SavvySearch: A Meta-Search Engine that Learns which Search Engines to Query

Adele E. Howe
Computer Science Dept.
Colorado State University
Fort Collins, CO 80523
howe@cs.colostate.edu

Daniel Dreilinger
MIT Media Laboratory
Cambridge, MA 02139
daniel@media.mit.edu

January 28, 1997

Abstract

Search engines are among the most successful applications on the Web today. So many search engines have been created that it is difficult for users to know where they are, how to use them and what topics they best address. Meta-search engines reduce the user burden by dispatching queries to multiple search engines in parallel. The *SavvySearch* meta-search engine is designed to efficiently query other search engines by carefully selecting those search engines likely to return useful results and by responding to fluctuating load demands on the Web. SavvySearch learns to identify which search engines are most appropriate for particular queries, reasons about resource demands and represents an iterative parallel search strategy as a simple plan.

1 The Application: Meta-Search on the Web

Companies, institutions and individuals must have a presence on the Web; each are vying for the attention of millions of people. Not too surprisingly then, the most successful applications on the Web to date are search engines: tools that assist users in finding information on specific topics.

A variety of search engines are available, from general, robot based (e.g., AltaVista [Monier and Burrows,], WebCrawler [Pinkerton, 1994]) to topic or area specific (e.g., FTPSearch [Egge *et al.*, 1996], DejaNews [Madere, 1995]). Each employs different algorithms for collecting, indexing and searching links; thus, each returns different results for similar queries. Empirical results indicate that no single search engine is likely to return more than 45% of the relevant results [Selberg and Etzioni, 1995a]. To find what they desire, users may need to query several search engines; meta-search engines automate this process by simultaneously submitting a single query to multiple search engines.

The simplest meta-search engines are forms that allow the user to indicate which search engines should be contacted (e.g., All-In-One [Cross, 1995], META Search [Services, 1996]). ProFusion [of Kansas DesignLab, ,Gauch *et al.*, 1996] gives the user the choice of selecting search engines themselves or letting ProFusion select three of six robot based search engines using handbuilt rules. MetaCrawler [Selberg and Etzioni, 1995a,Selberg and Etzioni, 1995b] significantly enhances the output by downloading and analyzing the links returned by the search engines to prune out unavailable and irrelevant links.

Meta-search engines reduce the burden on the user. They make available search engines that may have been unknown to the user. They handle the simultaneous submission of queries; some direct the query to appropriate engines and some post-process the results as well. They provide a single interface (with the downside that they may not support all the features of the target search engines).

Unfortunately, meta-search can lead to the "tragedy of the commons" problem from economics in which an individual's best interests run counter to society's. Individual users appear to be best served by simultaneously searching every possible search engine on the Web for desired information. Yet, the process may waste Web resources: network load and search engine computation.

We believe that a meta-search system can be a good Web citizen [Eichmann, 1994] by targeting those search engines likely to return useful results and responding to changing load demands on the Web. To provide this functionality, we incorporated simple AI techniques in a meta-search engine. Our meta-search engine learns to identify which search engines are most appropriate for particular queries, reasons about resource demands and represents an iterative parallel search strategy as a simple plan.

2 Our Meta-Search System: SavvySearch

SavvySearch is our meta-search system[Dreilinger, 1996,Dreilinger and Howe, 1996], available at `http://guaraldi.cs.colostate.edu:2000/`. It runs on five machines (three SUN SPARCStations and two IBM RS 6000s) at Colorado State University. The system was first made available in March 1995 and has undergone several revisions since the original design. At present, two versions of the system are available: the one described here and an experimental interface that will be mentioned in Section 3.

SavvySearch is designed to balance two potentially conflicting goals: maximizing the likelihood of returning good links and minimizing computational and Web resource consumption. The key to compromise is knowing which search engines to contact for specific queries at particular times. SavvySearch tracks long term performance of search engines on specific query terms to determine which are appropriate and monitors recent performance of search engines to determine whether it is even worth trying to contact them.

In this section, we describe SavvySearch from a user's perspective. We follow a running example indicating what a user sees and what goes on behind the scenes in processing a search request.

