Deep Sea Diving or 'What to do when you are tired of surfing?'

Judi R. Thomson
Supervisors: J.E. Cooke and J.E. Greer

Abstract

The World Wide Web is one of the fastest growing sources of information in the world. In 1994 the Web's size increased by 2% per month. However, an information service is useful only if the information provided can be retrieved when needed. The World Wide Web is not adequate in this respect. Finding specific information on the Web is difficult. The additional information providers that appear daily only add to the difficulty of information retrieval.

This paper will survey some of the information retrieval methods available on the World Wide Web. The attributes of these methods will be examined through investigation of specific search engines. A classification of the characteristics of each example retrieval method will be developed. Characteristics desirable for precise information retrieval will be discussed and compared to the characteristics of the example retrieval engines. This comparison will provide insight into designing an improved World Wide Web search engine and demonstrate the need for metrics for WWW resource location.

1 Introduction

Searching through ordered information with a computer is rapid and effective. Unfortunately, information on the World Wide Web is not organized to facilitate simple information retrieval. A browsing user is challenged to examine even a few of the estimated 140 000 World Wide Web pages when looking for specific information [1].

Most information seekers make use of search engines or meta-libraries that are resident on specific servers. Such tools are useful but often limited in scope. They sometimes provide information that is too coarse or are inaccessible because of network traffic. The expanding information needs of World Wide Web users require improved search facilities. As the Web continues to grow, efficient, precise information retrieval will be one of the predominant issues facing information providers and information seekers.

This paper will examine several methods of searching for information on the World Wide Web. The methods will be classified and their individual characteristics examined. Through this examination, insights into improving information retrieval are gained.

2 Information Retrieval from the World Wide Web

The World Wide Web can be modelled as a distributed, textual database utilizing client and server machines, allowing application of traditional information retrieval terminology and methods to the problem of searching the Web. World Wide Web documents are stored as text with embedded formatting commands, known as HyperText Markup Language (HTML). The formatting commands add no meaning to the document and are ignored when retrieving documents.

