Improving Web searching with user preferences.

Web Search—Your Way

The World Wide Web was estimated at over 800 million indexable pages circa 1998 [9]. When searching the Web, a user can be overwhelmed by thousands of results retrieved by a search engine, few of which are valuable. The problem for search engines is not only to find relevant results, but results consistent with the user's information need. It is a user's information need that determines which documents are valuable. Typical search engines limit the user to entering only a keyword query, even though users consider more than the topic when making relevance judgments [1, 10].

We describe a metasearch engine architecture, in use at NEC Research Institute, that allows users to provide preferences in the form of an information need category. This extra information is used to direct the search process, providing more valuable results than by considering only the query. Using our architecture, identical keyword queries may be sent to different search engines, and results may be scored differently for different users.

Unlike typical search (or metasearch) engines, our architecture considers the user's information need when determining which sources are queried, how queries are modified for those sources, and how to score the retrieved results. Each of these can vary independently from the keyword query.

The Web is a very large collection of heterogenous documents, however, Web pages are unlike typical documents in traditional databases. Pages can be active (animations, Java), can be automatically generated in real time (current stock prices or weather information), and may contain multimedia (sound or video). The authors of Web pages have very diverse backgrounds, knowledge, cultures, and aims. Furthermore, the availability of metadata is inconsistent (for example, some authors use the HTML heading tags to denote headings and subheadings in their text, while others use different methods, such as the HTML font tags or images). Efforts such as XML and Dublin Core aim to improve metadata, however, it seems unlikely that all Web page authors will adhere to complex standards. Only about one-

Eric J. Glover, Steve Lawrence, Michael D. Gordon, William P. Birmingham, and C. Lee Giles

third of Web server home pages use the simple HTML META tag standard available today [9].

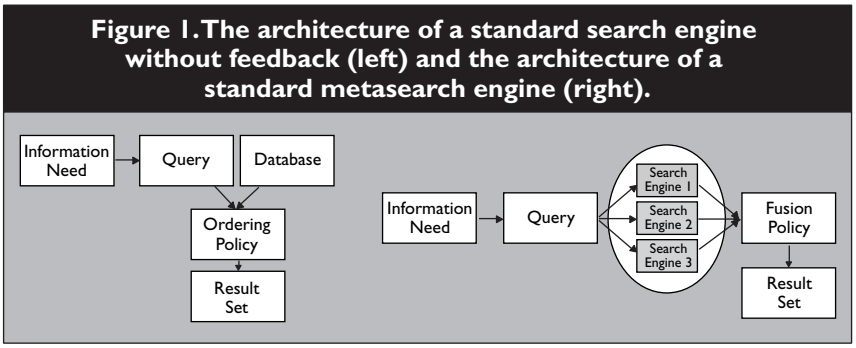


Figure 1. The architecture of a standard search engine without feedback (left) and the architecture of a standard metasearch engine (right).

Web Search Engines

Web search engines crawl the Web, downloading and indexing pages in order to allow full-text searching. There are many general-purpose search engines; unfortunately none of them comes close to indexing all of the Web [9]. There are also thousands of specialized search services that index specific content or specific sites. The great variability of search services available, and the lack of comprehensiveness of any of them, has led in part to the introduction of metasearch engines.

A metasearch engine typically provides a single interface to multiple search engines and combines the results into a single unified list [12]. The ordering of the list is usually determined by the short summaries or scores returned by the search engines, or both. Metasearch engines can have great difficulty determining valuable documents, due to the variability and lack of information known about the individual search engines. For example, if just one engine returns many irrelevant results, a typical metasearch engine may have no way of preventing these results from ranking highly. Some popular metasearch engines include MetaCrawler (www.metacrawler.com) and SavvySearch (www.savvysearch.com).

One of the goals in creating the metasearch engine Inquirus was to avoid difficulty in merging the results from multiple search engines [8]. Inquirus downloads and analyzes all pages listed by the search engines. With the full text of all pages, the document ordering problem returns to the easier, but still very difficult, problem encountered by standard search engines. The architecture of Inquirus also provides many other benefits, such as the ability to display query-sensitive summaries, results that are always up to date with the current contents of the Web (improving relevance), and improved duplicate detection.

Relevance and Value

Information-retrieval systems are concerned with both relevance and constraints. Relevance refers to the binary state of whether a document is on the same topic as the query or not. A constraint refers to an additional condition that must be met. One of the limitations of retrieval using solely relevance and constraints is that users may have preferences over relevant documents that cannot be expressed. For example, a user searching for current events about a recent earthquake might find two similar documents. One is yesterday's news; the other is today's. Although both may be "relevant," the user may prefer today's news. If the user were to apply constraints, he or she could only say things such as "in the last week." This may rule out old documents; however, it does not provide any information about how the user differently values documents that meet the constraints. Additionally, if the "best document" was one week and one second old, it would be excluded.