2.1 Submitting a Query

To find out about "artificial intelligence conferences", we enter the query, select the "integrate results" option, and click on the "SavvySearch!" button, as shown in an image of the interface in Figure 1. The search form, the query interface to SavvySearch, asks the user to specify a set of keywords (query terms) and options for the search. Users typically enter two query terms.

The options cover the treatment of the terms, the display of results and the interface language. Query terms may be combined with logical "and" (all query terms must be included in documents), "or" (any query term should be present) or as an ordered phrase. Three aspects of the results display can be varied: the number of links returned, the format of the links description and the timing. By default, 10 links are displayed with the URLs and descriptions when available, and the results of each search engine are listed separately *as they arrive*. Alternatively, we could change the number of links up to 50, return less or more description of the links and interleave the results of the separate search engines. The interface is also available in 23 different languages[1].

2.2 Processing a Query

When a user submits the query, SavvySearch must make two decisions: how many search engines to contact simultaneously and in what order the search engines should be contacted. The first requires reasoning about the available resources and the second about ranking the search engines.

2.2.1 Resource Reasoning

Each search engine queried expends network and local computational resources. Thus, modifying concurrency (number of search engines queried in parallel) is the best way to moderate resource consumption. Concurrency is a function of:

Network Load Estimates which are determined from a lookup table created from observations of the network load at this time of day in the past,

Local CPU Load which is computed using the UNIX `uptime` command.

[1]We thank the users who translated the interface for us.

