Experiences with Selecting Search Engines using Meta-Search

Daniel Dreilinger
MIT Media Laboratory
Cambridge, MA 02139
e-mail: daniel@media.mit.edu
Web: http://www.media.mit.edu/ daniel

Adele E. Howe
Computer Science Dept., Colorado State University
Fort Collins, CO 80523
e-mail: howe@cs.colostate.edu
Web: http://www.cs.colostate.edu/ howe

Submitted: August 8, 1996

Revised: December 2, 1996

The material contained in this document has not been submitted for publication elsewhere.

Abstract

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 paper describes and evaluates SavvySearch, a meta-search engine designed to intelligently select and interface with multiple remote search engines. The primary meta-search 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 *meta-index* approach to selecting search engines by analyzing the effect of time and experience on performance. We also compared the meta-index approach to the simpler categorical approach and showed how much experience is required to surpass the simple scheme.

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 specialty 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 un-related 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 [14] and Point [23] search within smaller, human-reviewed collections of Web site descriptions. DejaNews [19] and the Stanford Information Filtering Tool [32, 31] specialize in searching archives of recent Usenet news articles. Tools such as FTPSearch [12] and C|Net's `shareware.com` [8] 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—a new database seems to appear daily to satisfy yet another specialized information need.

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 [25]. Consequently, working efficiently with the entire collection of search engines can be a challenge and burden for even the most experienced users.

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

[1] At this time, these include Lycos [20], WebCrawler[22], Infoseek[6], Open Text[5], Inktomi[16], Excite[29], and Alta Vista[21].

In this paper, 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: meta-search. We describe our design for a meta-search system and present results of evaluating its usage by a large number of users on the Web. The purpose of this paper is to help motivate the design of other meta-search systems and the development of new techniques by relating what we have learned from a deployed system.

For our own meta-search engine, we adopt the principle that Web resources should be used as efficiently as possible. Thus, the success of meta-searching depends critically on carefully selecting which resources to use. Our meta-search engine, SavvySearch [10], 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 *meta-index* 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 meta-search with that of some other approaches.

2.1 Information Retrieval

Information retrieval systems are software tools that help users find documents contained in a specific corpus or database. Information retrieval tools are becoming ubiquitous; they are used for finding scholarly information (e.g., CARL[2]) as well as for news dissemination, shopping, and many other recreational activities.

[2] CARL is the Colorado Alliance of Research Libraries' online catalog.

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 common $tf \cdot idf$ (Term Frequency times Inverse Document Frequency) ranking algorithm exploit two important qualities of natural language text to perform accurate retrieval [30, pages 141–148]:

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.

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 focussed databases—names and e-mail 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 only try the general search engines should the specialized engines fail on the query. Because we wish to accomodate 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 paper 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 Meta-Search Engines

Just as robot-based search engines were developed in response to rapid Web growth, meta-search engines are being developed in response to the increasing availability of conventional search engines. Meta-search engines, such as 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, meta-search engines cannot directly access the corpus of Web documents, but instead use a "corpus" of search engines.

2.3.1 Architecture of Meta-Search Engines

For comparison purposes, we view meta-search engines in terms of three components:

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 paper is devoted to analyzing one such mechanism.

Interface agents. These self-contained programs manage the interaction with a particular search engine. The interface agents adapt the user's query format to match the format of a particular search engine. Interface agents are also responsible for interpreting the diverse native result formats.