Rather than relying on relevance, we use the concept of value. The value of a document is subjective. Users with identical queries may place different value judgments on the same document. Even the same user's value judgments can change over time. When there are a small number of relevant results, it may be acceptable to present them all to the user. However, when a search finds hundreds or thousands of possibly relevant results, ordering decisions that incorporate the concept of value and are based on more than just keywords become far more desirable.

Architecture

Figure 1 shows the architecture of typical search and metasearch engines. The user's information need (IN) is approximated, often poorly, by a query. The query is applied to a local database of Web pages and the results are ordered and shown to the user. Most search engines have a single ordering policy: all users with the same query get the same results presented in the same order. Figure 1 also shows the architecture of a typical metasearch engine. A metasearch engine does not have a local database and relies on other sources (other search engines), as shown in the figure. The results returned from the other sources are combined through some combination policy, also called a fusion policy. When ordering results, metasearch engines typically consider only the titles, summaries, and URLs provided by the sources. Inquirus changes the process by fetching and analyzing individual pages. This allows the use of a consistent scoring function, making the ordering problem more like that of a standard search engine.

Figure 2. The architecture of the Inquirus 2 search engine.

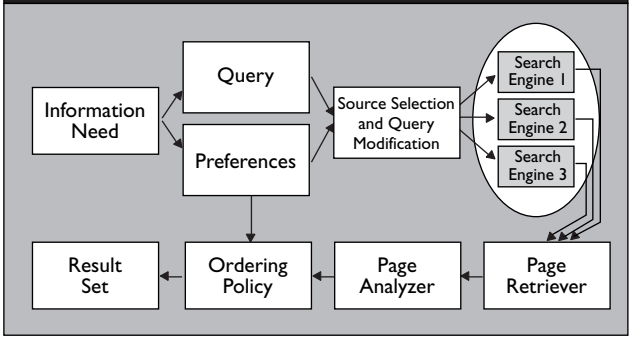


Table 1. Sample information need categories.

Name	Description	Sample attributes used
Current Events	Current events or recent news	TopicalRelevance, DaysOld
Detailed	Detailed pages about the given query terms	TopicalRelevance, AverageGrade, WordCount, WordsPerSection
Research Papers About	A document in the style of a research paper, about the given query	TopicalRelevance, AverageGrade, ResearchPaper, WordCount
General Introductory	General or introductory pages about the given keywords	TopicalRelevance, GFOG, GenScore, WordCount
Individual Home pages	The home page of the specified query	TopicalRelevance, Homepage
Organizational Home page of	The home page of the organization specified in the query	Keywords in title or domain or summary, Homepage, Pathlength
Organizational Home page About	The home page of an organization related to the given query	TopicalRelevance, Homepage, Pathlength

To allow search decisions to be consistent with user information needs, we have created a new architecture that extends typical metasearch engines, as shown in Figure 2. The new architecture adds user preferences to the query. Rather than being limited solely to the use of keywords for expressing an information need, the user can provide an information need category that controls the search strategy used by the metasearch engine. We add explicit user preferences that directly influence source selection, query modification and the ordering policy. The architecture does not specify the explicit form of the preferences, or exactly how to use them.

Each information need category has an associated list of sources, modification rules, and a scoring function. For example, choosing "current events" instructs the system to search ABCNews, News.com, Snap.com, AltaVista, Yahoo, and HotBot. The query to HotBot is modified to constrain the search to only pages updated within the last two weeks. The queries to ABCNews and News.com specify the results should be sorted by date with the most recent results appearing first. When a search engine responds, the pages listed are downloaded and analyzed, then scored using the associated utility function. Every category has its own list of sources, query modifications, and utility function (ordering policy).

Source selection. In theory, a metasearch engine could search all possible sources. In reality, limitations on network bandwidth and the accuracy of document ranking make it preferable to only search sources that are likely to yield valuable results.

A standard metasearch engine always uses the same source search engines: the source-selection process does not change. Metasearch engines such as SavvySearch, ProFusion, Inquirus, and MetaSEEK might not send all queries to the same search engines. Some engines allow the user to select groups of search engines (such as "News" or "Sports"), or to select individual engines. Others attempt to map the keywords in the query to the best search engines.

Inquirus 2 does source selection based on user preferences. Preferences could be a set of sources, similar to other metasearch engines. The preferences currently used by Inquirus 2, however, provide a high-level description of the user's information need. A sample of the currently supported information need categories is shown in Table 1. Currently, sources are hand-coded for each information need category.

