Experiences with Selecting Search Engines Using Metasearch

DANIEL DREILINGER
MIT Media Laboratory
and
ADELE E. HOWE
Colorado State University

Search engines are among the most useful and high-profile resources on the Internet. The problem of finding information on the Internet has been replaced with the problem of knowing where search engines are, what they are designed to retrieve, and how to use them. This article describes and evaluates SavvySearch, a metasearch engine designed to intelligently select and interface with multiple remote search engines. The primary metasearch issue examined is the importance of carefully selecting and ranking remote search engines for user queries. We studied the efficacy of SavvySearch's incrementally acquired *metaindex* approach to selecting search engines by analyzing the effect of time and experience on performance. We also compared the metaindex approach to the simpler categorical approach and showed how much experience is required to surpass the simple scheme.

Categories and Subject Descriptors: H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval; H.3.4 [**Information Storage and Retrieval**]: Systems and Software

General Terms: Algorithms, Experimentation

Additional Key Words and Phrases: Information retrieval, machine learning, search engine, WWW

1. INTRODUCTION

Search engines are powerful tools for assisting the otherwise unmanageable task of navigating the rapidly expanding World Wide Web. Two types of search engines have been developed: large-scale robot-based and spe-

The experiments in this article were supported in part by ARPA AFOSR contract F30602-93-C-0100, NSF Research Initiation Award RIA IRI-930857, and NSF Career Award IRI-9624058.

Authors' addresses: D. Dreilinger, MIT Media Laboratory, Cambridge, MA 02139; email: daniel@media.mit.edu; A. E. Howe, Computer Science Department, Colorado State University, Fort Collins, CO 80523; email: howe@cs.colostate.edu.

cialty search engines. Large-scale search engines[1] exemplify the trade-off between breadth and quality. They try, fairly successfully, to be comprehensive; as a result, any search may return an abundance of related and unrelated information. The specialty search engines, on the other hand, are inadequate to most topics, but are more likely to quickly focus a search in their area. For example, Yahoo[2] and Point[3] search within smaller, human-reviewed collections of Web site descriptions. DejaNews[4] and the Stanford Information Filtering Tool[5] [Yan and Garcia-Molina 1995] specialize in searching archives of recent Usenet news articles. Tools such as FTPSearch[6] and C|Net's SHAREWARE.COM[7] assist users in finding software and other items available via File Transfer Protocol. Still more search engines target email addresses, newspaper articles, technical reports, books, movie reviews, music recordings—new databases seem to appear daily to satisfy yet more specialized information needs.

The advent of so many search engines leads to another problem: knowing *which* to use *when*. For each search engine, a user needs to know how it works, what it is designed to retrieve, where it is located, when it will provide the best response times, and even the simple fact that it exists; additionally, if one search engine does not locate what was desired, then the user needs to switch to another. Empirical results indicate that no single search engine is likely to return more than 45% of the relevant results [Selberg and Etzioni 1995]. Consequently, working efficiently with the entire collection of search engines can be a challenge and burden for even the most experienced users.

Metasearch engines—tools which access multiple individual search engines—are designed to deal with these problems. By automatically interfacing with multiple conventional search engines, metasearch engines add an additional level of abstraction to Web searching. Several different approaches to metasearching have been deployed. Section 2 introduces a general framework and background information about the various approaches.

In this article, we adopt a pragmatic approach to the problem of information retrieval on the web by focusing on the unusual requirements of one relatively new proposed solution: metasearch. We describe our design for a metasearch system and present results of evaluating its usage by a large number of users on the Web. The purpose is to help motivate the design of

[1] At this time these include the following: Lycos (http://www.lycos.com), WebCrawler (http://webcrawler.com), Infoseek (http://www.infoseek.com), Open Text (http://www.opentext.com), Inktomi (http://inktomi.berkeley.edu), Excite (http://www.excite.com), and Alta Vista (http://www.altavista.digital.com).

[2] http://www.yahoo.com.

[3] http://www.pointcom.com.

[4] http://dejanews3.dejanews.com.

[5] http://sift.stanford.edu.

[6] http://ftpsearch.unit.no/ftpsearch.

[7] http://www.shareware.com.

other metasearch systems and the development of new techniques by relating what we have learned from a deployed system.

For our own metasearch engine, we adopt the principle that Web resources should be used as efficiently as possible. Thus, the success of metasearching depends critically on carefully selecting which resources to use. Our metasearch engine, SavvySearch,[8] carefully selects resources for an individual user's query and balances resource consumption against expected results quality. The primary version of SavvySearch described here employs a unique *metaindex* approach for selecting relevant search engines based on the terms in a user's query; previous experience about query successes and failures is tracked to enhance selection quality.

We evaluated the success of our approach in terms of knowledge requirements. Section 4 describes the experiments, the evaluation measures, and the experimental findings. Section 5 closes with a summary of the results and implications for future research.

2. BACKGROUND

The application, selecting search engines for the Web, is relatively new; thus, we review related ideas from information retrieval and Web search engines. We contrast our view of metasearch with that of some other approaches.

2.1 Information Retrieval

Information Retrieval (IR) systems are software tools that help users find documents contained in a specific corpus or database; these tools are becoming ubiquitous. They are currently used for finding scholarly information (e.g., CARL[9]) as well as for news dissemination, shopping, and many other recreational activities.

The goal of a search engine is to locate relevant information within a corpus. One popular technique involves combining the full text of all documents into an *inverted index*, which maps words to sets of documents that contain them. Each word appearing one or more times in the corpus has a corresponding entry in the inverted index. Along with every transformed word in the inverted index is a list of pointers to each document where that word occurs. Other information can be stored in the index, such as the total number of occurrences of the term in all documents combined, the number of occurrences of the term in each document where it appears, and even the exact location of each occurrence of the word within the page might be included. Including this extra information enables very fast *phrase searching* (for locating groups of contiguous or nearby words) at the cost of a much larger inverted index.

When a user submits a query, the search engine looks up the information for each query term in the inverted index. Search engines using the

[8] http://guaraldi.cs.colostate.edu:2000.

[9] CARL is the Colorado Alliance for Research Libraries' online catalog.

common $tf \cdot idf$ (term frequency times inverse document frequency) ranking algorithm exploit two important qualities of natural-language text to perform accurate retrieval [Witten et al. 1994, pp. 141–148]:

(1) *Term frequency:* If a term occurs frequently in a document, that document is considered more relevant to a query containing that term than other documents with fewer or no occurrences of the same term.

(2) *Inverse document frequency:* In a multiple-word query, the rarer terms (those that occur in very few documents) receive more weight in determining document relevance. For example, if the query is "medieval history," documents containing only the word "medieval" are ranked as more relevant than those containing only the word "history."

2.2 Web Search Engines

For naive users of the Internet, information retrieval means simply following links. As a more informed alternative, some of the larger Web search engines attempt to index the Web in its entirety; many smaller Web search engines search considerably more focused databases—names and email addresses or the full text of Shakespeare's plays, for example. Informed users may go directly to the specialized search engines for appropriate queries and otherwise only try the general search engines should the specialized engines fail on the query. Because we wish to accommodate users who may not even know of the existence of appropriate specialized or general search engines or may wish to simultaneously query several engines, the project described in this article avoids making any distinction between these two types of searching resources and addresses the general problem of deciding which subset of known search engines to query.

2.3 Web Metasearch Engines

Just as robot-based search engines were developed in response to rapid Web growth, metasearch engines are being developed in response to the increasing availability of conventional search engines. Metasearch engines, such as our SavvySearch, are tools that can automatically and simultaneously query several Internet search engines, interpret the results, and display them in a uniform format. Unlike the search engines on which they rely, metasearch engines cannot directly access the corpus of Web documents, but instead use a "corpus" of search engines.

2.3.1 *Architecture of Metasearch Engines.* For comparison purposes, we view metasearch engines in terms of three components:

(1) *Dispatch mechanism:* This is the algorithm, or decision-making approach, for determining to which search engines a specific query is sent. The experimental portion of this article is devoted to analyzing one such mechanism.