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

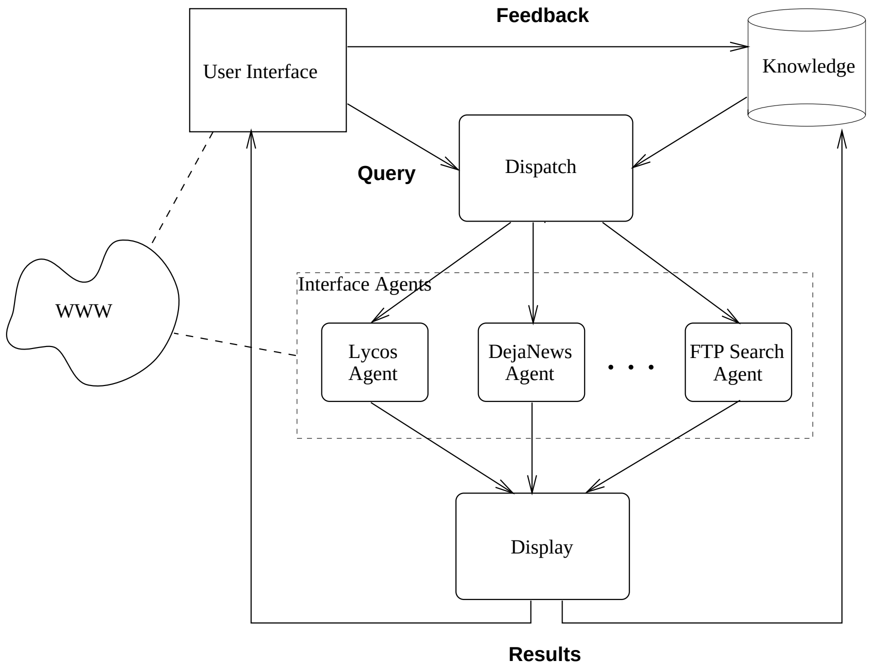

Figure 1: General framework for characterizing the capabilities of meta-search engines.

Figure 1 illustrates our idealized architectural view. A user submits a query via the meta-search 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 Meta-Search Engines

Over the past few years, many meta-search tools have been developed. We survey a few of the best known of these efforts.

The *GlOSS* (Glossary-of-Servers Server) project [17] uses a meta-index to direct simultaneous search of multiple databases (which they call the *text-database discovery problem*). A meta-index is constructed by integrating the indexes of each of the databases. For each database and each word, the number of documents containing that word is included in the meta-index. When a query is submitted to GlOSS, relevant databases are selected by using the meta-index to predict the ones that are likely to produce relevant results. One problem is that each of the search engines must cooperate with the meta-searcher by supplying up-to-date index information. As the number of databases increases, the administrative complexity may become prohibitive.

The Harvest system [3, 4] 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[28]. 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 [26, 25] meta-search 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 meta-search is to allow the user to completely direct query dispatch. Tools such as All-In-One[7], CUSI[18], `search.com` [9], Infi-NET's META Search [27] and InterNIC[1] 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—they must type their 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.

The ProFusion system [2] combines many features of the other meta-search 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, InforSeek, 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 ten search engines (the six large search engines, Meta-Crawler, 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 [15].

3 SavvySearch

SavvySearch is an experimental meta-searching 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 e-mailed hundreds of favorable comments and several dozen less-than-favorable ones; the daily usage has increased to over 20,000 queries per day.

Queries can be submitted to SavvySearch at `http://guaraldi.cs.colostate.edu:2000/`. The 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 (shown in 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 aribitrarily assigned a score of 0.5. We did not improve the