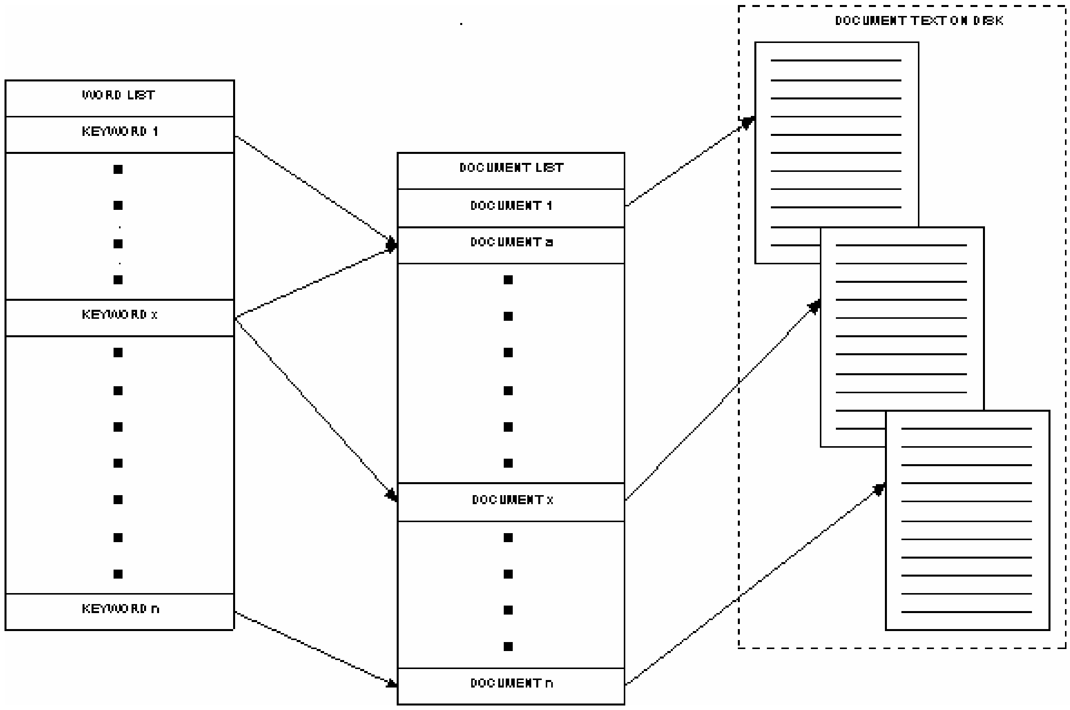

Figure 1: Pictorial representation of an Inverted Index

Retrieving information from a textual database is more complicated than retrieving information from a relational database because text is unordered and irregular. Searching through unordered documents for particular information is slow and inefficient. There are many different methods for imposing order on a collection of documents such as inverted indices, vector space models, clustering and probabilistic retrieval [20, 9]. Salton gives good descriptions of most methods [21, pp 229-266, 312-365].

World Wide Web retrieval engines generally use an inverted index. One representation of an inverted index is shown in Figure 1. The index is created using manually or automatically assigned keywords. Key words for Web documents are sometimes taken from the title and text links. More often every meaningful word in the text is used as a key word for the document. Once each document has a list of key words the list is inverted so that each term is associated with a number of documents. Querying the database results in retrieval of the documents linked to the words in the query.

2.1 Resource Discovery

The process of finding information from the Web is often called resource discovery. There are several different approaches to resource discovery on the Web. The end result for each approach is a list of one or more Uniform Resource Locators (URL) that identify documents that may be relevant. A URL is a unique identifier given to every Web document and is used as an address for the document. For example, the URL for the Computer Science Department at the University of Saskatchewan is http://alf.usask.ca. Browsers use the URL to find the location of the WWW page and retrieve a copy. A browser is a program that allows navigation through the repository of documents using embedded hyperlinks. To find information a user must spend many hours randomly looking at documents to determine their usefulness. Browsing is very slow and often the user finds no relevant documents. The following sections describe some common aids to resource discovery as well as some recent approaches to the problem.

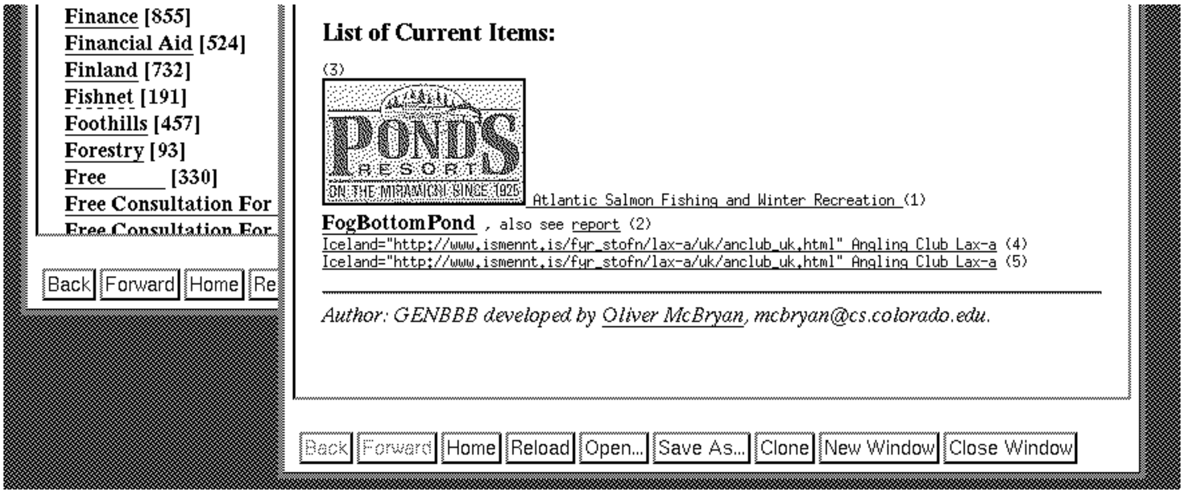

Figure 2: Partial GENVL hierarchy- user has selected *fishnet*

2.2 WAIS

Wide Area Information Servers (WAIS) have been used for information retrieval from WWW pages as well as from other internet services for some time. Information provider sites use WAIS software to create a full text index for their collection of documents. The software has no ability to share information between servers so the index created only contains information about documents referenced by the provider site. Users access WAIS indexes through a listing of available servers. For users who know which index contains references to relevant information, WAIS searches are useful. The listing of WAIS servers can be searched with keywords to give a shortened list of potentially useful servers [17].

