CONTEXT AND PAGE ANALYSIS FOR IMPROVED WEB SEARCH

NEC Research Institute has developed a metasearch engine that improves the efficiency of Web searches by downloading and analyzing each document and then displaying results that show the query terms in context.

STEVE LAWRENCE AND C. LEE GILES
NEC Research Institute

Several popular and useful search engines—such as AltaVista, Excite, HotBot, Infoseek, Lycos, and Northern Light—attempt to maintain full-text indexes of the World Wide Web. However, relying on a single standard search engine has limitations. The standard search engines have limited coverage,[1,2] outdated databases, and are sometimes unavailable due to problems with the network or the engine itself. The precision of standard engine results can also vary because they generally focus on handling queries quickly and use relatively simple ranking schemes.[3] Rankings can be further muddled by keyword "spamming" to increase a page's rank order. Often, the relevance of a particular page is obvious only after loading it and finding the query terms.

Metasearch engines, such as MetaCrawler and SavvySearch, attempt to contend with the problem of limited coverage by submitting queries to several standard search engines at once.[4,5] The primary advantages of metasearch engines are that they combine the results of several search engines and present a consistent user interface.[5] However, most metasearch engines rely on the documents and summaries returned by standard search engines and so inherit their limited precision and vulnerability to keyword spamming.

We developed the NEC Research Institute (NECI) metasearch engine to improve the efficiency and precision of Web search by downloading and analyzing each document and then displaying results that show the query terms in context. This helps users more readily determine if the document is relevant without having to download each page. This technique is simple, yet it can be very effective, particularly when dealing with the Web's large, diverse, and poorly organized database. Results from the NECI engine are returned progressively after each page is downloaded and analyzed, rather than after all pages are downloaded. Pages are downloaded in parallel and

 1089-7801/98/$10.00 © 1998 IEEE

the first result is typically displayed in less time than a standard search engine takes to display its response.

The NECI metasearch engine is currently in use by employees of the NEC Research Institute. This article describes its features, implementation, and performance.

A recent study by Anastasios Tombros verified the advantages of summaries incorporating query term context.[6] His study found that users working with query-sensitive summaries found relevant documents faster and performed relevance judgments more accurately and rapidly than users working with an abstract or query-insensitive document summary. Query-sensitive summaries also greatly reduced the need for users to access the full text of documents.

THE NECI METASEARCH ENGINE

Figure 1 shows a simplified control flow diagram of the NECI metasearch engine, which consists of two main parts: the metasearch code and a parallel page retrieval daemon. The page retrieval engine is