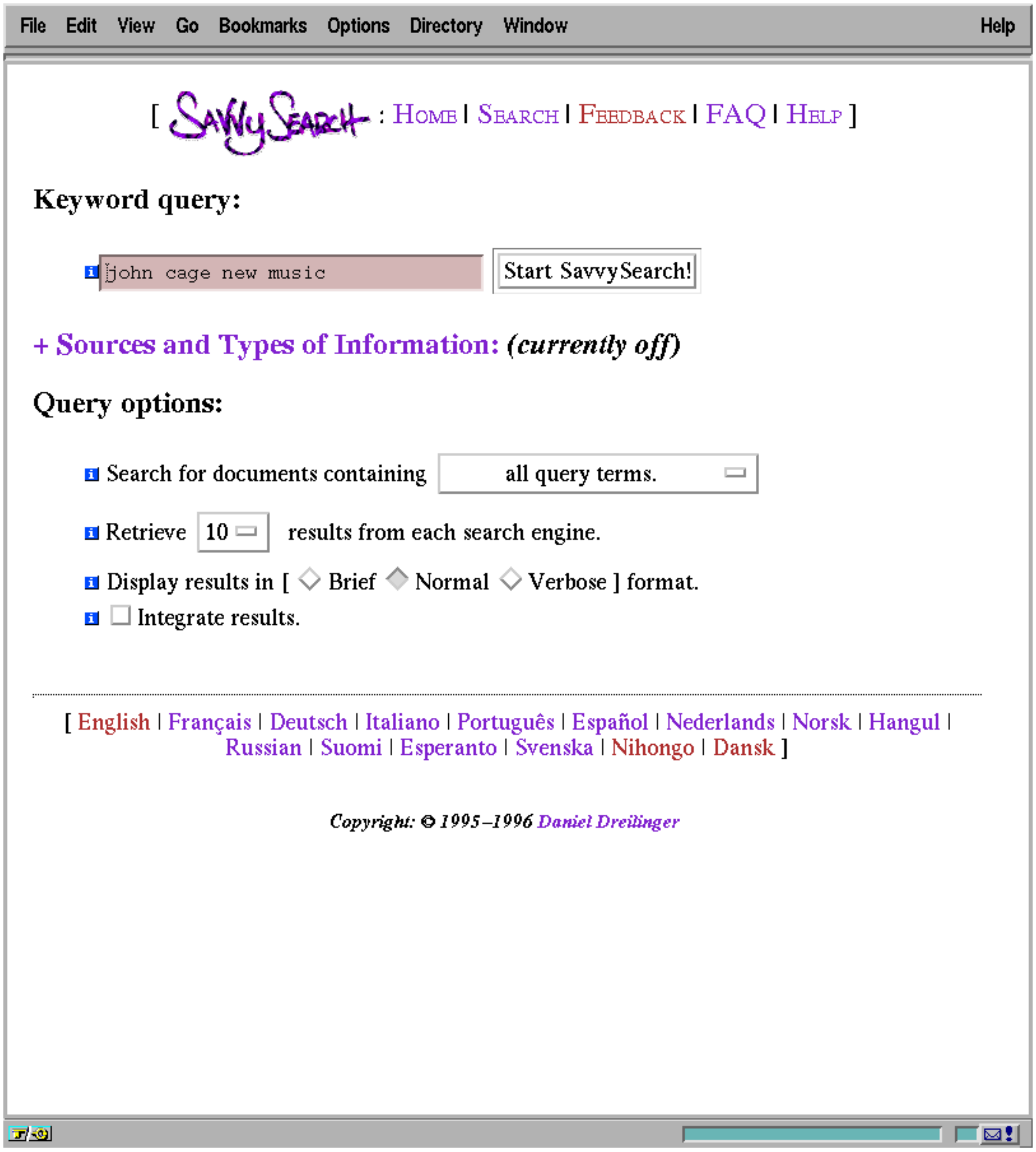

Figure 2: Query form for user entry of queries for the meta-index version of SavvySearch.

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 satisfy two conflicting goals: minimizing resource consumption and maximizing search quality[33]. 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 [13]. 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

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

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[3] and the estimated network traffic. Thus, late night users experience a higher degree of parallelism.

3.2 Meta-Index 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 *meta-index* 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 meta-index as well as recent data on search engine performance. Third, the degree of parallelism and thus the resource expenditure is dictated by current machine and network conditions. These three mechanisms are described in the remainder of this section.

3.2.1 Meta-Index of Prior Query Experience

As a substitute for the lack of direct access to the corpus, SavvySearch's meta-index, similar to that described in [17], tracks the effectiveness of each search engine in responding to previous queries. The meta-index is a $t \times n$ matrix, where t is the number of terms that have been used and n is the

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

number of search engines. Obviously, the meta-index 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 indexes are updated and their algorithms are improved. Consequently, the association between expertise and search engine is likely to change as well. Thus, the meta-index is constructed from passively accumulated user feedback[4] over long periods of time. The meta-index 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 meta-index; no *stop words* were removed.