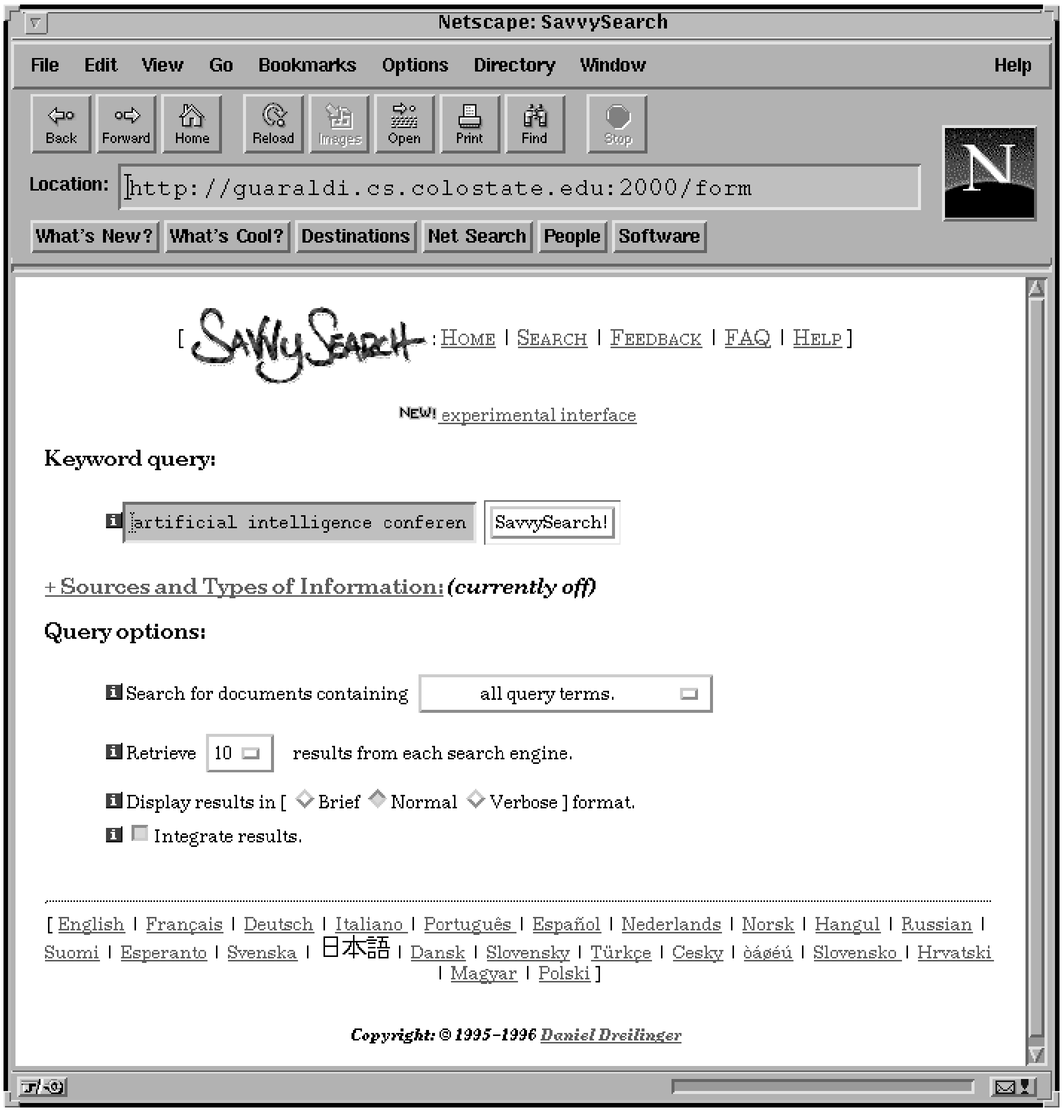

Figure 1: User interface to SavvySearch for entering a query

Concurrency has a base value of two; up to two additional units are added per load estimate for periods of low load. Thus, the maximum concurrency value is six.

2.2.2 Ranking Search Engines

SavvySearch includes both large robot-based search engines and small specialized search engines in its set. The large search engines are likely to return links for any query, but these links may not be as appropriate as links returned by a specialized search engine for a query in its area.

The purpose of ranking is to determine which search engines are most worthwhile to contact for a given query. Search engines are ranked based on:

- learned associations between search engines and query terms (stored in a meta-index) and
- recent data on search engine performance.

The Meta-Index: A Compendium of Search Experience The meta-index maintains associations between individual queries terms (simplified by stemming and case stripping) and search engines as effectiveness values. High positive values indicate excellent performance of a search engine on queries containing a specific term; high negative values indicate extremely poor performance.

The effectiveness values are derived from two types of observations of the results of users' searches. We used observations (passive measures) because we obtained a low rate of response to requests for user feedback, as well as some questionable responses. For each search, we collect two types of information:

No Results: search engine failed to return links,

Visits: number of links explored by the user.

No results reduces confidence that the search engine is appropriate for the particular query; *Visits* indicates that the user found some returned links to be interesting and so increases confidence.

SavvySearch employs a simple weight adjustment scheme for learning effectiveness values. *No results* and *visits* are treated as negative and positive reinforcement, respectively, amortized by the number of terms in the query. Thus, if a search engine returned nothing for the example query, the effectiveness values for "artificial", "intelligence" and "conferences" would each be reduced by $\frac{1}{3}$. Although simple, this scheme proved to be quite effective (see Section 3 for a brief description of our evaluation of the learning).

Tracking Recent Performance Search engines occasionally are inaccessible or slow to respond. Their network connections may be at fault or the engines themselves may be experiencing problems or upgrades. SavvySearch monitors recent performance by recording the number of links returned (hits) and the response time for the last five queries submitted to each search engine. Given current usage of the system, five queries corresponds to about 45 seconds for the large, general search engines and about fifteen minutes for the infrequently used search engines.

Calculating Rank from Experiences For a given query (q), a search engine's (s) rank ($R_{q,s}$) is its query score reduced by a penalty for recent poor performance on hits (h) and response time (r):

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

Each term is normalized to between 0 and 1.

The equation for $Q_{q,s}$ is based on a common approach from information retrieval called *Term Frequency times Inverse Document Frequency*[Witten *et al.*, 1994]. The query score, $Q_{q,s}$, sums the meta-index values for the terms in the query weighted by ubiquity of the query term and the search engine:

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

$M_{t,s}$ is the weight from the meta-index of the term t for search engine s. I_t is the inverse server frequency of the term, which estimates the ubiquity of the query term; frequently occurring, common terms provide little information for distinguishing search engine performance and so are discounted. T_s sums the absolute values of all meta-index values for search engine s; this term estimates the frequency of overall use of the search engine. Because the most general search engines are likely to be used more frequently and are likely to return something for any query, their weights will tend to grow larger and more quickly than the specialized search engines. T_s mitigates the tendency towards their dominance, allowing more specialized engines to be selected when appropriate.