2.3 Virtual Libraries

An alternate method of resource discovery is to use a hierarchical index or virtual library. These indices attempt to categorize the material available on the Web with increasing levels of detail. To retrieve information from a virtual library, the user traverses the library hierarchy until the desired information is found. Virtual libraries collect information about the Web in a variety of ways. Some libraries are entirely maintained and updated by individuals, some are maintained by Web users and some rely on a combination of user input and owner maintenance. The Whole Internet Catalog and the YAHOO reference library are two examples of virtual libraries [10, 4]. The remainder of this section will focus on a third library, GENVL.

GENVL (GEnerate Virtual Library) is a virtual library that is updated entirely by internet users. An example of part of the GENVL hierarchy is shown in 2. The user has selected the category Fishnet to reveal the subcategories associated with it. The initial categories of GENVL were provided by the designers along with mechanisms for users to add references to the library. A user may add a reference, which may be a single WWW page or another virtual library, to the hierarchy very easily. Password protection is provided to prevent others from removing the reference. GENVL does not impose a hierarchy on the users, rather the hierarchy evolves using the categories supplied by users who post references. The GENVL hierarchy is periodically compiled into an index file allowing keyword searches to be performed to assist navigation [14].

2.4 Robot Based Indexing

The search for information from the World Wide Web can be automated through the use of robots [13]. Robots are programs that traverse the Web automatically. Many robots navigate the Web by starting from a given document and following the hyperlinks from that document to new documents. The search can proceed breadth first or depth first. Indexing robots retrieve Web pages one at a time and store information about the pages in an index file. The index file becomes the inverted index used by the retrieval engine. Information is usually retrieved through a WWW document that allows the user to submit keywords or regular expressions as search keys. The index file is searched using the submitted keys and a list of document URLs is returned to the user. The key difference between WAIS indexing and most robot generated indices is the latter attempt to create one index for the entire World Wide Web. Several popular search engines make use of a robot created index. Five examples are described below.

2.4.1 World Wide Web Worm

The World Wide Web Worm creates an index using a limited-depth first search strategy. Each document is indexed by its title and by any text associated with its URL in referencing documents. Searches may be performed on either document titles, specific URLs or on referencing hypertext [15, 16].

2.4.2 WebCrawler

WebCrawler searches the Web breadth first, choosing the documents from as many different locations as possible. A graph representation of the Web is stored along with the full text index file to facilitate choosing new servers. The WebCrawler attempts to identify important words in a document by assigning weights to index terms based on term frequency. Any documents retrieved by WebCrawler that are not indexable (ie. graphics) are ignored. The index built by WebCrawler can be searched with keywords to find documents. The index can also be used as a starting point for running the WebCrawler in real time. In real time mode WebCrawler uses the hyperlinks in documents to determine the likelihood that following that link will satisfy the user query. A real time search continues until enough documents have been found that satisfy the query. The real time mode of WebCrawler is not available to the public [19].

2.4.3 RBSE spider

The RBSE (Repository Based Software Engineering) spider also indexes the full text of World Wide Web documents. The RBSE spider was designed find WWW resources related to software engineering. These resources are included in a repository of software engineering documents. The creators plan to include domain information with the spider to enable it to create other domain-specific indices. The RBSE spider traverses the web either breadth first or depth first retrieving only HTML documents. Retrieval from this index is supported by a slightly modified WAIS search [7, 8].

2.4.4 Harvest

Harvest [2, 22] is an resource discovery system built to reduce the number of documents retrieved and to improve robotic information sharing. Harvest has a decentralized approach to information gathering. Gatherer is a program that is run at information provider sites to create local indices. The index is stored on the server and is later collected by a Broker. A Broker combines information from many Gatherers as well as from other Brokers to create one index file. Harvest defaults to document by document retrieval if the server being examined does not have a Gatherer to pre-index documents. The combined information can be searched using a variety of search engines including WAIS and Glimpse. Glimpse is an indexing search engine that creates smaller indices than WAIS.