Effectiveness values in the meta-index 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 meta-index. With a *No Results* event, the meta-index is adjusted by decrementing the values of the query terms for the particular search engine(s) that produced the relevant result. For multi-word 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 1/3. With a *Visit* event, the meta-index is incremented by the same amortizing scheme.

The intuition behind the resulting values is that higher positive values represent a tendency of

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

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

In information retrieval, 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 $log(\frac{N}{f_t})$, 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:

$$weight(q, d) = \sum_{t \; \epsilon \; q} \frac{f_{t,d} \cdot log(\frac{N}{f_t})}{\sqrt{\mid d \mid}}$$

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 meta-search 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 meta-index value against term and server ubiquity. Thus, to adapt the information retrieval weighting formula, we substituted meta-index 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 meta-index values for a search engine (T_s for search engine s) for document length, as follows:

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

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 meta-index 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, more obscure 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.

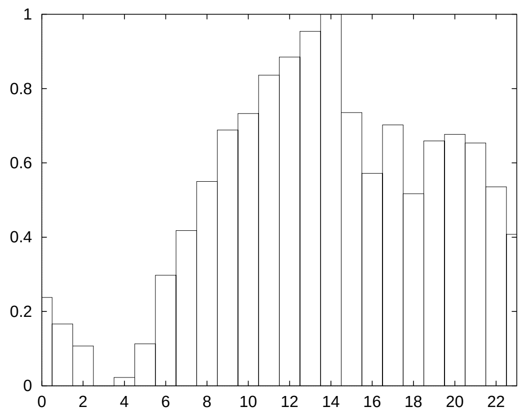

Figure 4: Relative request traffic throughout a typical day, normalized so that the highest traffic time is 1.0.

Expected network load 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 AM), a high value (up to two) is added to the concurrency, and vice versa.

Local CPU load 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.

Discrimination values 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 meta-index 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 meta-index). An average is used for multi-term 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 meta-search 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 meta-index. 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 meta-index scheme described in the previous section. Then, we ran a month long study to test the efficacy of the new meta-index scheme. Finally, we compared the performance of the meta-index 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 meta-search 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:

- Does plan order matter? More precisely, are plan steps being ordered so that higher quality results are returned early in the plan?
- 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 subsection 3.1: queries translated to parallel search plans where search engine rank is based on a meta-index and recent search engine performance. This earlier design was simpler in that the meta-index contained only positive weights and remained static throughout the data collection.

Approach B: as in 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: as in A except that the search engines were randomly selected without replacement for inclusion in each step of the plan.

Approach D: as in 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 backwards, 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 2,500 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 1 summarizes the data. We measured quality in two ways: number of visits per query (link-ratio)[5] 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%.

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 1: Summary of data from pilot experiment.

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.

[5] Link-ratio is the predecessor to the *Visits* measure from meta-index 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.

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

Table 2: Example contingency table for testing effect of plan step on quality (Approach A and plan length 4).

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 four appears in Table 2; note that the number of requests is larger for plan step of one due to Approach A automatically executing step one.

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 approaches 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 of seven in which every step included only two search engines[6]. In both C and D, in the cases with the most data (lengths three and four), 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.

[6]One hypothesis is that a plan length of seven with a parallelism of only two may force too much emphasis on particular search engines.

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 one, two, and four ($P < 0.01$). While higher on step three, 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 meta-index based dispatch approach is viable: users prefer the basic approach using the meta-index 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 between 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 meta-search engines is its reasoning about resources. By creating a "good Web citizen", we are potentially sacrificing user satisfaction because fewer 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).

As 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 Meta-Index Experiment

Based on the positive results of the pilot study, we enhanced the meta-index scheme to be more responsive to changing capabilities of the search engines: incorporating a negative weight adjustment of the meta-index for lack of results and updating the meta-index regularly. Because the new version depends critically on the meta-index, 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:

- Can the meta-index agent learn to make more effective searching decisions over time as it accrues more knowledge?
- How quickly does it learn, and how much knowledge is required to make intelligent search engine selection decisions?