The penalties ($P_{s,h}$ and $P_{s,r}$) are accrued only if thresholds on minimum number of hits and maximum response time are exceeded. The thresholds have been set somewhat arbitrarily to average of 1 hit and response time of 15 seconds recently. Once the thresholds are passed, the penalties increase quadratically up to the worst possible values: zero for hits and time out of 45 seconds for response time. Details of these equations are available in [Dreilinger, 1996].

2.3 Dispatching a Query

The degree of parallelism determines how many search engines to query; the rank order determines which should be queried. SavvySearch dispatches the query in parallel to each of the indicated search engines. For each search engine, a specialized interface agent formats the query according to the interface for the search engine and submits it. The interface agent waits a preset amount of time for a response, handles errors that might occur, parses the result into a uniform format and forwards it to another component for display.

2.4 Presenting Results

For our example query, the three search engines contacted (WebCrawler, Lycos and YellowPages) returned 30 different links with similar names. The first nine are shown in an image of the result page in Figure 2. These results were integrated; SavvySearch waited until all results had been received and then constructed a single ranked list. Results are integrated 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. Duplicate links are listed with the names of all the search engines that returned them.

The bottom of each results page displays the search plan, as shown in Figure 3. The query was issued during a time of moderately high demand; thus, each step includes only three search engines. Search plans can include between two and six search engines per step. The present version includes 11 search engines; the number fluctuates as search engines appear, disappear and merge.

To supplement the results collected so far, the user can execute another step in the search plan by clicking on the one that looks the most promising. The ordering represents SavvySearch's best guess about which search engines will return the best results; the user can easily override it by selecting a different step. In one long term study, we found that users executed 1.42 steps in their search plans on average.

3 Retrospective and Prospective Views of the SavvySearch Project

The SavvySearch project has been quite successful. Currently, SavvySearch processes over 20,000 queries each day and, based on electronic mail, has attracted a large well-satisfied user base.

SavvySearch, as described here, is the result of almost two years of work and a series of studies [Dreilinger and Howe, to appear]. The present design of the resource reasoning and learning algorithm resulted, in part, from the results of these studies [Dreilinger, 1996]. We studied the effects of the learning by starting from a minimal meta-index (compiled from 2 days worth of data) and allowing it to accumulate experience over a 28 day period. We found that our simple learning algorithm improved performance on both *visits* and *no results* overall (including all search engines and query terms): *visits*