Query Modification

To enhance the number of results relevant to a specific need, Inquirus 2 performs query modification. There are three types of query modification used: utilization of search engine-specific options, prepending terms to the query, or appending terms to the query. In addition, more than one modified query can be submitted for a given search engine. For example, when searching with the "General Resources" information need category, the user's query to AltaVista is currently modified as follows: three queries are submitted; the first query prepends `what is` and the second appends `links resources`, while the third is unmodified. As a result, a user searching for general pages about "Linux" will retrieve pages such as "What is Linux," or "Linux Links page," both of which might not normally score highly with AltaVista's ordering policy. The unmodified query is still submitted to ensure the query modifications do not cause very valuable results, normally ranked highly by

AltaVista, to be missed.

The ability to modify the query for a given need and search engine allows Inquirus 2 to include general-purpose search engines for a specific need. For example, a user searching for news might not normally use Northern Light or HotBot, but query modification can be used to cause their results to be ordered by date, or to add date constraints, thus returning many recent, potentially valuable documents. The Inquirus 2 source selection process allows many special-purpose search engines to be used, with individual engines only being queried when appropriate for the given information need.

Table 2. Some of the page-specific attributes currently available for use in Inquirus 2.

Name	Description
AverageGrade	Average of three grade-level algorithms, FOG, SMOG, and FK
GFOG	A reading-level algorithm optimized for less-advanced documents
WordCount	The number of words per page
WordsPerSection	The number of words divided by the number of "sections"
Homepage	A measure of the number of home page-like features present
GenScore	A measure of features indicative of a "general" page, such as the keywords "links" or "resources"
ResearchPaper	A measure of features indicative of a "research paper" page, such as having an abstract or references
SectionCount	The number of sections on a page
Pathlength	The depth of page from the top of a domain in levels
TopicalRelevance	A query-dependent attribute predicting how much a particular page is "about" the given query. The attribute is based on word distances, from each other and the top of the document, as well as the number of occurrences of each term
Summary	An automatically generated summary of the document
DaysOld	The predicted age (in days) of the content of the page

Ordering Results

To incorporate multiple factors into the ordering policy for Inquirus 2, we use Multi-Attribute Utility Theory [6] to represent user preferences. Inquirus 2 represents user preferences as an additive value function [6] over any of the available metadata. There are two factors for each attribute: the relative weight and the attribute-value function (the mapping from the attribute's assignment to its value). As a simple example, a user's preference for "current events" might be represented as a function of date, and an attribute we call TopicalRelevance, which is a measure of how much a document is about a given query. The preference for current events might be reflected by a 60% weight on TopicalRelevance and 40% weight on DaysOld. Our current system has one manually entered function for each information need category. We intend to use learning to "discover" the best function for each category, as well as to specialize the categories for different users.

Table 1 lists some of the attributes used for the various information need categories, and Table 2 describes some of the document-specific attributes available. In addition to document-specific attributes, such as WordCount, there are keyword-specific attributes that indicate if a particular keyword is in the title or URL, or how far the keyword is from the top of a document. The utility functions can be any linear combination of the attributes.

Sample Search

To use Inquirus 2, the user enters a query and chooses an information need category. The user can also select the maximum number of hits, results display format, and whether or not to use the dynamic display applet. The Java-based dynamic display applet dynamically reorders results as they are retrieved and analyzed, always displaying the highest ranked documents among those retrieved so far. The dynamic display applet allows users to examine the results that have been processed at any point during a search, while Inquirus 2 continues to download and process additional search engine responses and documents.

Figures 3 and 4 show the user interface for Inquirus 2, and two different result sets for the query `agent based information retrieval`. Figure 3 shows results for the information need category of "Research papers about," while Figure 4 shows results for the information need category of "General introductory about". When searching for research papers, Inquirus 2 searches AltaVista, Google, HotBot, Northern Light, Snap, and Yahoo. Google and Yahoo have modified queries submitted to increase the chance of finding pages that are research papers (or otherwise valuable pages, such as a reference list). Both modifications consist of appending "abstract keywords references" to the end of the user query, since a typical research paper will contain sections named "abstract," "keywords," and "references."

The results are scored based on several attributes, including TopicalRelevance. A very good page for a preference of research papers would be strongly about the topic, and have many characteristics of a research

Figure 3. The dynamic display of Inquirus 2 for the query `agent based information retrieval` and the information need category of "Research Papers About."

