Viewing Web Search Engines as Corpus Query Systems

Alexandros Ntoulas, Sofia Stamou, Manolis Tzagarakis,
Ioanna Tsakou, Dimitris Christodoulakis
{ntoulas, stamou, tsakou, tzagara, dxri}@cti.gr

Computer Engineering & Informatics Department
And Computer Technology Institute
University Of Patras
Building B', 26 500, Rion,
Patras, Greece

Abstract

In this paper we examine whether a web search engine can function as a corpus management system, by describing the similarities between corpus query systems and web search engines and we present a prototype implementation of a corpus management system incorporated in a Greek web search engine. We target our research towards the examination of common features that exist among query languages adopted in each of the aforementioned approaches and we compare the query analysis and searching modules, each system provides. We also report on distinct characteristics among corpus query systems and web search engines with emphasis given on their operation, the data type and structure they handle, and the storage techniques that they employ.

Keywords: Corpus Query Systems, Corpus Management, Web Search Engines, and Information Retrieval

1 Introduction

In this paper we present a prototype of a corpus management system, which handles dynamic collections of data. Traditional corpus query systems are targeted towards the acquisition of lexical information stemming from static lexical resources. Such systems are usually oriented towards a pre-specified user group (i.e. linguists, lexicographers etc.) and function mainly as assistant utilities to the work conducted within the framework of other specific applications, ranging from dictionary construction to linguistic research.

The aforementioned systems have some features in common. Namely, they all provide a query interface and/or a query language to the users, thus enabling them to express their search requests in a flexible way. Moreover, they all incorporate a query analysis and searching module, which undertakes the task of interpreting the user's query, retrieving the search results and presenting them to the users. Eventually, they comprise of a storage layer where the actual textual information is kept. Such systems though, do not vary greatly from a web search engine since they both handle large volumes of textual data and provide searching facilities.

In this work we report the similarities between corpus query systems and web search engines and we present a prototype implementation of a corpus query system incorporated in a Greek web search engine. More specifically, we examine whether the query languages adopted in each of the above approaches provide equivalent searching

facilities and we investigate in more detail whether the information can be presented in a similar way.

Apart from the similarities between web search engines and corpus query systems, the former have also some worth studying distinct characteristics. Firstly, they operate independently of the volume of data and they can also handle dynamic collections of texts, which are usually automatically acquired and continuously updated. Furthermore, they can handle data independent of language and structure. In particular, it is possible to process text in any language and any acceptable format (e.g. HTML, XML etc.). It is also feasible for a search engine to store documents marked with domain labels and therefore a user is capable of searching only in the desired domains. Finally, the operating environment of a web search engine is widely and easily accessible through the Internet.

In the following section we provide a detailed description of the purpose and architecture of a corpus query system and we continue with a comparison among such systems with search engines (3). Following on from this, a detailed description of the approach we adopted in this study is provided along with a presentation of the search engine we used to test our hypothesis (4). In the remaining sections of the paper we present the concordance module we incorporated into the search engine (5) and we describe its functionality. We summarize (6) by providing some general conclusions of the usefulness of such a system, along with some directions for future work.

2 Purpose and Architecture of a Corpus Query System

A Corpus Query System roughly consists of two layers: a Logical Access Layer, which is independent of the data access methods and storage details, and a Physical Access Layer, which is a data oriented interface to the actual data (Christ 1994). The Logical Layer handles the translation of the user's query into a format that is readable by the Physical Layer. The latter in its turn takes up the task of acquiring all the necessary data requested by the Logical Layer from the actual stored data. The stored data can be in many different forms, varying from flat text files and databases to dynamic knowledge sources (e.g WordNet (Miller et al. 1993)), but the most usual way of organizing this kind of textual data is using an index.

A Corpus Query System is used mainly for linguistic work such as lexicon construction or grammar development, language teaching or theoretical linguistic research. The main functions that help this kind of work include the presentation of concordance lines to the end user, the presentation of collocation data etc. (Sinclair 1991).

3 Similarities between Web Search Engines and Typical Corpus Query Systems

One especially popular way of sharing information in a decentralized way is through the Internet, where information sources containing non-homogenous data are stored (Mladenic 1999). A commonly addressed issue, widely conducted with the usage of search engines, is providing help to users in browsing the Web. Typically, web search engines are targeted towards the acquisition of information found over the Internet, by taking keywords from the user and searching the web for relevant documents.

Web search engines among others comprise of a local or distributed index where documents fetched by spiders are stored. Search engines' indexing mechanisms are more sophisticated than the ones provided by corpus management systems, since by

default the former need to handle larger volumes of data and serve more concurrent users.