2.4.5 Fish Search

The fish-search is significantly different from previously mentioned search engines in that it is a client based search engine. Instead of using an index of previously classified documents, the fish search performs searches in real time from the client machine. The fish search must be added to the NSCA Mosaic browser to be used.

To find information the fish search is given the URL of a document to initiate the search from. That document is retrieved, scanned for relevant information and for links to other documents. Then a number of additional robots are spawned and the process repeats from the new documents. The number of new robots created depends upon the relevance of the document retrieved and upon the number of URLs referenced by the retrieved document. If a search gets to a predefined depth without finding a relevant document then that branch of the search is terminated.

The fish search allows key word searches, searches by regular expression as well as providing for the use of an external program to calculate a relevance rating for the document [6].

3 Requirements of Resource Locators

The variety of approaches to WWW resource discovery combined with the lack of metrics makes comparing methods somewhat difficult. This section will look at attributes shared by all Web resource locators and discuss which attribute values could contribute to a better search engine [23]. Table 1 provides a summary.

Table 1: Characteristics of Resource Locators

	Information Gathering	Granularity	Distribution	Query Format	Updates
WAIS	collection must be local	full text	distributed	natural language, key word	incremental
GENVL	submitted by users	user assigned categories	central	natural language, key word	at user discretion
WebCrawler	follow links to documents	full text	central	key word, regular expression	incremental
WWW Worm	follow links to documents	titles, hyperlinks	central	key word, regular expression	incremental
RBSE Spider	follow links to documents	full text	central	natural language, key word	incremental
Harvest	get indices from Gatherers	Gatherer dependent, full text capable	central and distributed	dependent on retrieval engine	incremental
Fish Search	follow links to documents	full text	none	key word, regular expression, external filter	none

3.1 Information Gathering

The method used to gather information determines the resource locators impact upon World Wide Web resources. Programs that retrieve individual Web pages must access a Web server once for every page supported. Since most Web robots work without sharing information a server could be contacted by many different robots. Multiple accesses by robots can be a heavy load on the information server as well as a major consumer of internet bandwidth.

Robot collection of server specific index files greatly reduces server load and network traffic but makes the search engine dependent upon the quality of indexing performed at the local site. If a local indexing process is incompatible with a particular search engine, the index created is unusable and a portion of the Web becomes inaccessible. Significant parts of the Web can be missed by robots since most robots find new pages by following links from previously found pages. Any WWW site that is not referenced by hypertext links in documents will not be found by following hyperlinks. Also an increasing number of Web pages are referenced by hypertext links embedded in graphical images. These images are not indexed by most robots thus the pages referenced by them are not included in the indices. Local indexing simplifies the task of ensuring that all documents are included in the index whether a reference to that document exists or not. A corresponding problem is the inclusion of multimedia objects in the indexing process. Most indexers include textual information only. The Harvest system allows the user to define methods to handle non-textual objects.

3.2 Granularity

Each retrieval engine must classify documents according to some criteria. [1] When the classification is very detailed, more precise information can be retrieved. Granularity refers to the level of detail the indexing process has. An index that has a very fine granularity will be much larger than one that is quite coarse.

Virtual libraries classify material only by subject. For some libraries the subjects are chosen by the maintainer, but for GENVL the subjects are created by the users. Precise retrieval from this sort of system is only possible when each user has the same understanding of what the subject headings mean. Information retrieval can be improved by cross referencing material to more than one subject.

Indexing on a portion of the text in a document gives a more detailed representation of the information than subject indices. Careful selection of the text to be indexed improves the representation in the index. The World Wide Web Worm indexes on only a portion of a given document.

Full text indexing gives a clear indication of the content of the document. WebCrawler, WAIS, RBSE spider, and Glimpse all index the complete text of a document. When word phrases are included as part of the index the granularity increases even more. Most designers agree that full text indexing is necessary to improve information retrieval performance [7, 19, 2].

An even finer granularity can be realized by adding extra words. Significant words from the document can be expanded by adding synonymous terms to the index. The index then represents the document in more detail than is actually present in the document. The tasks of choosing the significant words and the synonyms for those words are non-trivial. No World Wide Web search engines have been found that create such a fine index. Interested readers are referred to the INQUERY project as an example of an information retrieval system that uses term expansion techniques [3].

