KTH Stockholm
Skolan för Datavetenskap och Kommunikation
Numerisk analys och datalogi
October 26, 2005
Course: *2D1418 Språkteknologi*
Autumn Term 2005
Course Instructor: Ola Knutsson

ConcorDance

A Simple Concordance Interface for Search Engines

Hendrik Buschmeier
hbuschme@TechFak.Uni-Bielefeld.DE

Oberer Steinbrink 8
32657 Lemgo
Germany

1 Introduction

> con·cor·dance |kənˌkɔːd(ə)ns|
> noun
>
> 1. an alphabetical list of the words (esp. the important ones) present in a text, usually with citations of the passages concerned [...]
>
> ORIGIN late Middle English : from Old French, from medieval Latin **concordantia**, from **concordant- 'being of one mind'**.

The citation above is a part of the Oxford American Dictionaries' entry for the word *concordance*. It refers to a way concordances were used when no full text search was available due to the lack of electronic text resources. The index of some books did not consist of simple keywords, but of *keywords in context* (KWIC) which allowed the readers a more precise search.

As it was very laborious to build a concordance list for books, they were only available for important works such as the Bible or Shakespeare's plays, later also for technical manuals and other long texts.

Today concordances—or rather concordance analysis programs—are a useful tool for linguists (especially in the field of corpus linguistics) where they are used to list all occurrences of a word or word form in its context. In connection with statistical analysis it is then possible to retrieve e.g. information about collocations and use these to disambiguate words or to track how a word is used in a language or to observe how the use of a word changes over time. (Müller, 2002, p. 370)

But a concordance program could also be useful for language learners. If one is not sure about the right preposition or the validity of an idiomatic expression it comes in handy to use "empirical methods" (e.g. query a search engine and compare the number of available results matching different phrases that are considered).

Of course it is desirable to be able to query as large and as current corpora as possible. Fortunately the largest corpora of the world are available free of charge over the Internet: the indices of the search engines.

In this assignment the author presents his concordance program called ConcorDance[1] which makes use of these resources.

2 About ConcorDance

ConcorDance is a simple concordance interface to Internet search engines (at the moment Google[2] and Yahoo![3]) which is designed as a web application, so that users can utilise it with every web browser.

[1] `http://buschmeier.org/bh/study/ccd/`
[2] `http://www.google.com/`
[3] `http://www.yahoo.com/`

In the *basic mode* the user can specify one or more words (i.e. a phrase) and choose one of the available search engines. After sending the "query" the program returns a maximum of 10 concordances in which the given word/phrase occurrs. The results are formatted in a way that the query words are printed in bold and centred. Figure 1 is an illustration for a concordance of the query word *incident* formatted in that way (for different reasons the resulting sentences/concordances in this example are taken from the Oxford American Dictionaries and not from a search engine).

One person was stabbed in the	**incident**	.
The winter passed without	**incident**	.
The changes	**incident**	to economic development.
When an ion beam is	**incident**	on a surface.

Figure 1: *Concordances for the word "incident".*

In the *advanced mode* the user is allowed to formulate more precise queries. Again he might give one or more words as a query but additionally he can restrict the search to a language (28 at the moment), a country (25 at the moment) or a hostname (e.g. `org`, `de`, `se`, `buschmeier.org`, `www.heise.de`, `www.nada.kth.se`, ...). All combinations of these restrictions are allowed, too. However at the moment it is not possible to combine several languages, countries or pages (e.g. all results in German and/or Swedish, all results form the USA and/or from the UK, ...). Furthermore the *advanced mode* allows to request more concordances (a maximum of 100 can be requested).

Due to its state-of-the-art user interface ConcorDance should be useable for everyone with some experience of search engine usage.

3 Design and Implementation

ConcorDance is implemented in the object-oriented and dynamically typed programming language Python and works via the CGI (*common gateway interface*) of the webserver. Queries from the user are passed to the selected search engine via the search engine's API (more on that later). The results are then checked for relevance (i.e. is the search word/phrase really part of the result? – which is sometimes not the case) and after that a simple linguistic extraction is done on the relevant results (i.e. if possible whole sentences are extracted).

ConcorDance's basis is the abstract class `SearchEngineWrapper` which defines the interface to the search engines (figure 2). Therefore it is quite easy to extend the program with new search engines (or even other data sources e.g. local file system search, search in text files etc.).