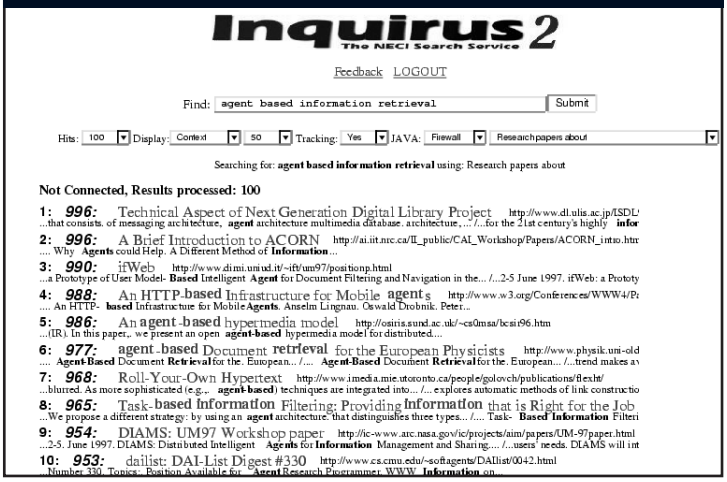
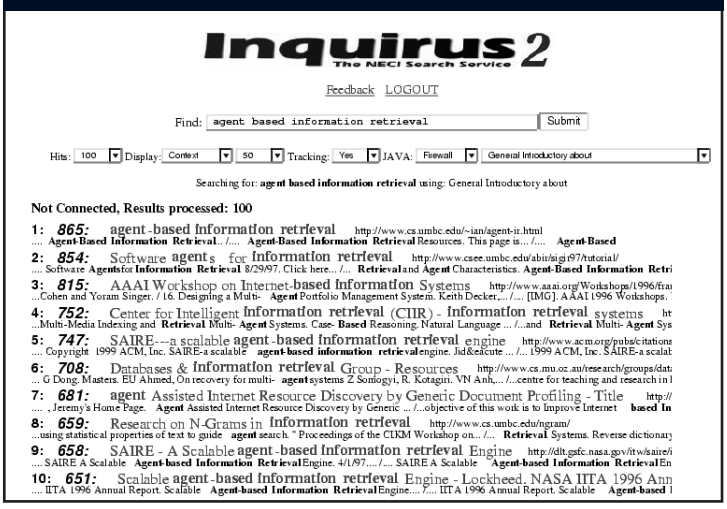


Figure 4. The dynamic display of Inquirus 2 for the query `agent based information retrieval` and the information need category of "General Introductory About."



paper. Some of the more important attributes in determining the latter include WordCount (longer pages are better), AverageGrade (higher grade level is better), and the attribute ResearchPaper, which is a measure of features characteristic of a research paper, such as having an abstract, an introduction, keywords, and references. To further improve the topical relevance, pages where the query terms occur closer to the top of the document (in the abstract, title, or keywords) score higher than pages where the terms occur further down.

Of the top 10 results shown in Figure 3, all but one was a research paper and of those, all but one was highly related to agent-based information retrieval. The same query (unmodified) submitted to Northern Light, resulted in only one research paper and one reference list out of the top 10 ranked results. Similarly for Yahoo, with an unmodified query, only two pages of the top 10 were research papers, with one reference list. The modified query to Yahoo returned six research papers out of the top 10 results (most, but not all of them about the correct topic).

Figure 4 shows the results for the same query, but a different information need category: "General introductory about." Unlike research papers, where the user wants detailed pages, here a user prefers more general pages, with a looser format requirement. The query is submitted to the same search engines (except Northern Light), but with different query modifications. For this category, three queries were submitted to AltaVista; one prepended "What is" and one appended "links resources," the third was unmodified. The query to Google was modified by appending "links resources." Unlike the results shown in Figure 3, very few results from the top 10 were found exclusively from modified queries. In fact, most of the top 10 results were found by more than one search engine; where for research papers, there was almost no overlap. The results shown in Figure 4 demonstrate a wider variety of pages, including home pages of organizations, a general pointers (resource) page, and two presentations relevant to the query. A general Web page is not necessarily of a single "category," but rather has certain properties, such as a lower grade level.

The architecture of Inquirus 2 opens up the search process, allowing experts, or individuals, to define how each attribute should be considered. In addition, the architecture allows easy addition of new search engines and query modifications. The specific search decisions that Inquirus 2 currently makes could be significantly improved, either manually or automatically; however, sample queries using the current decisions show the architecture of Inquirus 2 has the potential to provide substantial improvements over regular search engines in locating results of value to the user.