Experiment Control and Data Collection We collected data over a 29 day period on a version of SavvySearch that included the meta-index as described in Section 3. Because minimum performance constraints prohibited beginning with an empty meta-index[7], an initial meta-index was constructed from one day's worth of data that was collected while the system operated with the meta-index from the pilot study. When search plans were made, search engines for which the system had no data were 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 meta-index, 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, we collected data on the queries and their results. Traditional information retrieval measures such as *precision* and *recall* would aid in comparing SavvySearch to other systems; however, performing the expert relevance computations required 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

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

Length of study	29 days (17 December 1995–14 January 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	$154,962$

Table 3: Summary of data from meta-index experiment.

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 meta-index. Consequently, the evaluation is not completely independent of the tuning process; unfortunately, we have not found independent evaluation measures that can be reasonably calculated.

4.2.1 Experiment Results

Table 3 summarizes the month-long meta-index 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 meta-index ultimately contained 46,568 unique word stems, as constructed from the *Visit* and *No result* events.

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

Performance Improvement over Time To determine whether the performance improved as the meta-index 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 one total with the two 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 ($P < 0.05$)[8]. Three tests were significant at the $P < 0.01$ level. We had to disregard one 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[9]. Consequently, the 29 day period may have been too short to observe conclusive improvements.

Knowledge-Based Evaluation Throughout the experiment, the meta-index 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 into the meta-index. 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[24, page 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 meta-index. A large portion of the unique stems were misspellings, proper names, and

[8]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*.

[9]See [11] for some examples of the performance of individual search engines.

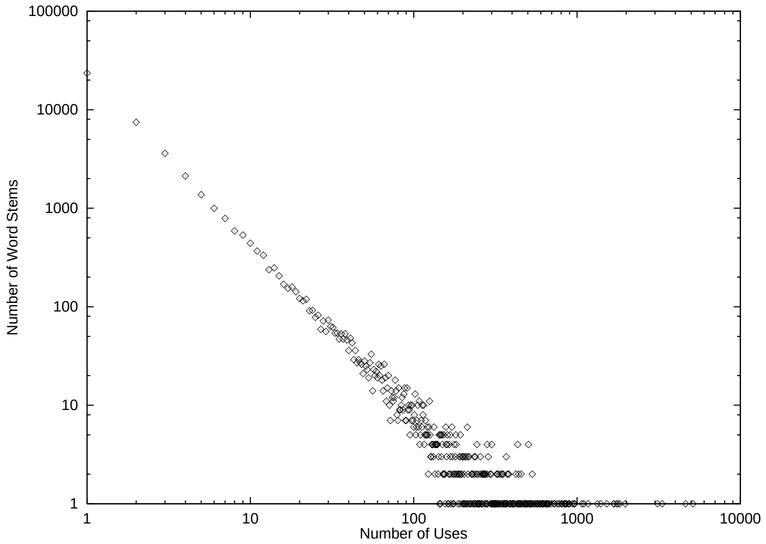

Figure 5: Histogram of word usage frequency over the course of the 29 day experiment; scale is logarithmic.

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

Table 4: Proportions of word uses, unique uses, and unusual spellings of terms in the meta-index partitioned by the number of times the word had been encountered during the experiment.

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 4 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.

Meta-index adjustment assumes that words will be seen repeatedly over the course of meta-index construction. In fact, as Table 4 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

