Large linguistically-processed Web corpora for multiple languages

Marco Baroni
SSLMIT
University of Bologna
Italy
`baroni@sslmit.unibo.it`

Adam Kilgarriff
Lexical Computing Ltd. and
University of Sussex
Brighton, UK
`adam@lexmasterclass.com`

Abstract

The Web contains vast amounts of linguistic data. One key issue for linguists and language technologists is how to access it. Commercial search engines give highly compromised access. An alternative is to crawl the Web ourselves, which also allows us to remove duplicates and near-duplicates, navigational material, and a range of other kinds of non-linguistic matter. We can also tokenize, lemmatise and part-of-speech tag the corpus, and load the data into a corpus query tool which supports sophisticated linguistic queries. We have now done this for German and Italian, with corpus sizes of over 1 billion words in each case. We provide Web access to the corpora in our query tool, the Sketch Engine.

1 Introduction

The Web contains vast amounts of linguistic data for many languages (Kilgarriff and Grefenstette, 2003). One key issue for linguists and language technologists is how to access it. The drawbacks of using commercial search engines are presented in Kilgarriff (2003). An alternative is to crawl the Web ourselves.[1] We have done this for two languages, German and Italian, and here we report on the pipeline of processes which give us reasonably well-behaved, 'clean' corpora for each language.

We use the German corpus (which was developed first) as our example throughout. The procedure was carried on a server running RH Fedora Core 3 with 4 GB RAM, Dual Xeon 4.3 GHz CPUs and about 2.5 TB hard disk space. We are making the tools we develop as part of the project freely available,[2] in the hope of stimulating public sharing of resources and know-how.

2 Crawl seeding and crawling

We would like a "balanced" resource, containing a range of types of text corresponding, to some degree, to the mix of texts we find in designed linguistic corpora (Atkins et al., 1992), though also including text types found on the Web which were not anticipated in linguists' corpus design discussions. We do not want a "blind" sample dominated by product listings, catalogues and computer scientists' bulletin boards. Our pragmatic solution is to query Google through its API service for random pairs of randomly selected content words in the target language. In preliminary experimentation, we found that single word queries yielded many inappropriate pages (dictionary definitions of the word, top pages of companies with the word in their name), whereas combining more than two words retrieved pages with lists of words, rather than collected text.

Ueyama (2006) showed how queries for words sampled from traditional written sources such as newspaper text and published essays tend to yield "public sphere" pages (online newspaper, government and academic sites), whereas basic vocabulary/everyday life words tend to yield "personal" pages (blogs, bulletin boards). Since we wanted both types, we obtained seed URLs with queries

---

[1] Another Web access option is Alexa (`http://pages.alexa.com/company/index.html`), who allow the user (for a modest fee) to access their cached Web directly. Using Alexa would mean one did not need to crawl; however in our experience, crawling, given free software like Heritrix, is not the bottleneck. The point at which input is required is the filtering out of non-linguistic material.

[2] `http://sslmitdev-online.sslmit.unibo.it/wac/wac.php`

for words from both kinds of sources. For German, we sampled 2000 mid-frequency words from a corpus of the *Süddeutsche Zeitung* newspaper and paired them randomly. Then, we found a basic vocabulary list for German learners,[3] removed function words and particles and built 653 random pairs. We queried Google via its API retrieving maximally 10 pages for each pair. We then collapsed the URL list, insuring maximal sparseness by keeping only one (randomly selected) URL for each domain, leaving a list of 8626 seed URLs. They were fed to the crawler.

The crawls are performed using the *Heritrix* crawler,[4] with a multi-threaded breadth-first crawling strategy. The crawl is limited to pages whose URL does not end in one of several suffixes that cue non-html data (`.pdf`, `.jpeg`, etc.)[5] For German, the crawl is limited to sites from the `.de` and `.at` domains. Heritrix default crawling options are not modified in any other respect. We let the German crawl run for ten days, retrieving gzipped archives (the Heritrix output format) of about 85GB.

3 Filtering

We undertake some post-processing on the basis of the Heritrix logs. We identify documents of mime type `text/html` and size between 5 and 200KB. As observed by Fletcher (2004) very small documents tend to contain little genuine text (5KB counts as "very small" because of the html code overhead) and very large documents tend to be lists of various sorts, such as library indices, store catalogues, etc. The logs also contain sha-1 fingerprints, allowing us to identify perfect duplicates. After inspecting some of the duplicated documents (about 50 pairs), we decided for a drastic policy: if a document has at least one duplicate, we discard not only the duplicate(s) but also the document itself. We observed that, typically, such documents came from the same site and were warning messages, copyright statements and similar, of limited or no linguistic interest. While the strategy may lose some content, one of our general principles is that, given how vast the Web is, we can afford to privilege precision over recall.

All the documents that passed the pre-filtering

stage are run through a perl program that performs 1) boilerplate stripping 2) function word filtering 3) porn filtering.