Moreover, search engines like corpus query systems support a query interface, where users can submit their search request(s) and a ranked list of relevant documents is retrieved as a response to a user's information need(s) instead of concordance or collocation lists returned by corpus management systems. Concerning the query language, a typical corpus query system, e.g. QWICK (http://www-clg.bham.ac.uk/QWICK/doc/query.html) offers facilities such as wildcards, compound queries (e.g. word sequences, alternatives) etc. On the other hand all these operations are supported by a search engine as well through the use of boolean logic operators (AND, OR, NOT), other set or exact match operators (+, -, " ") and wildcards as part of their query.

With respect to the queries issued, search engines also support multilingual and sub-domain searching. Furthermore, search engines can be easily exported to the web and it is possible to incorporate linguistic modules (e.g. stemmers, thesauri) in order to improve retrieval performance. Even though search engines do not provide concordance lists of the user's search request(s), they have the potential of offering such a service while at the same time they can also provide relevance or popularity scores, e.g. Google (http://www.google.com/technology/index.html), of the retrieved documents.

Thus, despite the similar functionality and structure among search engines and corpus query systems, the former have the advantage of incorporating more sophisticated services since they are, by default, targeted towards retrieving information from heterogeneous and dynamic collections of data. In addition, there is a great difference in the visualization of the retrieved results, since in web Information Retrieval (IR) the entire document is easily accessible, whereas in corpus query systems usually only a part of the document can be viewed. Summarizing, web search engines are at least equivalent to corpus query systems and with some slight modifications they can provide even more searching facilities to end-users.

4 Our Approach

In the proposed approach we examined the possibility that a web search engine can partially function as a corpus query system. We do not claim that a search engine is identical to a corpus management system nor that linguistic information extracted from the study of documents indexed in a search engine is equivalent to the one obtained after studying information stored in a typical corpus. On the contrary, we claim that different kinds of information can be found on online dynamic textual data in comparison to the offline ones. In addition, we are taking into consideration the fact that documents found over the web do not have a particular data structure, whereas documents or phrases comprising corpora are well-structured ones. What we suggest, however, is that both systems, namely search engines and corpus query systems share many things as far as the services they provide and their infrastructure are concerned. We also claim that a search engine can potentially support many components of a typical corpus management system and incorporate even more.

Motivated by the aforementioned observations we incorporated into a Greek web search engine a concordance module in order to test the similarities between search engines and typical corpus management systems. The search engine we used indexes the full

text of 700,000 Web pages with continuous update frequencies. It supports wildcards, boolean searching, term as well as phrase searching, field searching (e.g. title:governmental, url:home.html, keywords), and case insensitive searching. The engine provides two search options: plain and normalized (Ntoulas 2001), thus there are two different indices kept, each corresponding to an option. The plain version of the engine indexes the pages fetched by the spider (including stop words), while the normalized one passes the pages through the normalizer, where stop words are excluded and the remaining tokens are induced to their first inflected form. The display order or ranking of search results is determined by the engine from the location of matching words and occurrence of their frequencies. The concordance module we incorporated in the plain version of the engine is described in detail in the following section.

5 The Concordance Module and its performance

A concordance module, similar to those provided by corpus management systems, was incorporated into the default version of the engine. A query issued to the interface of the engine is analyzed and matched towards its plain index. All documents judged as relevant are retrieved and actually form the data from which a concordance list is extracted for each search request. More specifically, the retrieved documents are analyzed in order for the concordance lines to be extracted and presented to the end user as illustrated in ***Figure 1***.

