# 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.

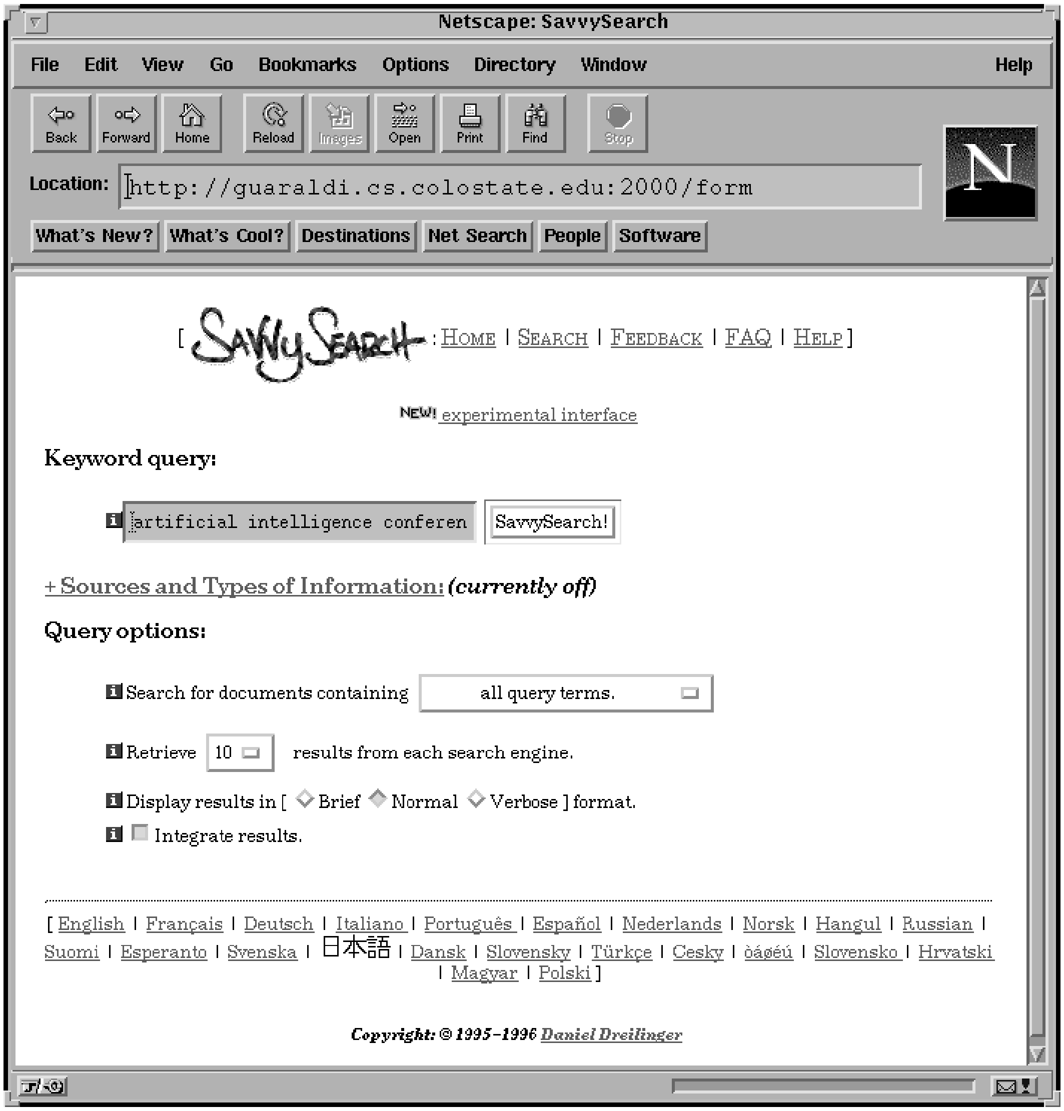
### 2.2.1 Resource Reasoning

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.



Location: http://guaraldi.cs.colostate.edu:2000/form

[ SavvySearch : Home | Search | Feedback | FAQ | Help ]

NEW! experimental interface

Keyword query:

artificial intelligence conferen [SavvySearch!]

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

Query options:

Search for documents containing [all query terms.]

Retrieve [10] results from each search engine.

Display results in [ ◇ Brief ◆ Normal ◇ Verbose ] format.

□ Integrate results.

[ English | Français | Deutsch | Italiano | Português | Español | Nederlands | Norsk | Hangul | Russian | Suomi | Esperanto | Svenska | 日本語 | Dansk | Slovensky | Türkçe | Cesky | òáøéú | Slovensko | Hrvatski | Magyar | Polski ]

***Copyright: © 1995–1996 Daniel Dreilinger***

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).

DOCKET ALARM

# Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

## Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

## Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

## Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

## API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

### LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

### FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

### E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.

WHAT WILL YOU BUILD? | sales@docketalarm.com | 1-866-77-FASTCASE

fastcase®
Smarter legal research.