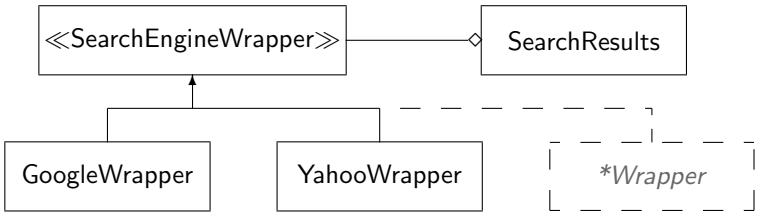

Figure 2: *A Simplified UML class diagram for the "plugin-interface".*

To add a new search source it is only necessary to wrap its API in a class which inherits `SearchEngineWrapper` and converts the returned results into a list of objects of the type `SearchResult`. Furthermore the new search source must be integrated into the web interface, which means that it has to be selectable for the user.

The design of the class `SearchEngineWrapper` goes with the intersection of the possibilities of the two search engine wrapper reference implementations `GoogleWrapper` and `YahooWrapper`. This should be general enough for most applications.

The Google API (Google, 2005), (Calishain and Dornfest, 2004, Chapter 9) is used through PyGoogle[4] because it would be rather complex to implement its SOAP interface from scratch. The relevant parts of the Yahoo! Search Web API (Yahoo!, 2005) on the other hand could be implemented for this project as Yahoo! uses a simple REST (Representational State Transfer) API.

ConcorDance extracts the concordances from the snippets the search engine returns with the results. A problem that sometimes occurs is that the snippet does *not* contain the phrase or the word the user requested concordances for. This is due to the fact that search engines also show matches where the requested word was only found in the title or in the URL. According to Calishain and Dornfest (2004, Chapter 8) Google even ranks those results better, where the search phrase occurs in the title or URL. Such results are simply not shown to the user. Therefore it can happen that the user requested more results than he gets, although there may be more in the search engine's index.

A task which ConcorDance has to accomplish is an elementary linguistic extraction. The best would be if the the results were whole sentences or phrases. Unfortunately the returned snippets are often very short. There are even ellipses in the sentences, which are indicated by "...".

As a matter of this ConcorDance uses a heuristic method to find sentence boundaries. First it splits the snippet into two parts. The words before the word or phrase the user requested and the words after it. Then these two parts are searched for punctuation marks (i.e. ".", "?", "!", ":" and ";") and cut down after them. This method has proven to be sufficient.

[4] `http://pygoogle.sourceforge.net/`

4 Evaluation

This section's intention is to present the problems with the concordances found by ConcorDance and the problems of ConcorDance in general.

1. Especially if one requests single words it is most likely to get many results which are of little or even no use in concordance analysis. Queries for the word "independent" for example return a lot of results which are related to a British newspaper called *The Independent*.

2. If the search query occurs several times on the same website, it also occurs several times in the snippet the search engine provides. Currently ConcorDance only considers the first match of the word in the snippet.

3. ConcorDance allows a maximum of 100 results for the requested word or phrase to be returned. This is on the one hand a deliberate limitation as Google only allows 100 queries per day and user and on the other hand it is not possible with Google to access results with an offset greater than 1000. Both of these constrains are similar with Yahoo! (but they are more generous).

   Additionally it is rather time consuming to request a lot of results (at least through Google). This problem seems to be located in PyGoogle but may also be a problem of the Google API.

Some of these problems are solvable but the first problem is a big one. A lot of words are used as trademarks or other corporate slogans and therefore often occur in this context – especially in the first 100 results.

In conclusion it can be said that ConcorDance technically works as it was intended but the possibility to access the largest corpus of the world as mentioned as a goal in section 1 of this assignment is quasi non-existent. It is only possible to scratch the surface of this huge resources – which is mainly coined by commercial influences.

5 References

Monographs

Calishain, T., Dornfest, R. (2004). *Google Hacks. Tips & Tools for Smarter Searching.*
2. ed., Sebastopol [et al.], O'Reilly.

Müller, H. M. (2002). *Arbeitsbuch Linguistik.*
1. ed., Paderborn [et al.], Schöningh.

Sources on the Internet

Google (2005). *Google Web APIs Reference.*
Retrieved October 26, 2005, from:
`http://www.google.com/apis/reference.html`

Yahoo! (2005). *Web Search Documentation for Yahoo! Search Web Services.* Retrieved October 26, 2005, from:
`http://developer.yahoo.net/search/web/V1/webSearch.html`

6 About This Document

This document and the web application ConcorDance are the author's results of the home assignment in the course 2D1418 Språkteknologi given by Ola Knutsson et al. at KTH Stockholm in the first study period of the autumn term 2005.

The task was to "build an interface to a search engine which returns the search phrase with concondances".

This document was composed in LaTeX. It was last revised on October 26, 2005.