(2) *Interface agents:* These self-contained programs manage the interaction with a particular search engine. The interface agents adapt the user's

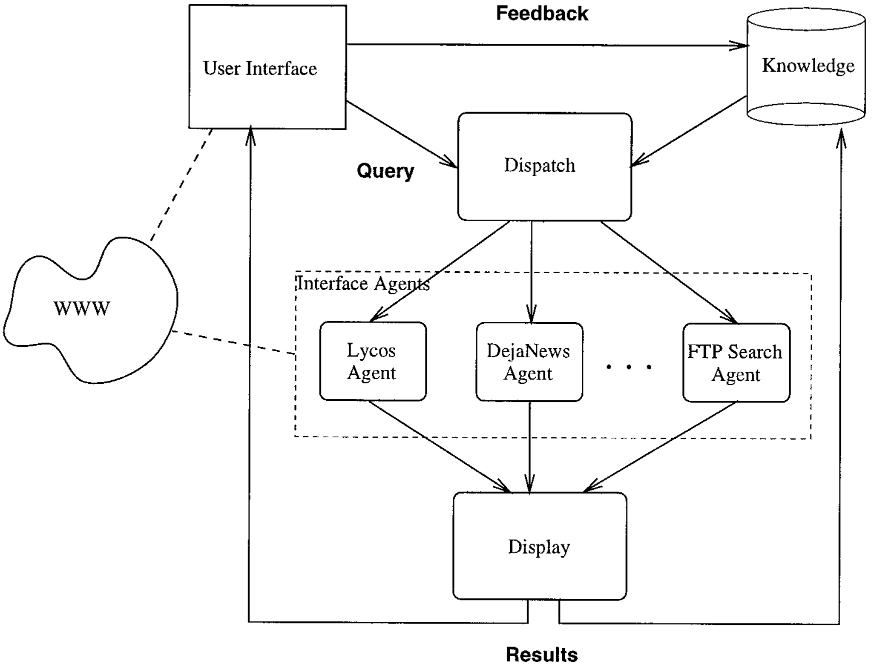


Fig. 1. General framework for characterizing the capabilities of metasearch engines.

query format to match the format of a particular search engine. Interface agents are also responsible for interpreting the diverse native result formats.

(3) *Display mechanism:* Raw results from individual search engines must be integrated for display to the user. Results can be displayed with little additional formatting and can be rank ordered or interleaved. The results can be further "cleaned up" by removing duplicates or verifying the links.

Figure 1 illustrates our idealized architectural view. A user submits a query via the metasearch engine's user interface. The dispatch mechanism determines which remote search engines to send the query. Simultaneously, the interface agents for the selected search engines submit the query to their corresponding search engines. When the results are returned, the respective interface agents convert them into a uniform internal format. The display mechanism integrates the results from the interface agents, removes duplicates, and formats them for display by the user's Web browser.

2.3.2 *Examples of Metasearch Engines.* Over the past few years, many metasearch tools have been developed. We survey a few of the best known of these efforts.

The *GlOSS* (Glossary-of-Servers Server) project [Gravano et al. 1994] uses a metaindex to direct simultaneous search of multiple databases (which they call the *text-database discovery problem*). A metaindex is constructed by integrating the indices of each of the databases. For each database and each word, the number of documents containing that word is included in the metaindex. When a query is submitted to GlOSS, relevant databases are selected by using the metaindex to predict the ones that are likely to produce relevant results. One problem is that each of the search engines must cooperate with the metasearcher by supplying up-to-date index information. As the number of databases increases, the administrative complexity may become prohibitive.

The Harvest system[10] [Bowman et al. 1994; 1995] comprises an integrated set of tools developed by the Research Group on Resource Discovery and Directory Service of the Internet Research Task Force. The Harvest tools provide a means of gathering information from heterogeneous resources, building and searching indexes of this information, and replicating the information throughout the Internet. Although Harvest engines only search a single database, it is possible to create custom gatherers which construct a composite index from multiple information repositories.

Discover provides both query refinement and query routing to over 500 WAIS sites [Sheldon et al. 1995]. The system suggests modifications to the user's query so that they are not inundated with useless results. Then it helps the user identify relevant information providers. The registry of information providers includes a compact description of the contents of each provider that directs query refinement and routing.

The MetaCrawler[11] [Selberg and Etzioni 1995] metasearch project at the University of Washington integrates a set of general Web search engines and dispatches queries to every one of them. MetaCrawler has demonstrated that precise and up-to-date rankings can be constructed by retrieving the HTML source of all referenced documents and applying further textual analysis. Result verification prunes out unavailable resources and irrelevant documents. The precision is obtained at the high cost of network utilization, since all referenced documents must be retrieved.

An alternative to automated metasearch is to allow the user to completely direct query dispatch. Tools such as All-In-One,[12] CUSI,[13] SEARCH.COM,[14] Infi-NET's META Search,[15] and InterNIC[16] are essentially pages full of forms for sending queries to a number of different search engines. The selection process is entirely up to the user—who must type the query into a separate form for each query submission. Only one search engine is activated at a time, and the results appear in the native format of whichever search engine produced them.

[10] http://harvest.transarc.com.

[11] http://metacrawler.cs.washington.edu:8080/home.html.

[12] http://www.albany.net/allinone.

[13] http://pubweb.nexor.co.uk/public/cusi/doc/about.html.

[14] http://www.search.com.

[15] http://members.aol.com/biblprof/meta.htm.

[16] http://ds2.internic.net/tools/meta.html.

The ProFusion system[17] combines many features of the other metasearch engines. It supports both manual and automatic query dispatch and integrates and prunes the results. In automatic query dispatch, the system maps the query to the best three out of six large search engines supported (Alta Vista, Excite, InfoSeek, Lycos, Open Text, and WebCrawler). Query mapping depends on a handbuilt knowledge base that relates query terms to a taxonomy and then associates search engines with the topics in the taxonomy. Recently, in a controlled experiment in which subjects were asked to rate the relevance of results from 10 search engines (the six large search engines, MetaCrawler, SavvySearch, and both manual and automatic versions of ProFusion) on 12 queries, the ProFusion system returned the highest number of links judged relevant by the subjects [Gauch et al. 1996].

3. SAVVYSEARCH

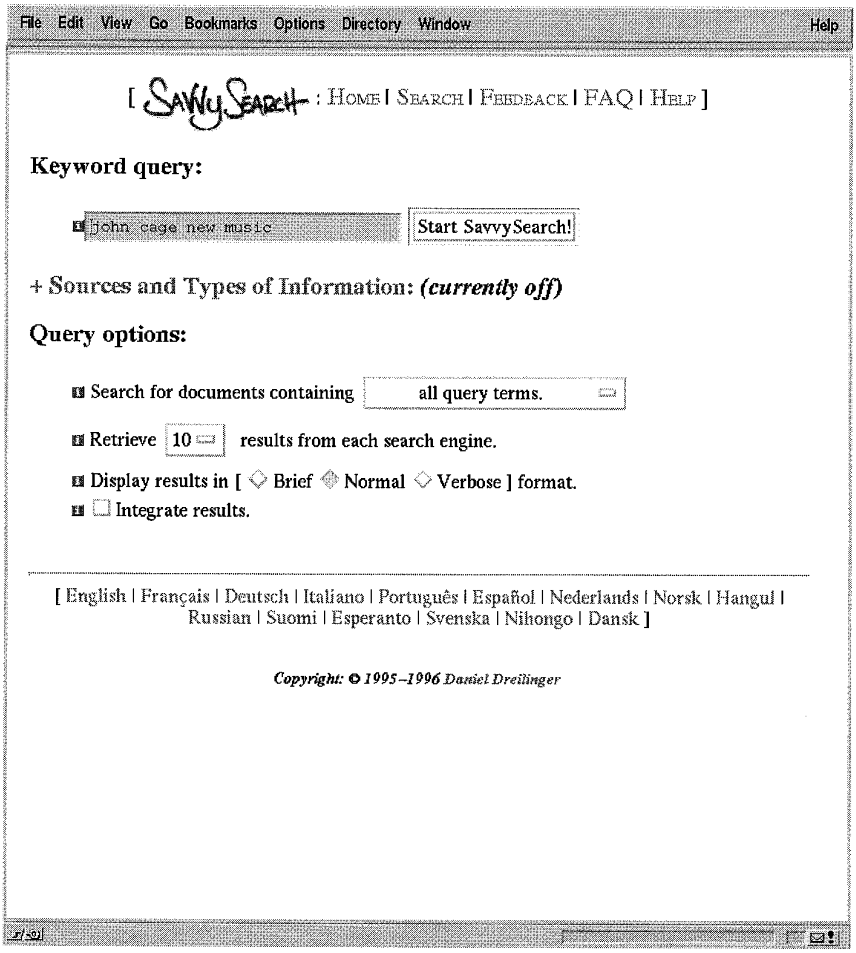
SavvySearch is an experimental metasearching tool which serves as a single interface to many conventional search engines. The original version of SavvySearch was made available on the Web in March, 1995. Since then, one or more of its experimental versions have been publicly available. Users have emailed hundreds of favorable comments and several dozen less-than-favorable ones; the daily usage has increased to over 20,000 queries per day.