3.3 Distribution

Index files can be stored at one location or distributed among several locations. Most Web search engines maintain a central index file that is administered by one server. Occasionally that index file might be copied to an additional site to reduce server load.

A central index file ensures that the search engine is not dependent on the functionality of other Web sites to complete a search. The index file is generated by the same program that will use it, guaranteeing compatibility. Searches can be completed very quickly since all the data is immediately

[1] The terms Granularity and Distribution have been borrowed from Schwartz [23]

at hand. Reliance on a central index file requires that the server providing the search facility is operational.

Distribution of indices reduces the load on any particular server and provides at least partial access to the search engine even if a server is down. A distributed data base doesn't require large amounts of storage space at one location. If information is clustered so that each index covers a particular domain, distributed index files might even speed up search times [7]. One difficulty with distribution of indices is knowing which index to search. The indexes provided by WAIS servers are an example of distributed data bases. The Harvest system makes use of both distributed and centralized index files.

3.4 Interface

The user interface is extremely important to the functionality of a search engine. The interface is the tool the users employs to communicate with the resource locator. A cumbersome interface lessens the utility of the program. The resource location tools available for the World Wide Web a combination of hierarchies, keywords, regular expressions and natural language for user communication.

Virtual libraries are hierarchical. One simply chooses a category from a list which reveals a list of sub-categories. The process is repeated for subcategories until the desired information is found. The user initially needs only a general concept to begin to find information.

Keyword searches are accepted by most resource discovery tools. A word or combination of words is submitted to the search engine which looks for documents containing the words. Searching by keyword requires the user to have a precise notion of what information is desired. Most users compose short, general queries making precise information retrieval difficult. The average length of queries submitted to the WebCrawler is only 1.5 words [19]. Keyword searching can be combined with hierarchical lists to allow users to narrow the search space via a hierarchy before performing a keyword search. Glimpse http allows this combination [12]. An increasing number of tools accept natural language as input and generate a keyword query internally. Users may find it easier to formulate information needs in natural language but creating accurate keyword queries from natural language is very difficult. Natural language processing is an area of ongoing research. WAIS search engines can accept natural language input.

Regular expressions are a specialized form of keyword searching. Words can be combined with special characters that have meaning to the program to produce very powerful queries. For instance, computer\s?(science|programs) is a Perl regular expression that initiates a search for either computer science or computer programs. Users of regular expressions must know precisely what their information needs are, and must know how to construct a valid regular expression [11]. Queries can be improved by adding more information. Query terms can be expanded using concept maps and thesauri to produce a more detailed query. Query expansion is a planned extension to the WebCrawler. The Fish Search's capability to include an external program as query processor gives it the flexibility to include query expansion as well.

3.5 Updates

Any database must be added to and updated periodically. The task of updating data about the World Wide Web is complicated by the rapid growth of the Web. All of the search engines discussed add new data to index files and update old information incrementally. It is not feasible to attempt to index the entire Web each time new information is added. Most of the programs discussed update their information in 1 to 2 week cycles [1].

4 Improving Resource Location

Users of information from the World Wide Web have different needs than providers of information. Users need information to be specific, quickly available, updated regularly and easy to find. The granularity, distribution, interface and update method of a tool are important attributes from the user viewpoint. Information providers require indices to be as small as possible to reduce disk space requirements, minimized server load and ensured security. Providers will place more importance on the attributes of information gathering, and distribution. Improved resource location tools must account for these differing requirements. An improved tool should be a good Web citizen and minimize the use of Web resources and bandwidth. The data used for information retrieval must be easy to update and should be at least partially distributed to allow access to more users and to avoid server load. Documents should be indexed on full text and the interface should be intuitive and easy to use. The resource locator should be able to include graphics and other multimedia in its database.