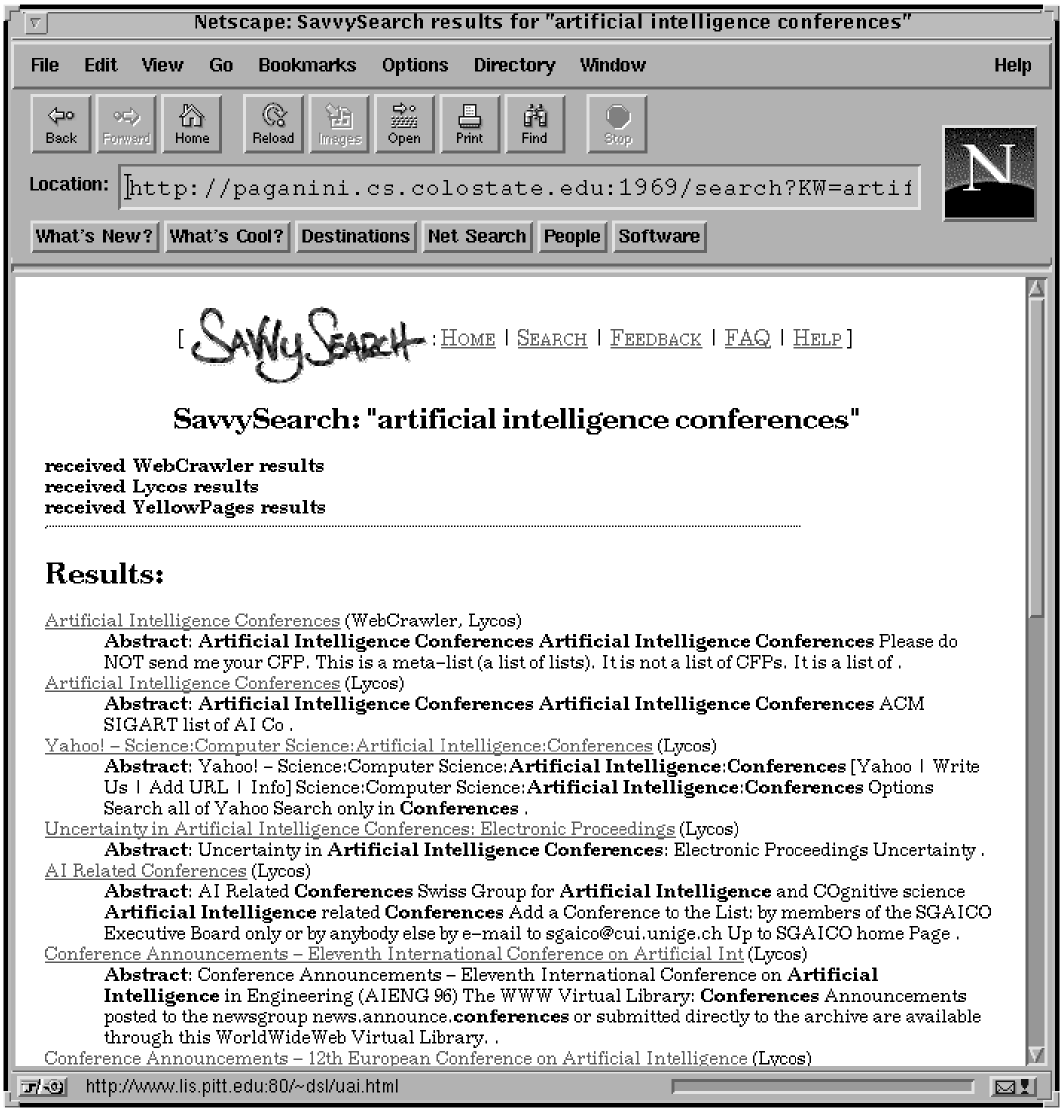

Figure 2: SavvySearch display of results of *artificial intelligence conferences* query, interleaved display