SavvySearch's underlying system is implemented primarily in Perl and currently runs on three platforms (two SPARC and an IBM). The user interface has been translated, mostly by volunteers from the user community, into 23 languages to make it more accessible to users from around the world. Queries are sent to remote search engines via HTTP in a similar manner to Web browsers such as Netscape and Mosiac.

The conventional search engines included in SavvySearch are both robot-collected databases of Web documents and specialized search engines. When a user submits a query using the query form (Figure 2), a group of appropriate search engines is selected by SavvySearch and queried simultaneously; the results from the search engines are organized and displayed in a unified format. Users can select to receive results as they arrive from each search engine (the default) or to have the results of different search engines integrated. The latter is implemented by normalizing the scores returned by search engines to between 0 and 1.0 and summing them for each link; links for search engines that did not return scores were arbitrarily assigned a score of 0.5. We did not improve the integration beyond this ad hoc approach because our focus was the selection problem. Users can subsequently request results from additional search engines if they wish to supplement their initial results.

SavvySearch assists Web users in finding relevant information by submitting their queries to multiple search engines. This operation must

[17] http://www.designlab.ukans.edu/profusion.


File Edit View Go Bookmarks Options Directory Window Help
[SavvySearch : Home | Search | Feedback | FAQ | Help]
Start SavvySearch!

Keyword query:

john cage new music

+ Sources and Types of Information: *(currently off)*

Query options:

- Search for documents containing all query terms.
- Retrieve 10 results from each search engine.
- Display results in [◇ Brief ◆ Normal ◇ Verbose] format.
- □ Integrate results.

[English | Français | Deutsch | Italiano | Português | Español | Nederlands | Norsk | Hangul | Russian | Suomi | Esperanto | Svenska | Nihongo | Dansk]

Copyright: © 1995–1996 Daniel Dreilinger

Fig. 2. Query form for user entry of queries for the metaindex version of SavvySearch.

satisfy two conflicting goals: minimizing resource consumption and maximizing search quality [Zilberstein 1995]. Resource limitations make it impractical to send every query to every known search engine; programs that did so would be considered poor citizens of the Web [Eichmann 1994]. It is quite expensive, in terms of Internet resources, to query dozens of search engines for each individual user; moreover, it is unnecessary because some of the search engines are so specialized that they are unlikely to return relevant results for most queries, and some are redundant due to significant overlap in their coverage of sites. Furthermore, the broadcasting approach may inundate the user with quasi-relevant information. Search quality trades-off recall and precision; while broadcasting a query or even submitting it to the large search engines will certainly increase recall, precision will be sacrificed due to the inappropriateness of many returned results.

SavvySearch is designed to query the most relevant search engines first. This section describes two key features of the architecture of SavvySearch: searching advice presented in the user interface as a *search plan* and one of the main dispatch (search engine selection) algorithms.

3.1 Search Plans

From the user's point of view, information gathering on the Web can be viewed as a simple planning problem in which the goal is to find sites satisfying specific criteria and where the actions are queries to search engines. Search plans are constrained by the resources available: how much time should be allocated to the query and how much of the Internet's resources should be consumed by it. Consequently, in this view, a plan to gather information for a specific query consists of a sequence of parallel queries to search engines where the user gets to decide at intermediate points whether further searching, and thus resource consumption, is necessary or desirable.

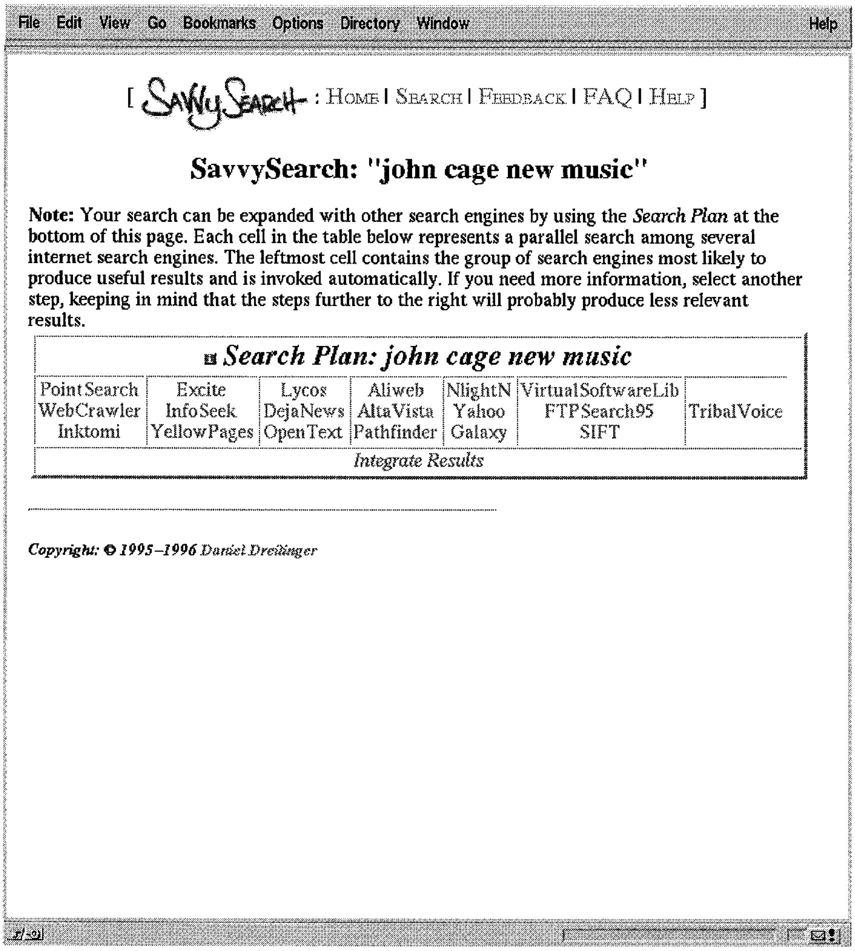
The search plan facilitates user control of parallel searching. SavvySearch proposes an ordering over all search engines in its available set, starts the first step, and then presents the remainder of the plan for the user to decide what to do next. Figure 3 illustrates an example of a search plan generated when 19 search engines were included. Each step normally includes between two and five search engines. After the set of search engines are ranked, they are divided into steps (or groups). Users are encouraged to select another step in the plan when the first fails to find the object of their search. The number of search engines per step, or concurrency value, is inversely related to the load of the machine that handles the query to SavvySearch[18] and the estimated network traffic. Thus, late-night users experience a higher degree of parallelism.

3.2 Metaindex Dispatch Approach

Dispatch is complicated by four issues. First, the corpus is not directly available; the Web is indexed by the other search engines. Second, both general and specific search engines comprise the search engine set; thus their expertise will vary widely. Third, the capabilities of the search engines change regularly; in particular, their indexes are updated. Fourth, to be a good citizen of the Web, resource consumption must be balanced against results quality.

These four issues are resolved through three mechanisms. First, a *metaindex* tracks experiences (i.e., successes and failures) in dispatching queries to the search engines. Second, to maximize the quality obtained for the effort expended, search engines are ranked based on the information in the metaindex as well as recent data on search engine performance. Third, the degree of parallelism and thus the resource expenditure are dictated by

[18] At present, queries to SavvySearch are distributed across five different machines: one SPARC 10, two SPARC ELCs, and two IBM RS6000s.


File Edit View Go Bookmarks Options Directory Window Help
[SavvySearch : Home | Search | Feedback | FAQ | Help]

SavvySearch: "john cage new music"

Note: Your search can be expanded with other search engines by using the *Search Plan* at the bottom of this page. Each cell in the table below represents a parallel search among several internet search engines. The leftmost cell contains the group of search engines most likely to produce useful results and is invoked automatically. If you need more information, select another step, keeping in mind that the steps further to the right will probably produce less relevant results.

Search Plan: john cage new music

PointSearch WebCrawler Inktomi	Excite InfoSeek YellowPages	Lycos DejaNews OpenText	Aliweb AltaVista Pathfinder	NlightN Yahoo Galaxy	VirtualSoftwareLib FTPSearch95 SIFT	TribalVoice

Integrate Results

Copyright: © 1995–1996 Daniel Dreilinger

Fig. 3. *Search Plan* component of the SavvySearch user interface. Users can click on any step in the plan.

current machine and network conditions. These three mechanisms are described in the remainder of this section.

3.2.1 *Metaindex of Prior Query Experience.* As a substitute for the lack of direct access to the corpus, SavvySearch's metaindex, similar to that described by Gravano et al. [1994], tracks the effectiveness of each search engine in responding to previous queries. The metaindex is a $t \times n$ matrix, where t is the number of terms that have been used, and n is the number of search engines. Obviously the metaindex grows as new terms are encountered.