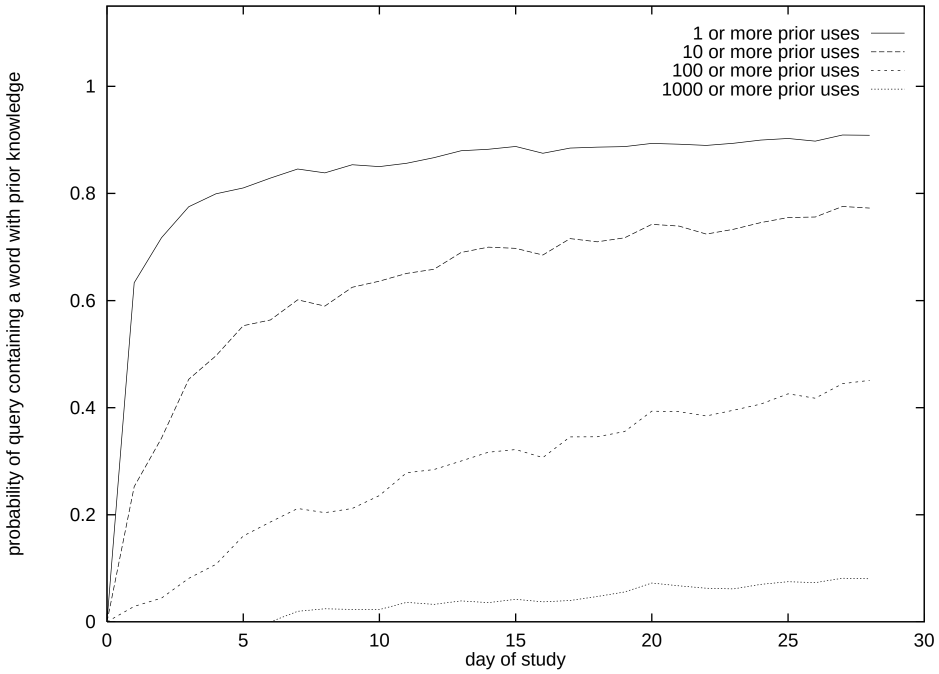

Figure 6: Accumulation of word usages in the meta-index over 29 day period.

seen previously a set number of times. The four lines denote 1 or more, 10 or more, 100 or more, and 1,000 or more previous uses[10].

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 1,000 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 number of prior examples of a word's use increases, *Visits* increases and *No Results* decreases. Figure 7 illustrates this trend in performance as a function of how much knowledge (prior uses of a term)

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

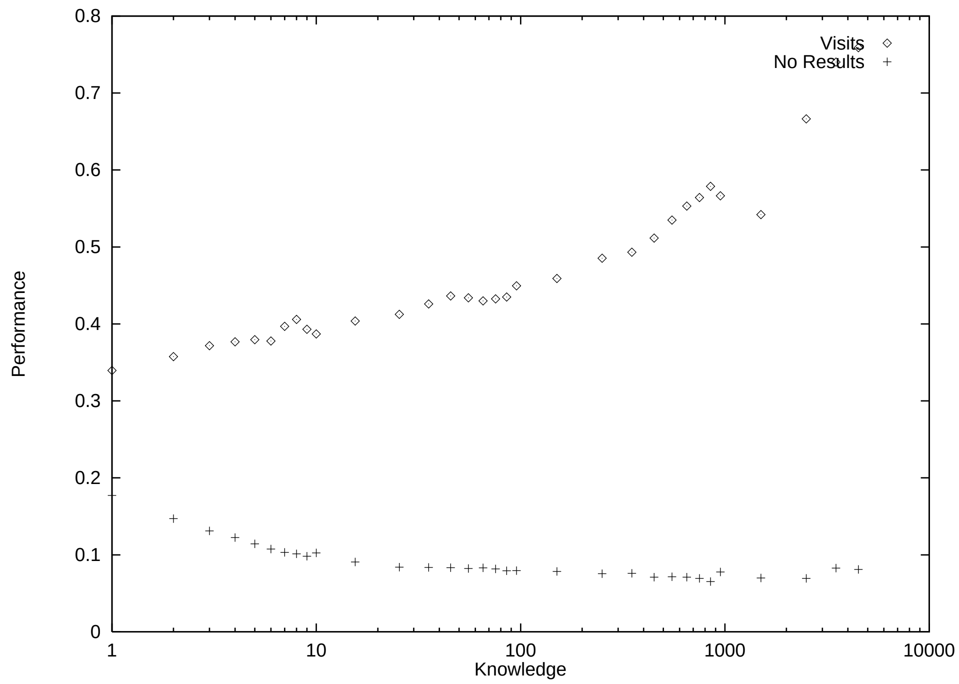

Figure 7: Performance metrics, *Visits* and *No Results* rates, as a function of knowledge: number of previous encounters with a word in the query.