5 Metrics for the Web

Once a new resource location tool is developed how does one decide if it is an improvement? There are few metrics available to assess the functionality of resource locators except for comparisons between the number of documents retrieved for a specific query, or the number of accesses to a tool in a given time frame. In order to make more meaningful comparisons information is needed about how effectively the tool finds required documents, how long the tool takes to process a request for information, and how easy the tool is to use. The metrics must be applicable to all forms of resource location. Measures of this sort would provide information for designers trying to improve WWW resource location. These measures could be combined to create an overall rating of each tool. The following sections propose a preliminary set of metrics for WWW resource location. Some of the necessary information can be collected from inspections of existing data on computer systems. Much of the information would need to be collected during test sessions for individual resource location tools. Measures of recall, precision, interface useability, availability and wait time are most useful for WWW users attempting to improve information retrieval. Measures of disk space used, bandwidth requirements and processor time are more relevant to information providers.

5.1 Retrieval effectiveness

Two metrics used for measuring the effectiveness of an information retrieval system are recall and precision. Recall is the ratio of the number of items retrieved to the total number of relevant items in the database. An absolute measure of recall requires that the entire database is manually searched for items relevant to a particular set of queries. Clearly, measuring absolute recall over anything but a very small portion of the Web is not feasible. It is possible to arrive at a relative measurement of recall by comparing the number of documents retrieved by different resource locators. In the case of virtual libraries this would be a measure of the number of documents at the leaf level of the hierarchy. The recall of a particular search method is measured to find out if enough documents are being retrieved. When retrieving information from the Web the problem is usually that too many documents are retrieved. To alleviate the problem of too much information the precision of the search method must be improved.

Precision is the ratio of the number of relevant items retrieved to the total number of retrieved items. Precision is a feasible metric to apply to information retrieval from the World Wide Web because it doesn't require an exhaustive classification of all documents. To measure precision, the relevance of each retrieved document must be judged. Usually this is done by human judges who compare each retrieved document against a set of criteria and classify it as relevant or not relevant. Each retrieval method is tested using the same query. Because of the volatile nature of WWW data this approach provides only a time- specific snapshot of a tool's retrieval precision. A different approach is to store information about each use of a resource locator in a log file. Each entry in the log would consist of a user assessment of the number of relevant documents and the number of documents retrieved for the query. The two sets of information could be combined into a measure of precision. Unfortunately, having uniform criteria to judge the relevance of a document is impossible using this method. Together, measurements of recall and precision give a reasonable picture of how well a resource location tool performs.

5.2 Use of Resources

An evolving issue for all Web citizens is the consumption of resources. The resources in question are: disk space, network bandwidth, computer processor time, and time spent waiting. A good search engine should be conservative with resources.

Some of these resources are easy to find metrics for. It is straightforward to find out how much disk space is required for any particular resource locator. Calculating the computer processor time a program uses is relatively easy even when multiple incarnations of the same program are working in parallel. It is also possible to monitor the amount of network time a particular program uses by logging all the conversations between that program and host computers. The information in the log allows calculation of how much time a particular page takes to transfer from server to client. The sum of processor time and page loading time gives the time a human must spend waiting while using the resource locator.

5.3 Usefulness

The usefulness of a resource locator is a combination of its useability and availability. The useability of a tool is directly dependent on the quality of its human-computer interface. Measuring useability is a complicated process because there are many variables that affect the useability of a program [5]. A useability metric for WWW resource locators should include some definition of who the tool is useable for, measures of performance and allow comparison of one tool to another. These attributes could be measured successfully using an experiment in which equal numbers of users are given tasks to perform with each resource locator. The time taken to successfully perform the tasks is monitored, as well as the number of user retries needed before successfully finding useful documents. An exit survey is used to discover subjective opinions about the resource location tools. The participants in such an experiment should be categorized to ensure that like numbers of expert and novice users are represented for each resource locator. The data collected from this experiment, combined with data collected about wait time and precision would give a good indication of the useability of a tool.

Another issue affecting the usefulness of a resource locator is its availability. A tool that is extremely difficult to access is less useful than one that is always available. The accessibility of a tool is governed by how often the server for it is down, and by how much network traffic is allowed on that server. Paradoxically, availability is also affected by the popularity of a tool. A popular resource locator is harder to access but is either more useful than other tools, or is more widely known. To remain useful, a popular tool may need to be placed on a more accessible server. A log of attempted uses compared to a log of actual utilizations of a resource locator might allow one to draw some conclusions about availability.