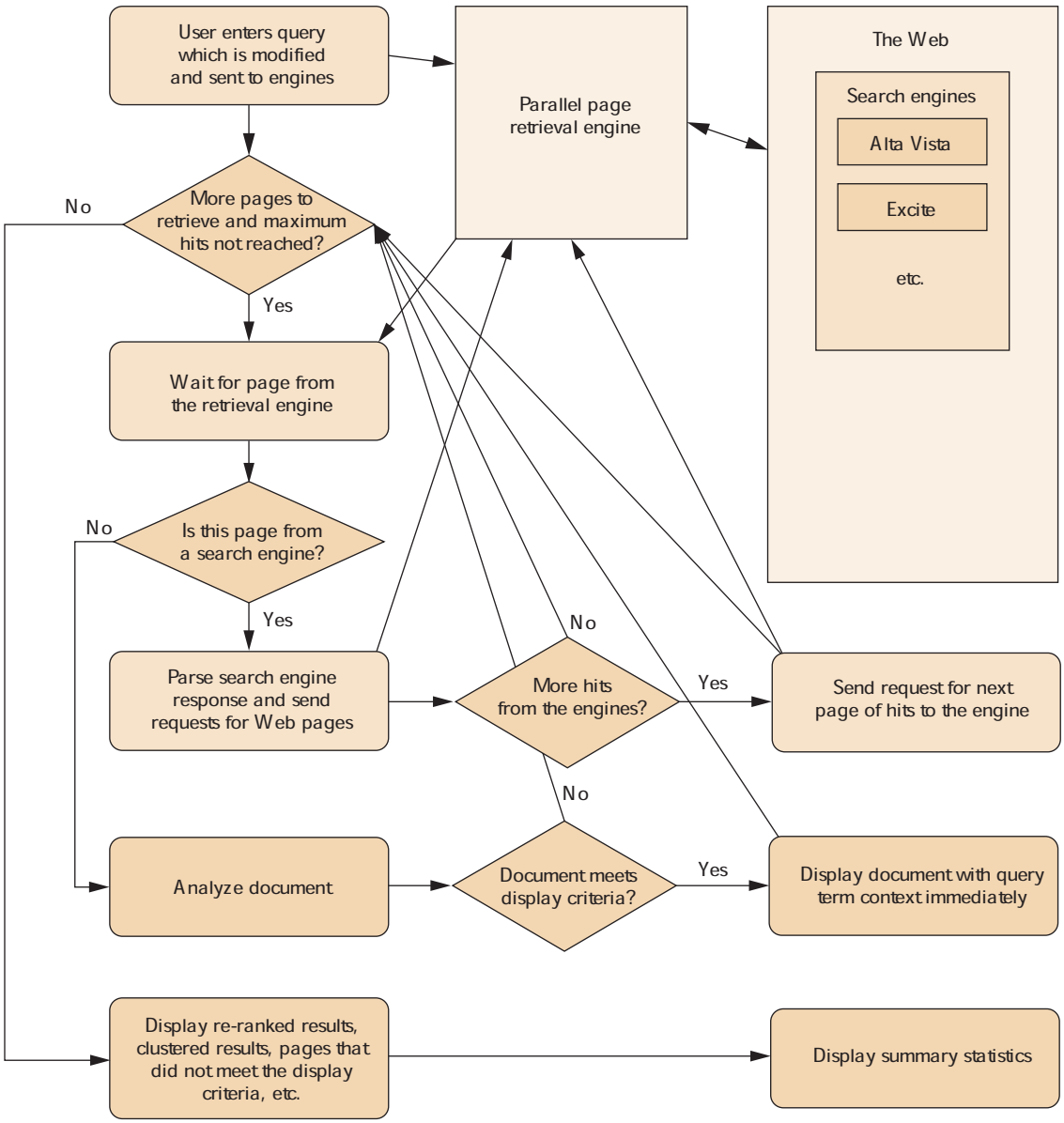


Figure 1. Simplified control flow of the metasearch engine.

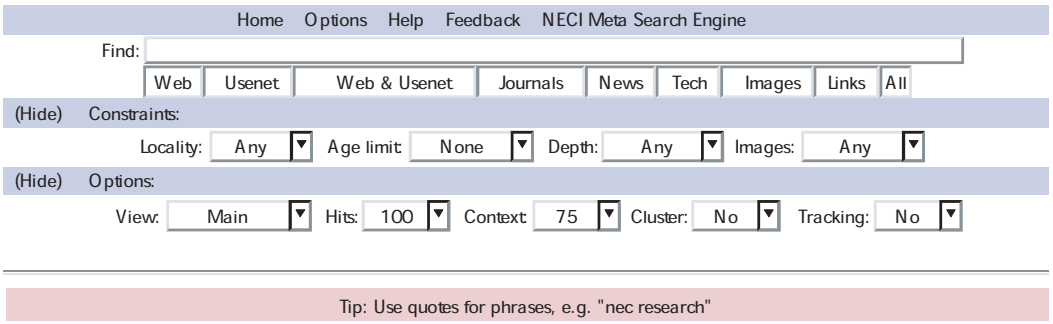


Figure 2. Search form for the NECI metasearch engine.

relatively simple, but does incorporate features such as queuing requests, load balancing from multiple search processes, and delaying requests to the same site to prevent overloading a site.

Figure 2 shows the main search form for the NECI metasearch engine. Users can choose which search engines to run, how many hits to retrieve, the amount of context to display (measured in number of characters), and so on. The engine supports all common search formats, including Boolean syntax. As with many other metasearch engines, the NECI metasearch engine dynamically modifies queries to match each search engine's query syntax.

Users can control the amount of text the NECI engine displays by specifying the number of characters it will show on either side of the query terms. To improve readability, the engine omits most nonalphanumeric characters and partial words at the beginning and end of the specified character count. At one point, we sought to improve context display by extracting logical sentences rather than a fixed number of characters. However, in general, users did not find this sentence-based method superior because including full sentences increased the screen space needed by each summary without significantly improving users' ability to determine relevance.

