Towards Better Integration of Dynamic Search Technology and the World-Wide Web

Douglas McKee (doug@navisoft.com)

Abstract

Most World-Wide Web (WWW) sites make minimal use of information retrieval (IR) technology. At best they start with a set of HTML documents and index them with WAIS, a fast but simple information retrieval engine. Users browsing these sites have the option of doing a keyword search of the database.

We are building new WWW server software that:

- Uses natural language processing (NLP) based retrieval software that has better precision and recall than WAIS;
- Incorporates WAIS's ability to search several databases at once and lets the users select those databases;
- Allows the user to pose mixed relational and natural language queries;
- Lets the user customize several retrieval features including number of documents to return, format of the list of returned documents, and any knowledge used by the system;
- Generates improved queries based on user feedback;
- Lets users see natural clusters in a set of documents; This is especially valuable for databases that change over time;
- Dynamically creates hyper-links between related documents and parts of documents;

1. Current State of the Web

Most users browse the World-Wide Web (WWW) using embedded hypertext links to move from document to document or within documents. As the success of the WWW and other hypertext systems like Apple's HyperCard and Microsoft Help have proven, this paradigm works well for many applications. For example, users looking around to see what the WWW offers for the first time can access a wide range of information in a short amount of time. When a person needs to do focused research, however, it can be frustrating to depend on someone else's organization of information. In these cases, a content-based search is more appropriate.

Global indices of information on the Internet include Archie, Veronica, and ALIweb. These software programs index information based on filenames and user authored descriptions, and let users issue searches. While Archie and Veronica build their indices automatically by roaming the net, ALIweb depends on authors to submit descriptions of their "chunk" of the WWW. These descriptions must be kept up to date. As sites put larger information sources on the Internet, it becomes very expensive, in terms of both time and storage, to build these global indices.

A new approach is that of client-based retrieval implemented at Eindhoven University of Technology [DeBra]. These researchers have extended NCSA's Mosaic to accept keyword queries and search for documents containing those words by automatically following links from the current "page." This approach is flexible and does not require any indexing on the server side, but requires bandwidth proportional to the size of the collection being searched, since it must get each document and search it on the client machine. It does not take advantage of any precomputed index.

Many sites are using WAIS to tackle the problem of indexing their own large collections. From certain pages at a site, a user will have the option of posing a keyword query. A list of documents will come back, each linked to the full text or multimedia version of the document. WAIS's search is more accurate than that of global indices such as ALIweb, Archie and Veronica, since it indexes the full text of a document, but it is still a simple keyword-based method. Although the standard WAIS client allows users to transparently search multiple distributed databases, most WAIS-indexed WWW sites do not.

To improve search accuracy, other sites have integrated retrieval systems with sophisticated query languages that add Boolean operators (AND, OR, NOT) and proximity operators (within-X-words, same sentence, same paragraph). These systems sometimes let the user associate weights with words in the query. While these features give the user more control over the retrieval, the languages are hard to learn and non-standard.

This paper describes new WWW server software that solves some of the problems enumerated above by integrating technology from information retrieval (IR) and natural language processing (NLP). Our server also lets users customize search parameters, mix relational and natural language queries, and browse large collections of documents that do not contain embedded hyperlinks.

2. Natural Language Technology

Instead of burdening the user with learning a sophisticated query language, our software lets the user pose his/her query in natural language and applies NLP technology to that query as well as the textual contents of each indexed document. Most IR systems, including WAIS, treat documents as linear lists of words. Function words like "the," "of," and "and" are ignored and often words are reduced to a stem (e.g., "computers" ⇒

"compute") . To improve accuracy, IR systems need richer representations of document content [Salton83].

Linguists have been studying natural language for years and recognize several distinct levels of representation of natural language. These include the morphological structure of words, syntactic structures, semantic predicate argument structures, and the discourse relations between pronouns, definite noun phrases, and their antecedents [Fromkin].

Our system does NOT attempt to do complete natural language understanding of documents and queries. Only recently have many linguists begun to scale their theories up to large quantities of real world text and automatic methods are even farther behind. Projects that are underway to tag large corpora with linguistic structure will help test these theories and provide test data for automatic methods. One of these, the Penn Treebank [Marcus] has been able to consistently tag millions of words of text with syntactic structure. While no existing parser can accurately identify complete syntactic structure, there are several syntactic parsers [Hindle, DeMarcken] that can accurately identify low-level constituents (e.g., simple noun phrases, prepositional phrases) in naturally occurring text.