A cell in the index summarizes the past performance of submitting the query term to the search engine. Values are signed numbers: positive numbers indicating good performance and negative indicating poor. The magnitude of the values suggest how well or poorly engines have performed.

The performance of individual search engines is expected to change over time as their indices are updated and their algorithms improved. Consequently, the association between expertise and search engine is likely to change as well. Thus, the metaindex is constructed from passively accumulated user feedback[19] over long periods of time. The metaindex is updated at the end of each day by processing the logs of system performance. Stemming and case stripping were applied to all query terms before adding them to or looking them up from the metaindex; no *stopwords* were removed.

Effectiveness values in the metaindex are updated according to two types of events: *No Results* and *Visits*. A No Results event occurs when one of the queried search engines finds no relevant documents; a Visit event occurs when a user follows one of the links suggested by a search engine. No Results is viewed negatively based on the intuition that if a search engine fails to produce results for a query, then that search engine is probably not a good place to search for terms contained in the query. Of course, search engines can provide uninteresting or irrelevant links. Thus, Visits is viewed positively based on the intuition that if a user follows a link it is relevant or at least interesting.

These two events form the basis of two heuristics used in a simple weight adjustment scheme for the metaindex. With a No Results event, the metaindex is adjusted by decrementing the values of the query terms for the particular search engine(s) that produced the relevant result. For multiword queries, a single unit of weight adjustment is amortized among the query terms. Thus, if no links are returned for a three-word query, each term is decreased by ⅓. With a Visit event, the metaindex is incremented by the same amortizing scheme.

The intuition behind the resulting values is that higher positive values represent a tendency of a search engine to produce interesting results when presented with queries containing that word, and negative values represent a tendency of a search engine to fail to produce results for queries containing that word. Thus, a negative score will cause a search engine to be avoided as being a waste of resources.

3.2.2 *Search Engine Ranking for a Query.* Search engines are ranked by predicting those which are most likely to return useful results. Utility has two components: whether a search engine has, over the long term, done well on the words in the query and whether the search engine has recently returned results quickly. Long-term performance is computed as a query score using a weighting algorithm similar to a classic IR approach. The

[19] An earlier version of the system was updated base on *actively* accumulated feedback. Unfortunately, as will be discussed in the evaluation section, the accuracy and veracity of such feedback is suspect.

query score is then adjusted to reflect recent search engine performance data on the expected waiting time and number of results.

In IR, documents are often ranked for retrieval by combining term frequency and inverse document frequency (i.e., $tf \cdot idf$). Symbolically, we can represent the term frequency of term t in document d using $f_{t,d}$. A formula often used for calculating idf is $\log(N/f_t)$, where N is the total number of documents in the corpus, and f_t is the number of documents in which term t appears at least once. Because longer documents will tend to be counted as more relevant simply because of having more words, the basic ranking formula can be augmented by factoring in document length. Thus, one version of a formula for ranking the suitability of document d for query q is

$$\mathrm{weight}(q, d) = \sum_{t \in q} \frac{f_{t,d} \cdot \log(N/f_t)}{\sqrt{|d|}}$$

which sums the $tf \cdot idf$ of each term t in the query q and normalizes by the square root of document d's length.

We adapted this $tf \cdot idf$ weighting to a metasearch context, in which search engines are treated as documents. The query score for query string q and search engine s ($Q_{q,s}$) balances the metaindex value against term and server ubiquity. Thus, to adapt the IR weighting formula, we substituted metaindex value ($M_{t,s}$ for a query term t and search engine s) for term frequency, inverse search engine frequency (I_t for term t) for inverse document frequency, and the absolute value of all metaindex values for a search engine (T_s for search engine s) for document length, as follows:

$$Q_{q,s} = \sum_{t \in q} \frac{M_{t,s} \cdot I_t}{\sqrt{T_s}}.$$

The query scores are normalized, so that the highest ranking score becomes 1 to facilitate later adjustment.

Inverse search engine frequency I_t is computed in a similar manner to inverse document frequency:

$$I_t = \log \frac{N}{f_t},$$

where N is the total number of search engines, and f_t is the number of metaindex entries for term t which have a positive search engine score. This computation is based on the observation that the more ubiquitous a term is, the less apt it is to perform as a good discriminator. In other words, obscurer query terms should be weighted more when computing the relative search engine rankings.

Because search engines may go down or get swamped, the scores are adjusted by incorporating recent performance information for each search engine. The five most recent queries for search engine s are used to compute the average number of hits (links returned in response to a query) and average response time. If these values fall within guidelines for normal operation—an average number of hits greater than a hit threshold (h_{thresh}) and an average response time of less than a time threshold (t_{thresh})—then no adjustments are made. However, when the guidelines are violated, that search engine's rank is lowered using a quadratically increasing penalty. Thus, if the average number of returned hits h for search engine s drops below h_{thresh}, the penalty is computed as

$$P_{s,h} = \frac{(h_{thresh} - h)^2}{(h_{thresh})^2}.$$

Similarly, when search engine s's average response time t exceeds t_{thresh} seconds, the penalty is computed as

$$P_{s,t} = \frac{(t - t_{thresh})^2}{(t_{timeout} - t_{thresh})^2}.$$

The thresholds are set by default to $h_{thresh} = 1$ and $t_{thresh} = 15$ seconds. The maximum allowed response time before a timeout occurs ($t_{timeout}$) is 45 seconds.

These penalties are subtracted from the query score only when the threshold is exceeded. Thus, the overall rank $R_{s,q}$, for search engine s and query q is

$$R_{q,s} = Q_{q,s} - (P_{s,t} + P_{s,h}).$$

3.2.3 *Calculating Concurrency.* The purpose of concurrency calculation is to reduce the resources demanded by SavvySearch in periods of high network and machine demand. Concurrency is inversely proportional to estimated query cost: the more it costs to submit search engine queries at present, the fewer search engines will be queried. Concurrency is computed from three cost variables: expected network load, local CPU load, and query discrimination value.

Expected Network Load. This is based on the traffic observed on SavvySearch servers (number of queries per unit time) in the past at this time of day. The value is computed by referring to a lookup table created from the Web server log files (Figure 4). During periods of low network load (e.g., 3:00 a.m.), a high value (up to two) is added to the concurrency, and vice versa.

Local CPU Load. This is computed using the *UNIX uptime* command; lower loads contribute more, and vice versa. Our servers are not dedicated to SavvySearch; thus this value responds to other user demands. If many searches arrive simultaneously, or the local CPU is required for other tasks during a typically slow period, concurrency is temporarily lowered.

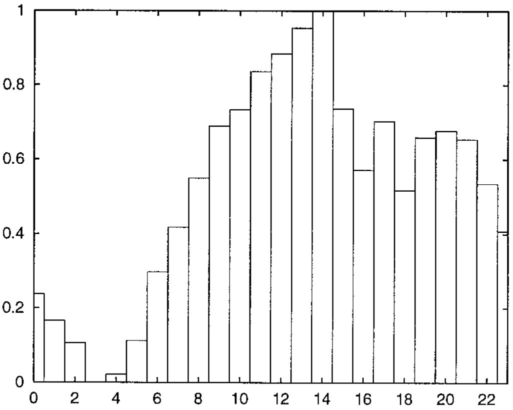


Fig. 4. Relative request traffic throughout a typical day, normalized so that the highest traffic time is 1.0.

Discrimination Values. These attempt to assess how much search effort is likely to be needed to find a satisfactory response by measuring how specific or general the query is. If a term has a relatively large amount of metaindex data, the query presumably represents an area that many search engines cover, and fewer search engines will need to be queried. The discrimination value is calculated by dividing the total number of references counted for the term by the number of references of the most frequently referenced term in the database (i.e., the same sum but for the most dominant term in the metaindex). An average is used for multiterm queries. In practice, the discrimination value only deviated from its usual near-zero value for extremely common words.

4. EVALUATION EXPERIMENTS

The design of SavvySearch, as described in the previous section, was developed over a period of a year. As indicated in Section 2, we started with a list of desirable qualities of a metasearch agent. For example, it should ideally select a small subset of highly relevant search engines to which a query should be forwarded, and furthermore, the selection process should have as little impact on the user as possible. We constructed a version of the system that included the search plan interface, but a simpler version of the metaindex. We ran a pilot study on that system to probe how well it fit some of the criteria. Based in part on the results of that study, we designed the more complicated metaindex scheme described in the previous section. Then, we ran a month long study to test the efficacy of the new metaindex scheme. Finally, we compared the performance of the metaindex-based system to a simpler categorical-based system.