Because the NECI engine returns results progressively as it downloads and analyzes each page, the results are not necessarily displayed in the order listed by the individual search engines, but the order is approximately the same. Perhaps because Web search engines are not good at relevance ranking to begin with, this difference in document ranking was not a problem for users.

Figure 3 shows a sample response of the NECI metasearch engine for the query "digital watermark." The bar at the top lets users switch between views of the search results; below it are links to the individual engine results. The "tip" that follows might be query sensitive, such as providing specific query format suggestions when the query looks like a proper name.

The shaded bars to the left of the document titles indicate how close query terms are to each other in the document. With a single query term, the bar shading indicates how close the term is to the top of the document. The information to the right of the document title shows which engine found the document and the document's age, for example, in the first listing, "A" refers to AltaVista, "n/a" indicates that the document's age is not available.

Specific Expressive Forms

Information on the Web is often duplicated and expressed in a variety of forms. If all information was (correctly) expressed in all possible ways, precise information retrieval would be simple: A search for any one particular way of expressing the information would succeed.

The NECI engine recognizes and transforms certain queries submitted in the form of a question into queries phrased in the form of an answer—*specific expressive forms* (SEFs). For example, the query "What does NASDAQ stand for?" is transformed into the query "NASDAQ stands for" "NASDAQ is an abbreviation" "NASDAQ means." Clearly the information may be expressed in forms other than these, but if the information exists in just one of these forms, it is more likely to satisfy the query. The technique thus trades recall for precision.

Our informal experiments indicate that using SEFs is effective for certain retrieval tasks on the Web. Figure 4 shows the NECI engine's results for the query "What does NASDAQ stand for?" The answer to the query is contained in the local context displayed for four out of the first five pages. In contrast, the standard search engines we queried did not have the answer in any of the documents

listed on the first page—even for engines that list support for natural language queries.

As the amount of easily accessible information increases, so too will the viability of the SEF technique. An extension to it that we have not yet implemented is to define an order over the various SEFs. For example, "*x* stands for" might be more likely to find the answer than "*x* means." If none of the SEFs are found, the engine could fall back to a standard query.

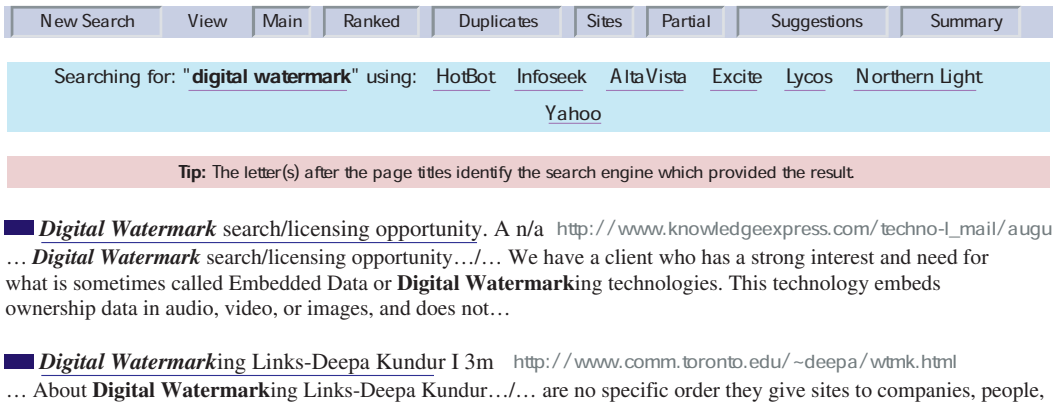


Figure 3. Sample response of the NECI metasearch engine for the query "digital watermark."