NLP technology has been successfully applied in many areas to improve the accuracy of IR systems. Syntactic parsers can identify multiword phrases that can serve as indexable terms [Fagan, Stralkowski]. Other researchers have generated thesauri automatically [Stralkowski], and by automatically disambiguating words with multiple meanings (e.g., crane (*construction equipment*) vs. crane (*bird*)) [Yarowsky, Krovetz] systems can become much more accurate. Our software incorporates these methods.

NLP techniques are especially effective when applied to collections of short documents. Picture Network International (PNI) indexes a collection of hundreds of thousands of images with natural language captions. Because of the size of the images that are returned (usually across phone lines), it is important that any retrieval be as accurate as possible. Since captions are short, content words usually only appear once, reducing the likelihood of an exact keyword match. The information in so-called stop words is critical to distinguish cases such as "people inside a house" from "people outside a house."

3. Distributed Queries

WAIS servers index a set of documents and answers Z39.50 information requests. Many WWW sites that use WAIS will index their HTML documents and pass along users' keyword queries to this server. They show the user a list of the returned documents. The standard WAIS client, however has many features that are not usually integrated into the WWW. One of the most powerful is its ability to let the user search several distributed databases at once.

Users do not like to repeat searches or use different query languages for each source they want to search. Our software lets a user search directories of servers or add servers

manually to a list of servers to search. Queries are translated into a form suitable for each server, sent to each server, and the results assimilated (See Fig 1).

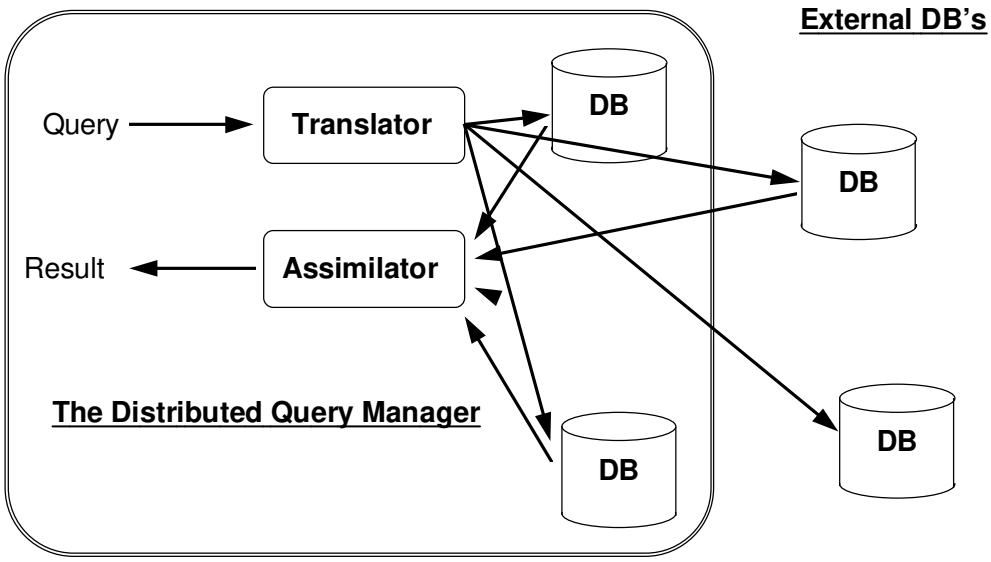

Figure 1: Invisibly to the user, queries can be distributed to internal or external databases.

4. Relational Attributes

Many sites may want to associate relational attributes with documents. For example, a virtual art museum may want to index their pieces by natural language description and the following:

- Artist,
- Date of creation,
- Media,
- Country of origin,
- Style

Our server integrates a standard DBMS and lets the museum associate any number of attributes with each piece of art. The site may specify each attribute's allowable type or attach stored procedures to attributes.

When querying the document base, users may specify required values for attributes that act as a filter before the natural language search is initiated. Sometimes, our software will translate part of a natural language query into relational form. For example, in the query

"Give me sculptures of people by Rodin," the prepositional phrase "by Rodin" is translated into ARTIST = "Rodin," and the phrase "sculptures" is translated into MEDIA = "sculpture." In more complex cases, queries may be posed using SQL.

5. User Preferences

Our server keeps track a user's identity throughout each session and maintains a set of personalized preferences for each user. Most of these parameters control the behavior of the search and retrieval.

To control the amount of information returned in response to a query, users can specify a maximum number of documents, or a minimal probability of relevance. The probability of relevance is used to order documents when they are displayed to the user.

Users may also specify the features to display in a summary of a document set. In some cases, the user may want to see the title of the document and the date it was written, but in others, the title and author might be more useful information. In still other cases, the user may want to see as many documents as possible on the screen and opt for just the title.