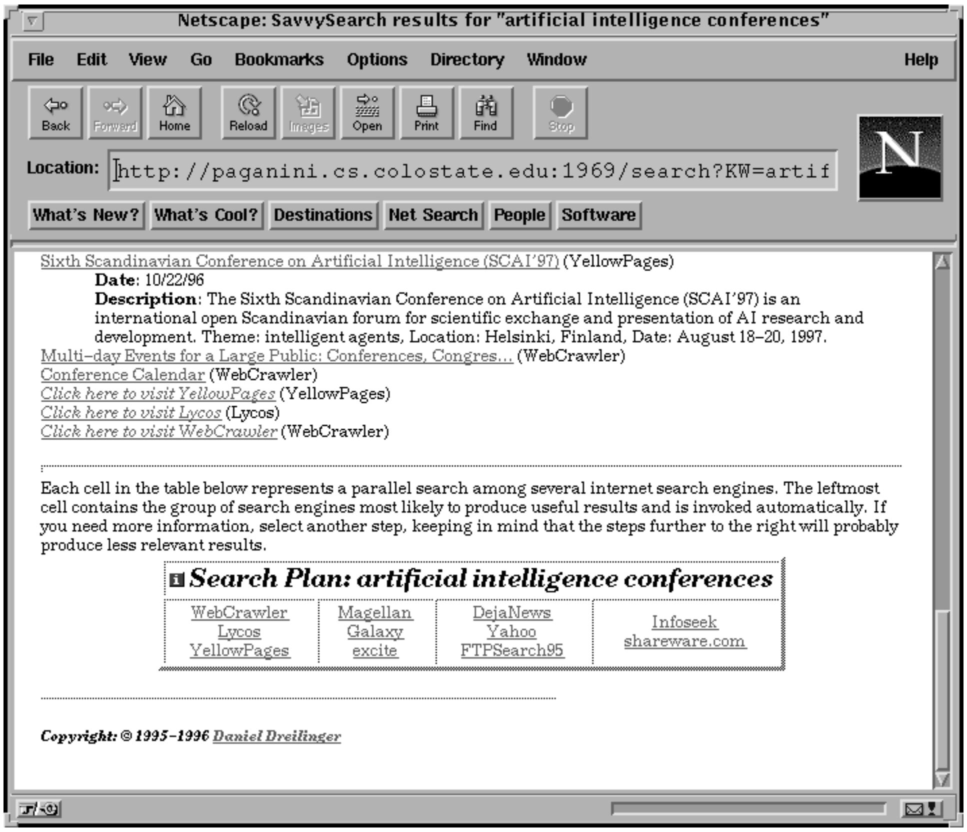

Figure 3: Search plan for *artificial intelligence conferences* query, which allows users to continue search if current results are inadequate

averaged .36 in the first five days and .42 in the last five days; *no results* averaged .142 in the first five days and .135 in the last. Thus, this learning algorithm leads to better selection of good search engines than pruning of poor.

In a followup study, we examined how much knowledge was needed to significantly improve performance using learning. We found that as few as 10 usages of a query term results in a halving of *no results* from .18 to .09. While *visits* increases more slowly with learning, an increase from .34 to .45 is obtained with just 100 examples. By the end of the study, the most used term (5000 usages) had *visits* of .76 and *no results* of .08. Given the level of usage of most search engines (millions of queries per day), meta-search engines should have little difficulty in collecting enough data to significantly improve performance through learning.

As Web usage and users becomes more sophisticated, meta-search will need to be able to be personalized and embedded in other systems. Meta-search currently takes a "one-size-fits-all" approach in which the knowledge underlying query processing is shared by all users. A new experimental version of SavvySearch (available via the main Web page) takes a first step toward personalization by dividing searches into eight categories; each category translates to a set of rules for creating a search plan for that type of search. Ideally, users themselves could create and store their own stereotypical search plans or a system could infer them.

Just as primary search engines are now a resource for meta-search engines, so meta-search engines should become a means of achieving higher level goals, rather than an end in itself. Intelligent agent technology promises to alleviate the tedium and frustration of mundane tasks and navigate vast information spaces. Meta-search should be a information gathering tool for helping human users and their intelligent agents find what they need on the Web.

The Web will continue to grow. Thus, search tools will continue to be critical for managing the information deluge. To keep pace with the expansion, the next generation must include far more sophisticated AI techniques than the current, while retaining some of the benefits of the current systems: be easy to use, require little feedback from the users and be mindful of shared resources.

4 Acknowledgments

The experiments and equipment were supported in part by ARPA-AFOSR contract F30602-93-C-0100, NSF Research Initiation Award #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

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

[Dreilinger and Howe, 1996] Daniel Dreilinger and Adele Howe. An information gathering agent for querying web search engines. Technical report, Colorado State University, 1996. TR 96-111.

[Dreilinger and Howe, to appear] Daniel Dreilinger and Adele E. Howe. Experiences with selecting search engines using meta-search. *ACM Transactions on Information Systems*, to appear.

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

[Egge *et al.*, 1996] Tor Egge, Hugo Eide Gunnarsen, and Stig Sæther Bakken. FTP Search v3.1. http://ftpsearch.unit.no/ftpsearch, 1996.

[Eichmann, 1994] 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.

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

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

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

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

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

[Selberg and Etzioni, 1995a] 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.

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

[Services, 1996] InfiNET Services. InfiNET META search. http://members.gnn.com/infinet/meta.htm, 1996.

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