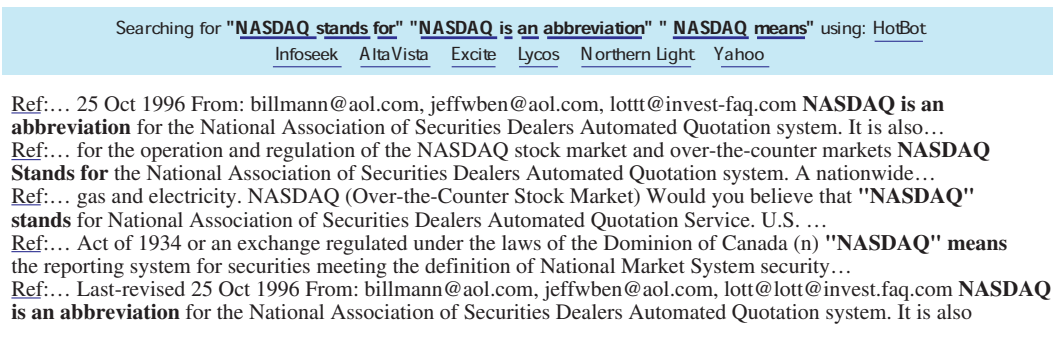


Figure 4. NECI metasearch engine response for the query "What does NASDAQ stand for?"

Currently, the NECI metasearch engine uses the SEF technique for a number of queries. For example, the engine recognizes "What [is|are] *x*?," "What [causes|creates|produces] *x*?," "What does *x* [stand for|mean]?," and "[Why|how] [is|are] (a|the) *x y*?" As examples of the transformations, "What does *x* [stand for|mean]?" is converted to "*x* stands for," "*x* is an abbreviation," and "*x* means"; and "What [causes|creates|produces] *x*?" is converted to "*x* is caused," "*x* is created," "causes *x*," "produces *x*," "makes *x*," and "creates *x*." Although we created the SEF transformations manually, an interesting area of research would be to learn SEFs from implicit or explicit feedback.

The NECI engine downloads and analyzes the actual pages, so it can apply a uniform ranking measure to documents returned by different engines.

The SEF technique often relies on the engine's ability to search for a phrase containing what are typically "stop" words. These words are almost universally filtered out by traditional information retrieval systems. Web search engines vary in their use of stop words, and we have found it necessary to filter out certain phrases on an engine-by-engine basis to prevent the engines from returning many pages that do not contain the phrases.

Results Ranking

Steve Kirsch has proposed a ranking scheme whereby the underlying search engines are modified to return additional information, such as how many times a term occurs in each document and the entire database.[7] With the NECI engine, this step is unnecessary as it downloads and analyzes the actual pages. It can therefore apply a uniform ranking measure to documents returned by different engines. Currently, the engine displays documents in descending order of query-term occurrence. If none of the first few pages contain all terms, the engine displays documents with the maximum number of query terms found so far.

Once all pages are downloaded, the engine relists documents according to a simple relevance measure. This measure considers the number of query terms in the document, the proximity between query terms, and term frequency (inverse document frequency can also be useful[8]). We use the following equation for pages containing more than one of the query terms; when only one query term is found we currently use the term's distance from the start of the page.

$$R = c_1 N_p + \frac{\left(c_2 - \frac{\sum_{i=1}^{N_p-1} \sum_{j=i+1}^{N_p} \min(d(i,j), c_2)}{\sum_{k=1}^{N_p-1} (N_p - k)} \right)}{\frac{c_2}{c_1}} + \frac{N_t}{c_3}$$

where N_p is the number of query terms that appear in the document (each term is counted only once); N_t is the total number of query terms in the document; $d(i, j)$ is the minimum distance between the ith and jth query terms (currently measured in number of characters); c_1 is a constant that controls the overall magnitude of R, which is the document's relevance score; c_2 is a constant that specifies the maximum useful distance between query terms; and c_3 is a constant that specifies term-frequency importance (currently $c_1 = 100$, $c_2 = 5000$, and $c_3 = 10c_1$).

This ranking criterion is particularly useful for Web searches. Because the Web database is so large and diverse, searching for multiple terms can return documents that use the terms in unrelated sections, such as terms that exist in different bookmarks on a bookmarks page.