4.1 Pilot Study

The primary objective of the pilot study was to determine whether search engines were being selected appropriately. Other metasearch engines search their entire set of search engines. By limiting the number searched, we had to be sure that we were still returning interesting information.

Search engine selection is manifest to the user in the search plan and the automatic execution of the first step. Thus the pilot study was intended to answer two questions:

(1) *Does plan order matter?* More precisely, are plan steps being ordered so that higher-quality results are returned early in the plan?
(2) *Is the selection strategy adequate?* More precisely, does quality depend upon which search engines are included in each plan step?

We collected data from four versions of SavvySearch. The versions varied along two dimensions: ordering of search engines within a plan and selection of the first group of search engines queried. In all cases the degree of parallelism was based on load.

Approach A. A predecessor to the basic design described in Section 3.1, Approach A involves queries translated to parallel search plans where the search engine's rank is based on a metaindex and recent search engine performance. This earlier design was simpler in that the metaindex contained only positive weights and remained static throughout the data collection.

Approach B. This is as in Approach A except that the step to be executed first was selected randomly; thus the first display was generated from any step in the plan.

Approach C. This also is as in Approach A, except that the search engines were randomly selected without replacement for inclusion in each step of the plan.

Approach D. Last, this too is as in Approach A, except that the first plan step executed was selected randomly, and the search engines were selected randomly.

Approach A is the original design. We varied both ordering and first step because we query sets of search engines simultaneously. If the ranking were relatively correct but completely backward, then Approach B would be best because it maintains relative ranking, but changes order of first execution. If the ranking is not even relatively correct, then random selection, as in Approaches C and D, should be indistinguishable or perhaps even better. Approach D is included as a control.

Each variant was run for about two days—enough time to collect data on at least 2500 queries. The user interface was identical for each one; users were told that a search plan would be constructed in which expected usefulness of the steps was displayed from left to right.

Table I. Summary of Data from Pilot Experiment

Length of study	8 days (September 1995)
Number of search engines	14
Incoming queries processed	10,575
Visits events	19,072
Yes-No quality feedback events	2,082

Table I summarizes the data. We measured quality in two ways: number of visits per query (link-ratio)[20] and self-reported user satisfaction. The link-ratio was collected passively. For user satisfaction, we asked the user to indicate whether the results returned by SavvySearch were satisfactory (*yes-no quality feedback* events); lack of answer was separated from a negative response. As can be noted from the summary data, one problem with user-reported satisfaction is that not everyone took the time to respond; the response rate was about 20%.

As expected, Approach A exhibited the highest quality performance on both measures, and Approach D was the least satisfactory. On average, users followed 2.0 links per plan step for Approach A, 1.76 links per step for Approach B, 1.89 links per step for Approach C, and 1.55 links per step for Approach D. The self-reported satisfaction was 72% for Approach A, 60% for Approach B, 65% for Approach C, and 60% for Approach D. On average, users took 1.42 plan steps per search.

Does Plan Order Matter? We analyzed the data by constructing contingency tables with *quality* as the dependent variable and *plan step* as the independent, for each approach and plan length. Quality was indicated by the number of Visits per each search request (step in the plan executed for some query). Plan step indicates which step in the search plan was executed; as users had control, several plan steps were often executed. Plan length indicates the number of steps in the constructed search plan and varied from two to seven, depending on the current level of parallelism. For example, the contingency table for Approach A and plan length 4 appears in Table II; note that the number of requests is larger for plan step of one due to Approach A automatically executing the first step.

Quality was expected to depend on plan step for Approaches A and B, but not for C and D because C and D simply selected the composition of steps randomly. In fact, for all but one plan length for Approach A and all plan lengths for Approach B, we found a statistically significant effect, using a chi-square test, of plan step on quality ($P < 0.01$). For Approach C for all but one plan length, and D for two lengths, we found no significant effect of plan step on position (in these cases, the lowest was $P < 0.20$). The one exception in Approach A was plan length 7 in which every step included

[20] Link-ratio is the predecessor to the Visits measure from metaindex updating. We substituted Visits later because it is less sensitive to concurrency; it measures the number of links followed per outgoing search request, as opposed to number of links followed per incoming search request.

Table II. Example Contingency Table for Testing Effect of Plan Step on Quality (Approach A and Plan Length 4)

Plan Step	1	2	3	4
Requests	1684	227	149	96
Visits	3234	289	71	42

only two search engines.[21] In both C and D, in the cases with the most data (lengths 3 and 4), a decline in links followed due to position was enough to be significant. This could have been due to a placebo effect of assuming that the ordering is good and not looking too closely at the contents of later steps.

Does Search Engine Ranking Improve Performance? We addressed this question by comparing the performance of the approaches that did rank to those that selected randomly. Approach A was compared to Approach C, and Approach B was compared to Approach D because each pair used the same method for determining which plan step to execute first. Because we allowed plan length and plan step to vary during the data collection, we partitioned the data by these factors and analyzed only the most common plan length (four steps).

For each pair of approaches and each plan step, a two-by-two contingency table was constructed with a row for each approach, one column for number of queries, and one column for links followed. Quality was significantly higher for Approach A than for Approach C on steps (1), (2), and (4) ($P < 0.01$). While higher on step (3), the difference was not statistically significant ($P < 0.37$). For Approaches B and D, quality was significantly higher for three steps ($P < 0.05$); for step (2) it was higher but not significantly ($P < 0.06$).

Conclusions from Pilot Study. The results suggest that the metaindex-based dispatch approach is viable: users prefer the basic approach using the metaindex ranking and follow more links found from the beginning of a search plan than from the end. Past query success knowledge can be used to improve performance of future queries.

The differences in quality among each of the approaches was enough to be significant, but was not profound. Most likely, the lack of profound difference was due to the inclusion of a large proportion of general spider-based search engines (roughly half, depending on how one determines generality). Their proportion makes them likely to be included in a randomly selected first step, and their generality makes them likely to return some link worth checking.

One of the characteristics that distinguishes SavvySearch from other metasearch engines is its reasoning about resources. By creating a "good Web citizen," we are potentially sacrificing user satisfaction because fewer

[21] One hypothesis is that a plan length of 7 with a parallelism of only 2 may force too much emphasis on particular search engines.

sources are queried. We tried to assess the effect of decreasing resources on user satisfaction by testing the dependence of plan length on links followed. To control for the selection strategy and for the automatic execution of some plan step, we used only data from Approach A. We constructed two contingency tables: one in which the cells reflected only the links followed for the automatically executed first step and one in which the cells summed the data for all plan steps. The tables had plan length as the independent variable (with values of 3, 4, 5, and 7) and quality as the dependent. Approach A showed no significant effect of plan length on quality in either case ($P < 0.49$ and $P < 0.39$, respectively). These results suggest that user satisfaction is not significantly diminished by reducing the parallelism of the search. However, these results are inconclusive because we did not control for the amount of time required to return results (i.e., some users may not care how long they wait).

Regarding a lesson for future experiments, we also learned that users need to be carefully considered in experiment designs. Experiments should indicate whether the system is "experimental"; otherwise, long-time users get irate when the system does not behave as expected. Self-reported relevancy ratings are suspect, given the subject base. In addition to the problems of low response rate, we found evidence of ballot stuffing and other attempts to have one's vote count more.

4.2 Incremental Metaindex Experiment

Based on the positive results of the pilot study, our enhancements to the metaindex scheme were to be more responsive to changing capabilities of the search engines: incorporating a negative weight adjustment of the metaindex for lack of results and updating the metaindex regularly. Because the new version depends critically on the metaindex, we examined performance over a period of one month, looking for performance improvements that would accrue from the incremental updating. We focused on two questions:

(1) Can the metaindex agent learn to make more effective searching decisions over time as it accrues more knowledge?

(2) How quickly does it learn, and how much knowledge is required to make intelligent search engine selection decisions?

4.2.1 *Experiment Control and Data Collection.* We collected data over a 29-day period on a version of SavvySearch that included the metaindex as described in Section 3. Because minimum performance constraints prohibited beginning with an empty metaindex,[22] an initial metaindex was constructed from one-day's worth of data that was collected while the system operated with the metaindex from the pilot study. When search plans were made, search engines for which the system had no data were

[22] Some of the smaller search engines would not tolerate the high volume of traffic that SavvySearch would generate if decisions were made completely at random.

randomly assigned ranks close to zero. Thus, they appeared after those that had an overall positive score, but before those with negative scores.