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 meta-search of the Web is a useful and viable application. If some of the search engines were truly comprehensive, meta-search 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 shows that most query terms are repeated enough to direct meta-index 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 indexes 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 Meta-Index with Simple Pre-Programmed Design

Previous sections showed that the meta-index search agent improves on a relative scale, but did not address its absolute performance. Because the few current meta-search 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 (meta-index) and pre-programmed (categorical). In addition to the 29-day meta-index experiment, a two-day experiment was conducted, in which the meta-index approach was replaced with a simpler selection mechanism pre-programmed 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 pre-programmed 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 indexes had many 1s), and 0 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 meta-index 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 was needed to obtain performance measures of the categorical system (its behavior did not vary with experience), data from 20,000 queries was 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.

Length of study	2 days (27 January 1996–28 January 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

Table 5: Summary of data from categorical version experiment.

Comparison Results and Discussion Table 5 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 meta-index version during the 29 days (40% and 14% respectively). To determine how much knowledge is required before meta-index dispatch performs as well as the categorical, we can check Figure 7 for the point at which *Visits* and *No Results* is equivalent. The crossover point for *Visits* is between 100 and 200 previous word uses. In other words, for the meta-index 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 4 and 5 previous word uses. Thus, the meta-index 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 meta-index 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 ProFusion controlled experiment[15].

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 meta-index 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 (e-mail 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 meta-search adds value to Web searching.

The experimental findings suggest that a meta-index 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 meta-index version. With a small amount of term experience (only four or five uses), the meta-index surpasses the performance of the categorical version on the *No Results*; because temporary poor performance of a search engine is accomodated through the ranking penalty ($P_{s,h}$) rather than the query score, the meta-index will reflect long term poor performance on a query term. However, considerable knowledge (100–200 uses of a term) are required for meta-index to surpass categorical on the *Visits* measure. These findings suggest that the meta-index 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, meta-index *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 meta-index contains less than 50,000 unique stemmed tokens. The meta-index approach requires a large user base with frequent updating; assuming a word acquisition rate of 2,000 terms per day (the current approximate rate), one million search terms will be accumulated in just over a year.

Consequently, adequately supporting the meta-index scheme requires developing techniques for accumulating more words faster. For example, the meta-search agent could traverse search result links and add the documents' full text to the meta-index knowledge for the search engine that returned the links. Alternative meta-index weighting algorithms could also be tried. The current meta-index 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 e-mail address, it might be most productive to first search the three or four large e-mail directories. If that doesn't 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 focussed simultaneous search provided by SavvySearch.

While critical to meta-search 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 meta-search 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, formating 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.

6 Acknowledgments

We wish to thank David Lewis for reading an early draft and providing insightful comments. We hope we did justice to the excellent points made by the anonymous reviewers. The experiments 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. 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

[1] InterNIC Database Administration. InterNIC meta-search engines. http://ds2.internic.net/tools/meta.html.

[2] DesignLab at University of Kansas. ProFusion meta-search. http://www.designlab.ukans.edu/profusion/.

[3] C. Mic Bowman, Peter B. Danzig, Udi Manber, and Michael F. Schwartz. Scalable internet resource discovery: research problems and approaches. *CACM*, 37(8), August 1994.

[4] C. Mic Bowman, Peter B. Danzig, Udi Manber, Michael F. Schwartz, Darren R. Hardy, and Duane P. Wessels. Harvest: A scalable, customizable discovery and access system. Technical report, University of Colorado – Boulder, 1995.

[5] Tim Bray. The Open Text Web Index. http://www.opentext.com/.

[6] Infoseek Corporation. Infoseek Net Search. http://www.infoseek.com/.