Boilerplate stripping

By "boilerplate" we mean all those components of Web pages which are the same across many pages. We include stripping out HTML markup, javascript and other non-linguistic material in this phase. We aimed to identify and remove sections of a document that contain link lists, navigational information, fixed notices, and other sections poor in human-produced connected text. For purposes of corpus construction, boilerplate removal is critical as it will distort statistics collected from the corpus.[6] We adopted the heuristic used in the Hyppia project BTE tool,[7]: content-rich sections of a page will have a low html tag density, whereas boilerplate is accompanied by a wealth of html (because of special formatting, newlines, links, etc.) The method is based on general properties of Web documents, so is relatively independent of language and crawling strategy.

Function word and pornography filtering

Connected text in sentences reliably contains a high proportion of function words (Baroni, to appear), so, if a page does not meet this criterion we reject it. The German function word list contains 124 terms. We require that a minimum of 10 types and 30 tokens appear in a page, with a ratio of function words to total words of at least one quarter. The filter also works as a simple language identifier.[8]

Finally, we use a stop list of words likely to occur in pornographic Web pages, not out of prudery, but because they tend to contain randomly generated text, long keyword lists and other linguistically problematic elements. We filter out documents that have at least three types or ten tokens from a list of words highly used in pornography. The list was derived from the analysis of pornographic pages harvested in a previous crawl. This is not entirely satisfactory, since some of the words

---

[3] `http://mypage.bluewin.ch/a-z/cusipage/`

[4] `http://crawler.archive.org`

[5] Further work should evaluate pros and cons of retrieving documents in other formats, e.g., Adobe *pdf*.

[6] We note that this phase currently removes the links from the text, so we can no longer explore the graph structure of the dataset. In future we may retain link structure, to support research into the relation between it and linguistic characteristics.

[7] `http://www.smi.ucd.ie/hyppia/`

[8] Of course, these simple methods will not filter out all machine-generated text (typically produced as part of search engine ranking scams or for other shady purposes); sometimes this appears to have been generated with a bigram language model, and thus identifying it with automated techniques is far from trivial.

in the list, taken in isolation, are wholly innocent (*fat, girls, tongue,* etc.) We shall revisit the strategy in due course.

This filtering took 5 days and resulted in a version of the corpus containing 4.86M documents for a total of 20GB of uncompressed data.

4   Near-duplicate detection

We use a simplified version of the "shingling" algorithm (Broder et al., 1997). For each document, after removing all function words, we take fingerprints of a fixed number s of randomly selected n-grams; then, for each pair of documents, we count the number of shared n-grams, which can be seen as an unbiased estimate of the overlap between the two documents (Broder et al., 1997; Chakrabarti, 2002). We look for pairs of documents sharing more than t n-grams, and we discard one of the two.

After preliminary experimentation, we chose to extract 25 5-grams from each document, and to treat as near-duplicates documents that shared at least two of these 5-grams. Near-duplicate spotting on the German corpus took about 4 days. 2,466,271 near-duplicates were removed. The corpus size decreased to 13GB. Most of the processing time was spent in extracting the n-grams and adding the corresponding fingerprints to the database (which could be parallelized).

5   Part-of-speech tagging/lemmatization and post-annotation cleaning

We performed German part-of-speech tagging and lemmatization with TreeTagger.[9] Annotation took 5 days. The resulting corpus contains 2.13B words, or 34GB of data including annotation.

After inspecting various documents from the annotated corpus, we decided to perform a further round of cleaning. There are two reasons for this: first, we can exploit the annotation to find other anomalous documents, through observing where the distribution of parts-of-speech tags is very unusual and thus not likely to contain connected text. Second, the TreeTagger was not trained on Web data, and thus its performance on texts that are heavy on Web-like usage (e.g., texts all in lower-case, colloquial forms of inflected verbs, etc.) is dismal. While a better solution to this second problem would be to re-train the tagger on Web

data (ultimately, the documents displaying the second problem might be among the most interesting ones to have in the corpus!), for now we try to identify the most problematic documents through automated criteria and discard them. The cues we used included the number of words not recognised by the lemmatizer; the proportion of words with upper-case initial letters; proportion of nouns, and proportion of sentence markers.

After this further processing step, the corpus contains 1,870,259 documents from 10818 different domains, and its final size is 1.71 billion tokens (26GB of data, with annotation). The final size of the Italian corpus is 1,875,337 documents and about 1.9 billion tokens.

6   Indexing and Web user interface

We believe that matters of efficient indexing and user friendly interfacing will be crucial to the success of our initiative, both because many linguists will lack the relevant technical skills to write their own corpus-access routines, and because we shall not publicly distribute the corpora for copyright reasons; an advanced interface that allows linguists to do actual research on the corpus (including the possibility of saving settings and results across sessions) will allow us to make the corpus widely available while keeping it on our servers.[10] We are using the Sketch Engine,[11] a corpus query tool which has been widely used in lexicography and which supports queries combining regular expressions and boolean operators over words, lemmas and part-of-speech tags.