Except for the updated metaindex, we held the implementation constant. For example, although the roster of participating search engines has been in flux as new servers are discovered and old ones discontinue, we kept the roster the same during the experiment by halting addition of new ones and conducting the experiments over a short-enough time that none ceased to exist.

As with the pilot study, data were collected on the queries and their results. Traditional IR measures such as *precision* and *recall* would aid in comparing SavvySearch to other systems; however, performing the required expert relevance computations would be too expensive. Computing recall for a given query requires knowledge of *all* relevant documents on the Web, whether retrieved or not. Computing precision requires evaluating every single link returned.

Instead, the two key measures are the relative proportion of links visited per outgoing search request (Visits) and the proportion of occasions in which a search engine produced no results for a given search request (No Results). Clearly, No Results should be minimized in order to attain the goal of efficient use of resources. While Visits seems like a passive measure of quality or at least an indirect measure of interest, it is difficult to know whether this is actually a measure of low precision or high recall. For example, if a user visits several irrelevant references before finding a truly relevant one or even before giving up on finding a relevant one, it still registers as a high Visit value. However, if users consistently found no relevant links, they would be unlikely to continue using the system.

Both the Visits and No Results measures can be applied to individual search engines and a global average. Additionally, they can be applied to only a subset of queries that fit some constraint, such as containing words with a specific amount of associated knowledge. The next section shows how this versatility will be used to facilitate several different types of analysis. We also recorded average response times and average number of results (using the same figures used for computing performance penalties, described in Section 3.2.2) at 15-minute intervals throughout the duration of the study.

The Visits and No Results metrics are closely related to the events used to construct the metaindex. Consequently, the evaluation is not completely independent of the tuning process; unfortunately, we have not found independent evaluation measures that can be reasonably calculated.

4.2.2 *Experiment Results.* Table III summarizes the month-long metaindex experiment. The average number sent to each of the 20 individual search engines was 54,581, but the actual numbers were highly variable—the minimum number of queries sent to a single search engine was 12,604; the maximum was 174,046. The metaindex ultimately contained 46,568 unique word stems, as constructed from the Visit and No Result events.

Table III. Summary of Data from Metaindex Experiment

Length of study	29 days (December 17, 1995–January 14, 1996)
Number of search engines	20
Number of unique word stems used	46,568
Number of query terms	420,592
Incoming queries processed	211,887
Outgoing query requests made	1,091,630
Visit events recorded	437,243
No result events recorded	154,962

The analysis was divided into two parts: determining whether selection improves as the metaindex is constructed and determining the relationship between knowledge and selection.

Performance Improvement over Time. To determine whether the performance improved as the metaindex got larger, we compared performance at the beginning of the experiment to that at the end. In particular, for each search engine and evaluation measure, we ran a two-sample t-test comparing data from the first seven days to data from the last seven days. We expected Visits to increase and No Results to decrease.

The results were mixed. Of the 42 t-tests conducted (20 search engines plus 1 total with the 2 measures analyzed separately), 26 exhibited the expected relative difference of means (*early* < *later* for *Visits*, *early* > *later* for *No Results*), and 9 showed a significant improvement ($P < 0.05$).[23] Three tests were significant at the $P < 0.01$ level. We had to disregard 1 search engine, SIFT, because it was not responding for most of the days at the end of the study.

Four other search engines, DejaNews, Galaxy, LookUP, and Open Text, exhibited a significant decline ($P < 0.05$) in performance on the No Results measure. We hypothesize that the declines are due to the large amount of data for these search engines and the fact that the No Results measure is relatively static for search engines with huge vocabularies, such as these. The slopes of regression lines fitted to the data were very close to zero in most of these cases.

Performance on most of the search engines fluctuated considerably during the period.[24] Consequently, the 29-day period may have been too short to observe conclusive improvements.

Knowledge-Based Evaluation. Throughout the experiment, the metaindex increased both in the number of entries and the quantity of data associated with the entries. Each time a user followed a returned link, or a search engine failed to produce results, the information was incorporated

[23] Search engines significant according to the Visit measure included Inktomi, Pathfinder, Tribal Voice, and Yahoo, in addition to the overall total. Excite, Inktomi, Point, and Yahoo were significant according to No Results.

[24] See Dreilinger [1996] for some examples of the performance of individual search engines.

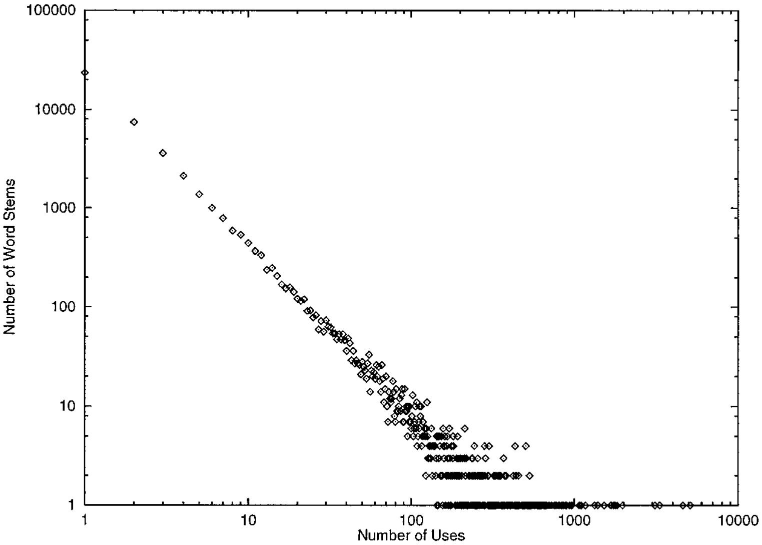


Fig. 5. Histogram of word usage frequency over the course of the 29-day experiment; scale is logarithmic.

into the metaindex. The question is "*Were the entries adjusted appropriately so that additional knowledge meant better performance?*" To determine this, we looked at how much knowledge was available and whether its inclusion improved performance.

The rank orderings of query terms supplied by users in the experiment agreed with Zipf's observations of word frequency [Salton 1989, pp. 106–107]. Because the histogram of word usage (Figure 5) is log-linear, we analyzed the data in increasingly large groups according to word usage.

While only 5.6% of all query terms were used just once, they contributed 50.4% of the unique stems in the metaindex. A large portion of the unique stems were misspellings, proper names, and non-English words; about 58.8% of all unique words were not found in an English spelling dictionary included in a standard UNIX distribution. Table IV summarizes the numbers for five frequency groupings; each column sums to 100%. Words with many occurrences (101–1000) accounted for the highest proportion of all word uses.

Metaindex adjustment assumes that words will be seen repeatedly over the course of metaindex construction. In fact, as Table IV indicates, 94.4% of words had been used previously by the end of the 29-day period. However, the summary numbers do not indicate how quickly that level was obtained. Figure 6 reports the proportion of words that, at the time they were used, had been seen previously a set number of times. The four lines denote 1 or more, 10 or more, 100 or more, and 1000 or more previous uses.[25]

[25] Note that these figures are somewhat approximate. Network exigencies dictated bootstrapping by using a small seeder metaindex which was gathered from a single days's system use. The initial metaindex was not counted in any of the word frequency analyses.

Table IV. Proportions of Word Uses, Unique Uses, and Unusual Spellings of Terms in the Metaindex Partitioned by the Number of Times the Word had been Encountered During the Experiment

Number of Occurrences	% of all uses	% of unique uses	% of unusual spellings
−1	5.6%	50.4%	58.8%
2–10	15.8	38.4	36.2
11–100	32.1	9.7	4.7
101–1000	38.3	1.4	0.42
1001–10000	8.1	0.03	0.002

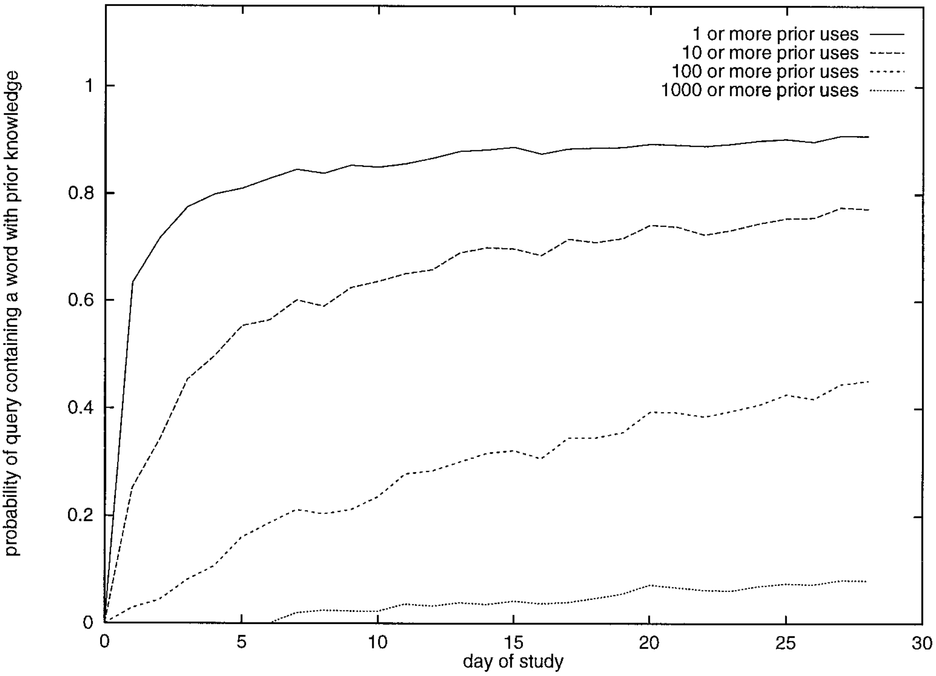


Fig. 6. Accumulation of word usages in the metaindex over the 29-day period.