Perhaps the most important user-settable retrieval parameters are the configuration of knowledge sources used by the search algorithms. Users may tell the system to use specialized domain (e.g., medical or computer) dictionaries and knowledge bases.

6. User Feedback

Many systems allow users to issue a query for documents that are like a particular document that the system returned earlier. They use a general similarity function and compare each document in the collection to the query document, returning the maximally similar documents.

Our system lets users mark what documents are relevant or irrelevant to a given query and builds a new query that takes advantage of this information. By iterating, users can make successively precise queries that find exactly the documents they are looking for.

Users may treat the results of a search as a logical document set and apply new queries to this set. This lets a user start with a general query and iteratively pare down the number of documents in that set.

7. Document Clustering

Oftentimes, users have access to a very large document database and want to get a general idea of its contents. This is easy when the database administrator has set up a high-level organization of the contents, but when the data has no introduction page or pages, this can be very difficult. Real world examples include a days worth of newspaper stories or a

news group archive. By using a general similarity metric, like document vector distance and a clustering algorithm, our server will show a user the major "chunks" of a database.

We are building two distinct display mechanisms. The first is textual and displays the titles of the prototypical documents in each cluster along with the titles of prototypes of any subclusters. The second shows the clustering as a graph in two-space, where each point represents a document. Users can point and click on documents to display them.

Users may also use clusters as logical databases, and initiate searches over them, just as they would over a complete database.

8. Dynamic Link Creation

Another way to help users browse a large collection of linear documents is to automatically insert hyperlinks [Salton94]. We create three distinct kinds of links. Applying the same similarity metric that we use for document clustering, our software will insert links at the bottom of each document that point to related documents. In other words, we are precomputing the result of a "more like this" query.

The second type of automatically generated hyperlink connects sections of documents that vary in size from chapter down to paragraph. If two sections are sufficiently alike, based on the similarity metric, a link is created from the smaller section to the probably more detailed larger section.

The third kind of link that we generate connects small phrases to documents that discuss those topics in detail. Since detecting phrases that correspond to meaningful topics is a difficult problem, we currently restrict ourselves to proper names. To determine if a document is sufficiently "about" a proper name, we look for several occurrences of the proper name in the document and the title of the document.

9. Conclusion

We are developing WWW server software that supports dynamic content-based search, an important and useful paradigm that supplements WWW hypertext navigation. Our retrieval is more accurate than WAIS or any of the global indexing software such as Archie, Veronica or ALIweb. Since it is based on NLP technology, users do not need to learn a complex query language. We also integrate tools that let users customize search parameters, mix relational and natural language queries, and browse large collections of documents that do not contain embedded hyperlinks.

References

DeMarcken, C., "Parsing the LOB Corpus", *Proceedings of the 28th Annual Meeting of the ACL*, 1990.

DeBra, P., G. J. Houben, and Y. Kornatzky, "Navigational Search in the World-Wide Web," ftp://ftp.win.tue.nl/pub/infosystems/www/Mosaic-2.1-fish-short-paper.tar.gz.

Fagan, J., "Automatic Phrase Indexing for Document Retrieval: An Examination of Syntactic and Non-Syntactic Methods," *Proceedings of the 10th Annual Meeting of ACM SIGIR*, 1991.

Fromkin, V. and R. Rodman, *An Introduction to Language*, New York, Holt, Rinehart, and Winston, 1974.

Hindle, D., "User Manual for Fiddich, a Deterministic Parser," Naval Research Laboratory Technical Memorandum 7590-142.

Kale, B. and A. Medlar, "An Information System for Corporate Users: Wide Area Information Servers," ftp://think.com/wais-corporate-paper.text, April 1991.

Krovetz, R., "Lexical Acquisition and Information Retrieval," in *Lexical Acquisition: Building the Lexicon Using On-Line Resources*, U. Zernik (ed.), pp. 45-64, 1991.

Marcus, M., B. Santorini, and M. A. Marcinkiewicz, "Building a Large Annotated Corpus of English: the Penn Treebank," *Computational Linguistics*, Vol 19 No 2, June 1993.

Salton, G. and M.J. McGill, *Introduction to Modern Information Retrieval*, McGraw-Hill Book Co., New York, 1983.

Salton, G. J. Allan, and C. Buckley, "Automatic Structuring and Retrieval of Large Text Files" *CACM*, Vol 37 No 2, February, 1994.

Stralkowski, T. and B. Vauthey, "Information Retrieval Using Robust Natural Language Processing," *Proceedings of the 30th Annual Meeting of the ACL*, 1992.

Yarowsky, D., "Word-Sense Disambiguation Using Statistical Models of Roget's Categories Trained on Large Corpora," *Proceedings COLING-92*.