7   Comparison with other corpora

We would like to compare the German Web corpus to an existing "balanced" corpus of German attempting to represent a broad range of genres and topics. Unfortunately, as far as we know no resource of this sort is publicly available (which is one of the reasons why we are interested in developing the German Web corpus in the first instance.) Instead, we use a corpus of newswire articles from the Austria Presse Agentur (APA, kindly provided to us by ÖFAI) as our reference

---

[9]http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger

[10]The legal situation is of course complex. We consider that our case is equivalent to that of other search engines, and that offering linguistically-encoded snippets of pages to researchers does not go beyond the "fair use" terms routinely invoked by search engine companies in relation to Web page caching.

[11]http://www.sketchengine.co.uk/

WEB		APA	
ich	hier	APA	NATO
dass	wir	Schluß	EU
und	man	Prozent	Forts
sie	nicht	Mill	AFP
ist	das	MRD	Dollar
oder	sind	Wien	Reuters
kann	so	Kosovo	Dienstag
du	mir	DPA	Mittwoch
wenn	ein	US	Donnerstag
was	da	am	sei

Table 1: Typical Web and APA words

point. This corpus contains 28M tokens, and, despite its uniformity in terms of genre and restricted thematic range, it has been successfully employed as a general-purpose German corpus in many projects. After basic regular-expression-based normalization and filtering, the APA contains about 500K word types, the Web corpus about 7.4M. There is a large overlap among the 30 most frequent words in both corpora: 24 out of 30 words are shared. The non-overlapping words occurring in the Web top 30 only are function words: *sie* 'she', *ich* 'I', *werden* 'become/be', *oder* 'or', *sind* 'are', *er* 'he'. The words only in the APA list show a bias towards newswire-specific vocabulary (*APA*, *Prozent* 'percent', *Schluß* 'closure') and temporal expressions that are also typical of newswires (*am* 'at', *um* 'on the', *nach* 'after').

Of the 232,322 hapaxes (words occurring only once) in the APA corpus, 170,328 (73%) occur in the Web corpus as well.[12] 89% of these APA hapaxes occur more than once in the Web corpus, suggesting how the Web data will help address data sparseness issues.

Adopting the methodology of Sharoff (2006), we then extracted the 20 words most characteristics of the Web corpus vs. APA and vice versa, based on the log-likelihood ratio association measure. Results are presented in Table 1. The APA corpus has a strong bias towards newswire parlance (acronyms and named entities, temporal expressions, financial terms, toponyms), whereas the terms that come out as most typical of the Web corpus are function words that are not strongly connected with any particular topic or genre. Several of these top-ranked function words mark first and second person forms (*ich*, *du*, *wir*, *mir*).

This preliminary comparison both functioned as a "sanity check", showing that there is considerable overlap between our corpus and a smaller corpus used in previous research, and suggested that the Web corpus has more a higher proportion of interpersonal material.

8 Conclusion

We have developed very large corpora from the Web for German and Italian (with other languages to follow). We have filtered and cleaned the text so that the obvious problems with using the Web as a corpus for linguistic research do not hold. Preliminary evidence suggests the 'balance' of our German corpus compares favourably with that of a newswire corpus (though of course any such claim begs a number of open research questions about corpus comparability). We have lemmatised and part-of-speech-tagged the data and loaded it into a corpus query tool supporting sophisticated linguistic queries, and made it available to all.

References

B. Atkins, J. Clear, and N. Ostler. 1992. Corpus design criteria. *Literary and Linguistic Computing*, 7:1–16.

M. Baroni. to appear. Distributions in text. In A. Lüdeling and M. Kytö, editors, *Corpus linguistics: An international handbook*. Mouton de Gruyter, Berlin.

A. Broder, S. Glassman, M. Manasse, and G. Zweig. 1997. Syntactic clustering of the Web. In *Proc. Sixth International World-Wide Web Conference*.

S. Chakrabarti. 2002. *Mining the Web: Discovering knowledge from hypertext data*. Morgan Kaufmann, San Francisco.

W. Fletcher. 2004. Making the web more useful as a source for linguistic corpora. In U. Connor and T. Upton, editors, *Corpus Linguistics in North America 2002*.

A. Kilgarriff and G. Grefenstette. 2003. Introduction to the special issue on the Web as corpus. *Computational Linguistics*, 29(3):333–347.

A. Kilgarriff. 2003. Linguistic search engine. In K. Simov, editor, *Proc. SPROLAC Workshop*, Lancaster.

S. Sharoff. 2006. Creating general-purpose corpora using automated search engine queries. In M. Baroni and S. Bernardini, editors, *WaCky! Working papers on the Web as Corpus*. Gedit, Bologna.

M. Ueyama. 2006. Creation of general-purpose Japanese Web corpora with different search engine query strategies. In M. Baroni and S. Bernardini, editors, *WaCky! Working papers on the Web as Corpus*. Gedit, Bologna.

---

[12]Less than 1% of the Web corpus hapaxes are attested in the APA corpus.