With some minor fluctuations, the curves shown in Figure 6 increase throughout the period, but at substantially different rates. On the last day of the study, for example, 9.1% of the queries still contained a term not previously encountered. On the last day, about 77.2% of all unique query terms had been used at least 10 times; 45.1% had been used at least 100 times; and 8% had been used more than 1000 times. It took six days before the first term reached 1000 uses, but less than a day before more than half of the queries contained a word that had been seen before.

From Figure 6, we can gauge how quickly word experience accumulates. We also need to know how well the system exploits the knowledge from that experience. We expect that as the number of prior examples of a word's use increases, Visits increases, while No Results decreases. Figure 7 illustrates this trend in performance as a function of how much knowledge

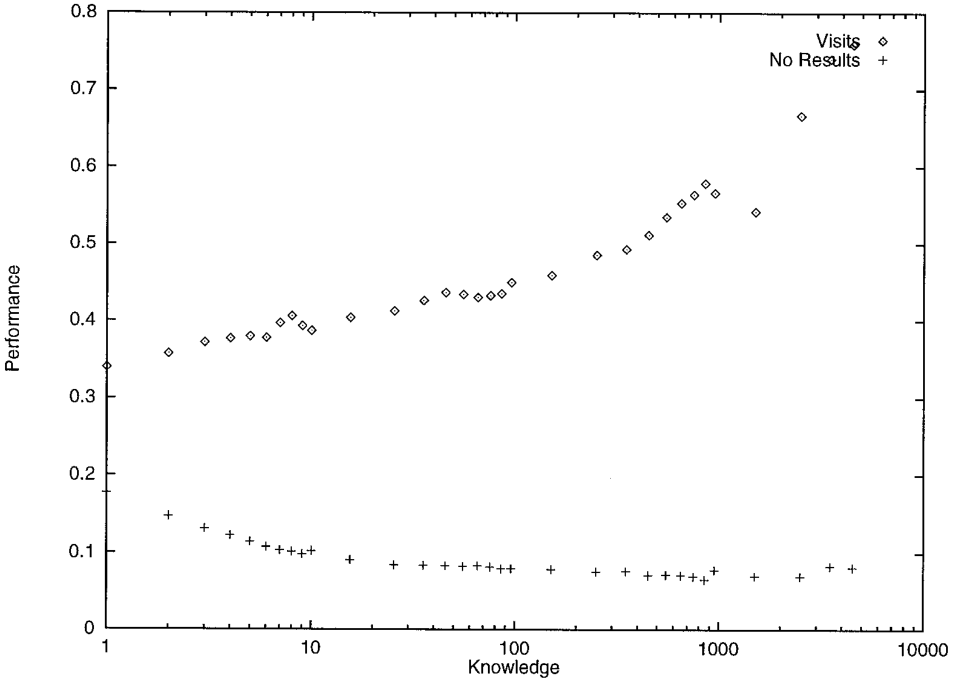


Fig. 7. Performance metrics, Visits and No Results rates, as a function of knowledge: number of previous encounters with a word in the query.

(prior uses of a term) the system has. Each point in the graph represents performance on queries that contained at least one word that had been experienced the indicated number of times previously. So the leftmost Visits point indicates that the mean Visits for those queries containing words that had been seen only once was 0.34. The "knowledge" axis is logarithmic to match the observed word usage pattern in Figure 5; each point incorporates the data starting from the previous point. As can be seen from the graphs, performance on No Results improves with the first 10 usages, but then levels off. Conversely, Visits seems to require many usages before it improves, but Visits continues to improve with additional usages. In the short term, tuning, as was done in this study, is more effective for minimizing the chance of not getting a response (and so wasting resources) and, in the long term, for maximizing the chance of getting a good response.

Experiment Discussion. The 29-day experiment shows that metasearch of the Web is a useful and viable application. If some of the search engines were truly comprehensive, metasearch might be unnecessary; however, the different algorithms employed by the search engines would probably still return different results. In the study, we managed to increase Visits and decrease No Results for some of the larger search engines by more carefully dispatching queries. At the end of the study, none of the search engines had been dropped in favor of the large search engines. Additionally, the data show that most query terms are repeated enough to direct metaindex tuning.

In fairness, in at least one case, we believe that some of the performance improvement was due to a change within the remote search engine itself. One of the search engines experienced a period of sporadic downtime followed by improvements on the evaluation criteria. This search engine could well have been upgraded during this period. Due to our lack of control over the external search engines, we cannot refute that some of the observed improvements could also have been due to changes made within the remote search engines themselves. However, it seems unlikely that the overall improvement would be fully due to across-the-board improvements in the search engine set.

Experiment control was problematic. Remote search engines vary in numerous ways. In the short term, their response time increases and decreases; in the medium term, they crash and are rebooted, and in the long term, their indices and even underlying search technology changes. User populations presumably varied over the course of the study. Users accessed the system using hundreds of different Web browser tools; most major browsers are evidenced by multiple versions across many platforms, in addition to a multitude of lesser-known tools. It was impossible to test the SavvySearch system with each of these tools; thus it is probable that there were incompatibilities with some of them. Collecting data over a long period should have mitigated the fluctuations in the individual search engines and ensured that the user population was at least representative.

4.3 Comparing Metaindex with Simple Preprogrammed Design

Previous sections showed that the metaindex search agent improves on a relative scale, but did not address its absolute performance. Because the few current metasearch engines are rather new and vary drastically in design and considerably in function, the comparison reported here is between two SavvySearch dispatch approaches: learned dispatch (metaindex) and preprogrammed (categorical). In addition to the 29-day metaindex experiment, a two-day experiment was conducted in which the metaindex approach was replaced with a simpler selection mechanism preprogrammed with our experience.

Categorical Selection Mechanism. We asked users to indicate up to three categories out of 10 possible as describing their query. We determined the categories based on our own experiences and the categories used by some of the search engines; the categories were WWW Resources, Software, People, Reference, Commercial, Academic, Technical Reports, Images, News, and Entertainment. We preprogrammed mappings of categories to search engines based on our own experiences with the search engines. The basic approach was to give each search engine a value of 2 for information that was central to the server's domain of expertise, 1 for domains where it could probably be useful (thus the large robot indices had many 1s), and 0

Table V. Summary of Data from Categorical Version Experiment

Length of study	2 days (January 27, 1996–January 28, 1996)
Number of search engines	20
Incoming queries processed	22,678
Outgoing query requests made	92,203
Visit events recorded	42,419
No result events recorded	10,942

otherwise (which occurred many times in the case of highly specialized search engines).

Data Collection. To facilitate comparison, we tried to eliminate all extraneous differences between the two SavvySearch versions and to hold constant any qualities that might otherwise vary. The mechanism used to make search engine selection decisions and the search form portion of the user interface were modified in the new version. The user interface included check-box-selectable categories and a note that up to three were to be chosen.

To meet user expectations as closely as possible, the two variants were accessed through separate URLs. Users were told that the metaindex was an experimental, new system. The categorical form was accessed from a location where users were told it was an improved version of a system they were already accustomed to (a much earlier version of SavvySearch). Thus, while the different addresses served to reduce user confusion and surprise, they also resulted in a form of uncontrolled variance: the user populations themselves.

Because less data were needed to obtain performance measures of the categorical system (its behavior did not vary with experience), data from 20000 queries were collected on just two weekend days for the categorical approach. We analyzed the resulting data for sensitivity to day-of-week and found no significant differences between weekdays and weekend days.