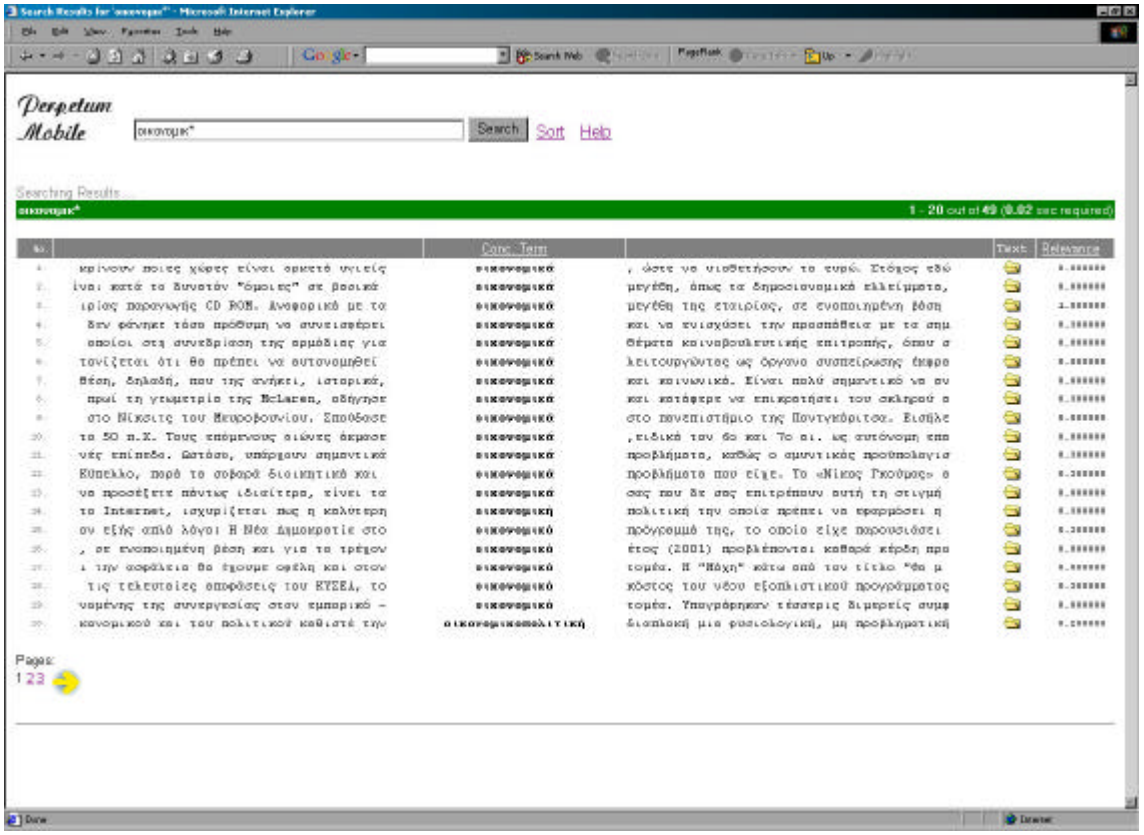

Figure 1 *The concordance module of the search engine*

The user has at his disposal a set of functions similar to the ones offered by a typical corpus query system, ranging from sorting the results in a desirable way, to viewing the context of the whole document containing some specific term(s) by following the links

provided by the engine. On top of that, the user apart from obtaining the concordance lines of a term found in a particular document he has also a more global view of the representativeness of the specific term in the document, since the engine displays also the relevance scores of each retrieved result.

The concordance module is not yet publicly available and thus we cannot report on users' interaction with it, since there is no feedback collected so far from end users concerning its impact in retrieval performance. We can however report on the system's response time, which ranges from 0.02 to 1 second, thus not affecting the overall system's performance. In addition, we have not incorporated the concordance module into the normalized mode of the engine, since induction of a term's word variants to their respective first inflected form would not give the actual concordance lines of the specific word form but solely of a particular word form.

6 Conclusions

There is a need for sophisticated software in the lexicographers' every day work in order to cope with and organize the rapidly growing bulk of textual data in electronic form and to automate the process of studying linguistic phenomena over the web.

This paper has outlined an approach to treat a web search engine as a corpus query system for aiding the study of linguistic structures that are common to texts, which appear on the Internet. While corpus management systems are quite trustworthy applications for linguistic research, search engines on the other hand provide the advantage that they are dealing with the everyday usage of language, thus enabling the easiest tracing of novel usages of language. Since no study has been reported so far about the structure of the language on the web, such a system could constitute a facility towards this direction. With this work we propose using search engines as a kind of a corpus management system, representing thus a robust and quite flexible infrastructure for linguistic analysis, research and applications. These may vary from detection of uses of terms over the web to the study of word senses that differentiate according to their context.

In the future we plan to extend the concordance module and apply it to the normalized index of the engine by incorporating also a robust parser into the entire system and implement other commonly used linguistic components such as collocations, word and lemma lists etc. Moreover, we plan to conduct a large-scale experiment with end users involved in order to collect feedback on the user's interaction with the module, and thus trace areas that need further enhancement.

References

Christ, O., 1994, "A Modular and Flexible Architecture for an Integrated Corpus Query System" In Proceedings of COMPLEX'94

Ntoulas A., Stamou S., Tzagarakis M. 2001 "Using a WWW Search Engine to Evaluate Normalization Performance for a Highly Inflectional Language". To appear in Proceedings of the ACL 2001 Student Research Workshop, Toulouse, France

Kokkinakis D. 2000 "Concordancing Revised or How to Aid the Recognition of New Senses in Very Large Corpora". In Proceedings of the 2nd International Natural Language Processing Conference NLP 2000, Patras, Greece

Miller G., Beckwith R., Fellbaum C., Gross D., Miller K 1993 “Introduction to WordNet: An on-line lexical database”. Technical report, Cognitive Science Laboratory, Princeton University

Mladenic D. 1999 “Text-Learning and Related Intelligent Agents” Revised Version. In IEEE Expert Special Issue on Applications of Intelligent Information Retrieval, July – August 1999

Sinclair, J.M., 1991 “Corpus Concordance Collocation”. Oxford: Oxford University Press

http://www.google.com/technology/index.html

http://www-clg.bham.ac.uk/QWICK/doc/query.html