6 Conclusions and Future Work

All WWW resource locators share certain attributes and can be classified accordingly. Users of the World Wide Web have different requirements of resource location tools than information providers. Improvements to current tools must consider these differences. Since few methods exist for comparing resource locators, it is very difficult to determine which approaches are superior. The development of metrics that could be applied to WWW information retrieval would make such comparisons more valid. Future work will include investigation of techniques to measure and improve the precision of information retrieval from the World Wide Web.

References

[1] Aviani, J. NIKOS: An Internet Resource Locator A preliminary Report. Located at http://www.rns.com/cgi-bin/nikos, accessed March 1995.

[2] Bowman, C. M., Danzig, P. B., Hardy, D. R., Manber, U., and Schwartz, M. R. Harvest: A Scalable, Customizable Discovery and Access System. Technical Report, Department of Computer Science, University of Colorado, 1994.

[3] Callan, J. P., Croft, W. B., and Harding, S. M. The INQUERY retrieval system. In *Proceedings of the Thirds International Conference on Database and Expert Systems Applications*, (Valencia, Spain, 1992).

[4] Chambers, A. editor The Whole Internet Catalog.
Located at http://nearnet.gnn.corn/gnn/wic/index.html, accessed March, 1995.

[5] Chapanis, A. Evaluating Usability. In *Human Factors for Informatics Usability*, B. Shackel, and S. Richardson, Eds. Cambridge Universty Press, Cambridge, 1991, pp. 359-395.

[6] Debra, P., and Post, R. Information Retrieval in the World-WideWeb: Making Client-based searching feasible. In *Proceedings of the first International World Wide Web Conference*, (Geneva, 1994).

[7] Eichmann, D. The RBSE Spider- Balancing Effective Search against Web Load. In *Proceedings of the First International World-Wide Web Conference*, (Geneva, 1994).

[8] Eichmann, D. RBSE spider query page.
Located at http://rbse.jsc.nasa.gov/eichmann/urlsearch.html, accessed March 1995.

[9] Faloutsos, C. Access methods for text. *ACM Computing Surveys 17*, (1985), 50-74.

[10] Filo, D., and Yang, J. Yahoo. Located at http://akebono.stanford.edu/yahoo/, accessed March 1995.

[11] Hill, J. Help on using Perl regular expressions when searching.
Located at http://www.lpac.qmw.ac.uk/SEL-HPC/Articles/PerlRegexp.html, accessed March 1995.

[12] Klark, P. Glimpse HTTP. Located at http://glimpse.cs.arizona.edu:1994/glimpsehttp.html, accessed March 1995.

[13] Koster, M. Robots, Spiders and Wanderers.
Located at http://web.nexor.co.uk/mak/doc/robots/robots.html, accessed March 1995.

[14] McBryan, O. GENVL.
Located at http://www.cs.colorado.edu/homes/mcbryan/public_html/bb/summary.html, accessed March 1995.

[15] McBryan, O. GENVL and WWWW: Tools for Taming the Web. In *Proceedings of the First International World-Wide Web Conference*, (Geneva,1994).

[16] McBryan, O. World Wide Web Worm Queries.
Located at http://www.cs.colorado.edu/home/mcbryan/WWWW.html, accessed March 1995.

[17] Oliveau, L. WAIS for UNIX, Technical Description.
Located at http://wais.com/company/Tech_TOC.html, accessed March 1995.

[18] Pinkerton, B. Finding What People Want: Experiences with the WebCrawler. In *Proceedings of the First International World-Wide Web Conference*, (Geneva, 1994).

[19] Pinkerton, B. The WebCrawler Index.
Located at http://webcrawler/cs.washington.edu/WebCrawler/WebQuery.html, accessed March 1995.

[20] Salton, G. Another Look at Automatic Text-Retrieval Systems. *Communications of the ACM 29*, 7 (1986), 648-656.

[21] Salton, G. *Automatic Text Processing- The Transformation, Analysis, and Retrieval of Information by Computer*. Addison-Wesley, New York, 1989.

[22] Schwartz, M. F. Harvest Home Page. Located at http:// harvest.cs.colorado.edu, accessed March 1995.

[23] Schwartz, M. F., Emtage, A., Kahle, B., and Neuman, B. C. AComparison of Internet Resource Discovery Approaches. *Computing Systems 5*, 4 (1992) 461-496.