Related and Future Work

There has been previous work related to using utility theory to score documents, and previous work in intelligent source selection. Grossman and Frieder [4] describe various algorithms and heuristics for information retrieval, both centralized and distributed. They also describe how some Web search engines work, and some of the problems they face. Mizzaro [10] provides an excellent summary of the concept of relevance, including a brief discussion of the relation between utility theory and information

retrieval. Kochen [7] suggested applying utility theory specifically to documents, and described four axioms, which if met imply the existence of a utility function that can be used to order documents. Previous implementations using utility theory for scoring documents include the DIVA system [11] for video recommendations, and the preference agent from the University of Michigan Digital Library project [3]. Several researchers have considered intelligent source selection. For example, Howe and Dreilinger [5] proposed a method of selecting search engines based on the query keywords, and Gauch et al. [2] describe how ProFusion chooses the best sources based on the predicted subject of the query. Northern Light provides a "custom folder" approach that clusters documents by type. Results are grouped into "folders" (possibly overlapping), with which a user can constrain the search. For example, a user could constrain a search to conferences, or the subject area of "information retrieval." The folders are determined at runtime based on the results returned from the query, and can include folders by subject, source, type, or language. In contrast, Inquirus 2 uses value-based ordering. The actual value of a particular result depends on more than its "type," and a valuable result may fall outside a given cluster boundary. For example, when searching for research papers, a reference list very strongly about a desired topic might be more valuable than a research paper that mentions the topic once in a footnote. Likewise, when searching for someone's home page, a CV or resume might be the second-best choice, even though neither fall into the home page cluster.

Inquirus 2 is currently in use at NEC Research Institute. In more recent work, we have implemented and tested the use of machine learning for query modifications and scoring functions, and performed a user study that has confirmed the effectiveness of the Inquirus 2 architecture.[1] In future work, we plan to allow users to easily generate their own categories, and we plan to extend our work on learning document scoring functions. C

References

1. Barry, C.L. *The Identification of User Criteria of Relevance and Document Characteristics: Beyond the Topical Approach to Information Retrieval*. Ph.D. dissertation, Syracuse University, NY, 1993.
2. Gauch, S., Wang, G., and Gomez, M. ProFusion: Intelligent fusion from multiple, distributed search engines. *Journal of Universal Computer Science 2,* 9 (Sept. 1996).
3. Glover, E.J., Birmingham, W.P., and Gordon, M.D. Improving Web search using utility theory. In *Web Information and Data Management (WIDM'98)*, Bethesda, MD, 1998,
4. Grossman, D.A. and Frieder, O. *Information Retrieval: Algorithms and Heuristics.* Kluwer Academic Publishers, 1998.
5. Howe, A.E. and Dreilinger, D. SavvySearch: A metasearch engine that learns which search engines to query. *AI Magazine 18*, 2 (Feb. 1997).
6. Keeney, R.L. and Raiffa, H. *Decisions with Multiple Objectives*. Wiley, NY, 1976.
7. Kochen, M. *Principles of Information Retrieval*. Melville Publishing Company, Los Angeles, CA, 1974.
8. Lawrence, S. and Giles, C.L. Context and page analysis for improved Web search. *IEEE Internet Computing*, (July–Aug. 1998), 38–46.
9. Lawrence, S. and Giles, C.L. Accessibility of information on the Web. *Nature 400* (July 8, 1999), 107–109.
10. Mizzaro, S. Relevance: The whole history. *Journal of the American Society for Information Science 48*, 9 (Sept. 1997), 810–832.
11. Nguyen, H. and Haddawy, P. The decision-theoretic video advisor. In *AAAI Workshop on Recommender Systems*, 1998.
12. Selberg, E. and Etzioni, O. The MetaCrawler architecture for resource aggregation on the Web. *IEEE Expert*, (Jan.–Feb. 1997), 11–14.

Eric Glover (compuman@research.nj.nec.com) is a scientist at the NEC Research Institute in Princeton, NJ.

Steve Lawrence (lawrence@research.nj.nec.com) is a research scientist in Computer Science at NEC Research Institute in Princeton, NJ.

Michael Gordon (mdgordon@umich.edu) is a professor and Chair of the Computer and Information Systems department at the University of Michigan BusinessSchool.

William Birmingham (wpb@eecs.umich.edu) is an associate professor in both the Electrical Engineering and Computer Science Department and in the School of Information at the University of Michigan.

C. Lee Giles (giles@research.nj.nec.com) is the David Reese Professor of Information Sciences and Technology, Professor of Computer Science and Engineering, and Associate Director of Research at the eBusiness Research Center at Penn State University.

[1]The recent work was part of the Ph.D. dissertation: *Using Extra-Topical User Preferences to Improve Web-based Metasearch*, E. Glover, University of Michigan, 2001.