Comparison Results and Discussion. Table V summarizes the two-day categorical experiment. In terms of query throughput, the second experiment encompassed more queries per day than the longer study. So, while the duration of the shorter study was only about 6.9% of the longer one, the total query volume was about 10.7%. User demand for the system continually increased over the experimentation period; thus this study, which was performed later, had a higher *per diem* usage.

Over the two-day period, the categorical version averaged 46% on Visits and 12% on No Results, which is slightly better than the overall performance of the metaindex version during the 29 days (40% and 14% respectively). To determine how much knowledge is required before metaindex dispatch performs as well as the categorical, we can check Figure 7 for the point at which Visits and No Results are equivalent. The crossover point for Visits is between 100 and 200 previous word uses. In other words, for the metaindex to perform as well as the categories according to the Visit measure, queries must contain one or more words that have been used at least 100 times previously. The crossover point for the No Results measure is between four and five previous word uses. Thus, the metaindex requires considerable experience with a word before surpassing the categorical performance on the Visits measure, but only a few experiences with a term before surpassing No Results.

As with previous studies, control was an issue. In particular, the user population participating in the metaindex and categorical experiments was not the same. Because the interface differed, we needed to explain the differences by indicating that one was an experimental version.

These results are consonant with the conclusions suggested by the controlled ProFusion experiment [Gauch et al. 1996]. A carefully generated taxonomy may return superior results. From our experiment, we have determined that, for a large group of users, the categorical approach is superior to the learned metaindex scheme on Visits, our indirect measure of relevancy, unless considerable information is available about the query. However, because the categorical scheme is hand generated, it is not responsive to changes in expertise of individual search engines and may not be superior on No Results.

5. CONCLUSION AND FUTURE WORK

Since its inception in March, 1995, SavvySearch has garnered considerable positive attention (email from users as well as positive reviews in magazines and other Internet sites) and steadily increasing usage. At present, SavvySearch handles over 20,000 queries each day, the limit of what our machine resources can handle. Based on these outward signs of success and the experimental findings about the distribution of the queries, we conclude that metasearch adds value to Web searching.

The experimental findings suggest that a metaindex approach can be effective in making search engine selection decisions. However, the potentially large amount of knowledge required to make these decisions raises some question about the overall efficiency of the system. Initially, the categorical version is far superior to the metaindex version. With a small amount of term experience (only four or five uses), the metaindex surpasses the performance of the categorical version on the No Results; because temporary poor performance of a search engine is accommodated through the ranking penalty ($P_{s,h}$) rather than the query score, the metaindex will reflect long-term poor performance on a query term. However, considerable knowledge (100–200 uses of a term) are required for metaindex to surpass categorical on the Visits measure. These findings suggest that the metaindex approach is better at predicting where *not* to send a particular query with relatively little word knowledge.

Given continued increases in how many words have been repeated, one would expect that, at some point, metaindex Visits performance would exceed categorical performance. However, considering the substantial proportion of new words still arriving by the end (9.1%), the crossover point of

100–200 previous uses is inordinately large. Major robot search engines report dictionary sizes in the millions of terms; the SavvySearch metaindex contains less than 50,000 unique stemmed tokens. The metaindex approach requires a large user base with frequent updating; assuming a word acquisition rate of 2000 terms per day (the current approximate rate), one million search terms will be accumulated in just over a year.

Consequently, adequately supporting the metaindex scheme requires developing techniques for accumulating more words faster. For example, the metasearch agent could traverse search result links and add the documents' full text to the metaindex knowledge for the search engine that returned the links. Alternative metaindex weighting algorithms could also be tried. The current metaindex ranking algorithm is still somewhat ad hoc. However, while there is certainly room for further experimentation and ranking, a new ranking formula alone will probably not lead to a major advance.

The relative success of selecting search engines by category suggests that knowing the user's intentions or goals focuses search. The information needs of Internet users are widely variable. The large search engines generally take a one-search-fits-all approach, while the smaller, specific tools address more narrowly defined information needs. If various categories of information-gathering goals could be identified, they might be best met with stereotypical search plans. For example, when searching for a colleague's email address, it might be most productive to first search the three or four large email directories. If that does not work, move on to Usenet news search engines to look for articles written by the colleague, and finally, proceed with large search engines to find the colleague's Web home page (if it exists). On the other hand, a user who wishes to find introductory information about a new hobby might be best satisfied by searching the human-reviewed databases first, then the robot search engines, and then Usenet news. We anticipate that stereotypical search plans will be of most utility to users with considerable experience in searching who basically know what they want, but might still appreciate the convenience of focused simultaneous search provided by SavvySearch.

While critical to metasearch as conducted in SavvySearch, search engine selection is only one component of the system. The other two major components—interface agents and display/user interface—have not been evaluated or refined during development. Developing and maintaining interface agents is time consuming and somewhat tedious. Their development could be partially automated. For example, the metasearch agent could roam the Web on autopilot, looking for search resources. Automated discovery and incorporation of new search engines combined with an accurate selection scheme could produce a new searching paradigm. Current display agents could be significantly improved through more intuitive ranking, formatting, and verifying of results.

Searching is a personal activity. Users have differing interests, expectations, and styles. To further improve Web searching, we must focus on the user, most immediately on how to identify and serve their goals. For example, the relative importance of waiting time, thoroughness, accuracy, and resource consumption all should be incorporated into determining where and how much to search. The resources of the Web are vast, but hardly limitless. In SavvySearch, we have started to explore how the user's need to find information can be most effectively satisfied without unduly wasting Web resources.

ACKNOWLEDGMENTS

We wish to thank David Lewis for reading an early draft and for providing insightful comments. We hope we did justice to the excellent points made by the anonymous reviewers. We gratefully acknowledge the assistance of Lawrence Livermore Laboratory and the Computer Science Department at Colorado State University for providing machines to run SavvySearch and Wayne Trzyna and his staff for patiently keeping the machines running.

REFERENCES

BOWMAN, C. M., DANZIG, P. B., MANBER, U., AND SCHWARTZ, M. F. 1994. Scalable internet resource discovery: Research problems and approaches. *Commun. ACM 37,* 8 (Aug.).

BOWMAN, C. M., DANZIG, P. B., MANBER, U., SCHWARTZ, M. F., HARDY, D. R., AND WESSELS, D. P. 1995. Harvest: A scalable, customizable discovery and access system. Tech. Rep., Univ. of Colorado, Boulder, Colo.

DREILINGER, D. 1996. Description and evaluation of a meta-search agent. Master's thesis, Computer Science Dept., Colorado State Univ., Fort Collins, Colo.

EICHMANN, D. 1994. Ethical web agents. In *Electronic Proceedings of the 2nd World Wide Web Conference '94: Mosaic and the Web.* Elsevier, London. Available as http://www.ncsa.uiuc.edu/SDG/IT94/Proceedings/Agents/eichmann.ethical/ethics.html.

GAUCH, S., WANG, G., AND GOMEZ, M. 1996. Profusion: Intelligent fusion from multiple, different search engines. *J. Univ. Comput. Sci. 2,* 9 (Sept.).

GRAVANO, L., GARCÍA-MOLINA, H., AND TOMASIC, A. 1994. Precision and recall of GlOSS estimators for database discovery. In *Proceedings of the 3rd International Conference on Parallel and Distributed Information Systems (PDIS'94).* IEEE Computer Society, Washington, D.C.

SALTON, G. 1989. *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer.* Addison-Wesley, Reading, Mass.

SELBERG, E. AND ETZIONI, O. 1995. Multi-service search and comparison using the MetaCrawler. In *Proceedings of the 4th International World Wide Web Conference.*

SHELDON, M. A., DUDA, A., WEISS, R., AND GIFFORD, D. K. 1995. Discover: A resource discovery system based on content routing. In *Proceedings of the 3rd International World Wide Web Conference.* Elsevier, North Holland, Amsterdam.

WITTEN, I. H., MOFFAT, A., AND BELL, T. C. 1994. *Managing Gigabytes: Compressing and Indexing Documents and Images.* Von Nostrand Reinhold, New York.

YAN, T. W. AND GARCIA-MOLINA, H. 1995. SIFT—A tool for wide-area information dissemination. In *Proceedings of the 1995 USENIX Technical Conference.* USENIX Assoc., Berkeley, Calif., 177–186.

ZILBERSTEIN, S. 1995. An anytime computation approach to information gathering. In *Working Notes of the AAAI Spring Symposium Series on Information Gathering from Distributed, Heterogeneous Environments.* AAAI, Menlo Park, Calif.

Received August 1996; revised December 1996; accepted December 1996