contain running text but rather lists, indexes or tables. If these are not filtered out, they might spoil collocations or other types of cooccurrence statistics. In addition, the web is not a balanced corpus, neither with respect to text types nor with respect to subject areas. (Agirre et al. 2000) found that sex related web pages strongly influenced their word sense disambiguation experiments for the word boy.

When working with low occurrence frequencies, one also has to beware of negative examples. For our study on German prepositions we checked which of them have corresponding pronominal adverbs. This test is used for determining whether the preposition can introduce a prepositional object, since pronominal adverbs are proforms for such objects. It is taken for granted that the preposition ohne (without) is an exception to this rule. It can introduce a prepositional object, but it does not form a pronominal adverb. But a Google query for the questionable pronominal adverb form darohne results in 14 hits. At first sight this seems to contradict the linguistic assumptions. But a closer look reveals that many of these hits lead to literary texts written more than a century ago while others discuss the fact that the form darohne does not exist anymore.

We believe that harvesting the web for linguistic purposes has only begun and will develop into an important branch for Computational Linguistics. We envision that future NLP systems will query the web whenever they need information that is not locally accessible. We think of parsers that will query the web for the resolution of structural or sense ambiguities. Or of MT systems that perform automatic lexical acquisition over the web to fill their lexicon gaps before translating a text.

One of the main arguments against using the web is its constant change. A result computed today may not be exactly reproducible tomorrow. But, as (Kilgarriff 2001) notes, this is same for the water in any river and nobody will conclude that investigating water molecules is therefore senseless. We will all have to learn to fish in the waters of the web.
5. References


Resnik, Philip 1999. Mining the web for bilingual text. – Proc. of 37th Meeting of the ACL. Maryland. 527-534.