[7] William Cross. All-In-One Search Page. http://www.albany.net/allinone/, 1995.

[8] C|NET. C|NET's shareware.com. http://www.shareware.com/.

[9] C|net. search.com. http://www.search.com/, 1994.

[10] Daniel Dreilinger. SavvySearch Home Page. http://guaraldi.cs.colostate.edu:2000, 1995.

[11] Daniel Dreilinger. Description and evaluation of a meta-search agent. Master's thesis, Computer Science Dept., Colorado State University, Spring 1996.

[12] Tor Egge, Hugo Eide Gunnarsen, and Stig Sæther Bakken. FTP Search v3.1. http://ftpsearch.unit.no/ftpsearch.

[13] David Eichmann. Ethical web agents. In *Electronic Proceedings of the Second World Wide Web Conference '94: Mosaic and the Web*, 1994. http://www.ncsa.uiuc.edu/SDG/IT94/Proceedings/Agents/ eichmann.ethical/ethics.html.

[14] David Filo and Jerry Yang. Yahoo Home Page. http://www.yahoo.com/, 1994.

[15] Susan Gauch, Guijun Wang, and Mario Gomez. Profusion: Intelligent fusion from multiple, different search engines. *Journal of Universal Computer Science*, 2(9), Sept. 1996.

[16] Paul Gauthier and Eric Brewer. Inktomi web services. http://inktomi.berkeley.edu/.

[17] Luis Gravano, Héctor García-Molina, and Anthony Tomasic. Precision and recall of GlOSS estimators for database discovery. In *Proceedings of the 3rd international Conference on Parallel and Distributed Information Systems (PDIS'94)*, 1994.

[18] Martijn Koster. Configurable Unified Search Engine (CUSI). http://pubweb.nexor.co.uk/public/cusi/doc/about.html, 1994.

[19] Steve Madere. DejaNews research service. http://dejanews3.dejanews.com/, 1995.

[20] Michael Mauldin. Lycos, the catalog of the internet. http://www.lycos.com/, 1994.

[21] Loius Monier and Mike Burrows. Alta Vista Search Engine. http://www.altavista.digital.com/.

[22] Brian Pinkerton. WebCrawler Home Page. http://webcrawler.com/, 1994.

[23] Editor in Chief R.F. Holznagel. Point Web Reviews. http://www.pointcom.com/ .

[24] Gerard Salton. *Automatic Text Processing: the Transformation, Analysis, and Retrieval of Information by Computer*. Addison-Wesley, Reading, Massachusetts, 1989.

[25] Erik Selberg and Oren Etzioni. Multi-service search and comparison using the MetaCrawler. In *Proceedings of the 4th International World Wide Web Conference*, December 1995.

[26] Erik Selberg and Oren Etzioni. The MetaCrawler WWW Search Engine. http://metacrawler.cs.washington.edu:8080/home.html, 1995.

[27] InfiNET Services. InfiNET META search. http://members.gnn.com/infinet/meta.htm.

[28] Mark A. Sheldon, Andrzej Duda, Ron Weiss, and David K. Gifford. Discover: A resource discovery system based on content routing. In *Proceedings of the Third International World Wide Web Conference*. Elsevier, North Holland, April 1995.

[29] Architext Software. excite Netsearch. http://www.excite.com/.

[30] Ian H. Witten, Alistair Moffat, and Timothy C. Bell. *Managing Gigabytes: Compressing and Indexing Documents and Images.* Von Nostrand Reinhold, New York, 1994.

[31] Tak Woon Yan. Stanford Information Filtering Tool (SIFT). http://sift.stanford.edu/.

[32] Tak Woon Yan and Hector Garcia-Molina. SIFT – a tool for wide-area information dissemination. In *Proceedings of the 1995 USENIX Technical Conference*, pages 177–186, Stanford University, 1995.

[33] Shlomo Zilberstein. An anytime computation approach to information gathering. In *Working Notes of the AAAI Spring Symposium Series on Information Gathering from Distributed, Heterogeneous Environments*, Palo Alto, CA, 1995.