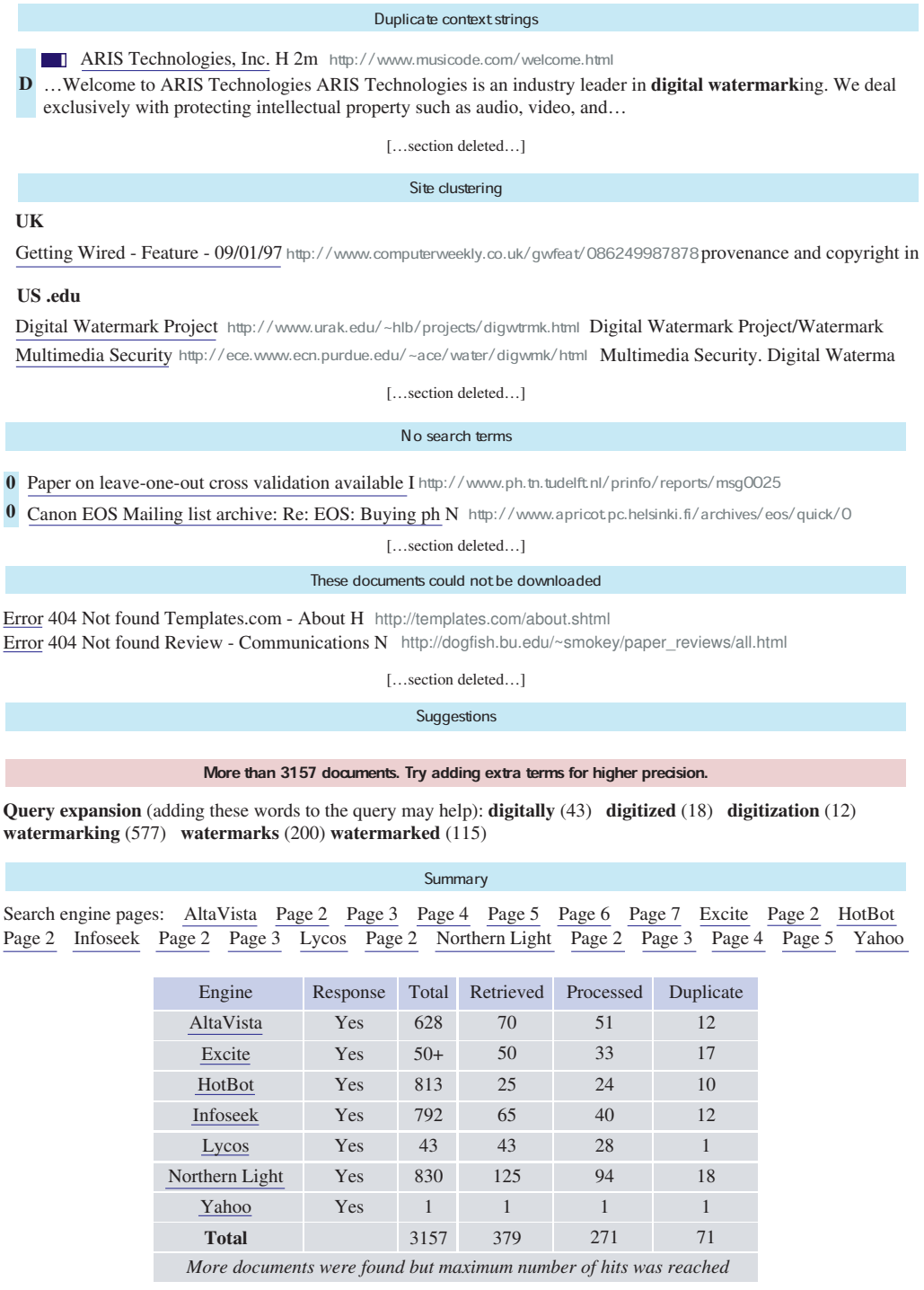
After all pages have been retrieved, the engine displays the top 30 pages ranked by term proximity. As Figure 5 shows, the engine then displays additional information: duplicate context strings, results clustered by site, documents with fewer or no search terms, and pages that could not be downloaded. It also displays a summary table with results for each engine queried and suggestions for subsequent queries, as the sidebar "Improving User Queries" on p. 44 describes.

These added features are important. Where other metasearch engines categorize pages as duplicate if the normalized URLs are identical, the NECI metasearch engine considers pages duplicate if the relevant context strings are identical. Thus, even duplicate pages with different headers and footers will be detected, such as when a single mailing list message is archived in several places. Knowing which pages do not match the query or are not available is

also important. Different engines use different relevance techniques; if one engine returns poor relevance results, it can lead to poor overall results from standard metasearch engines. Other metasearch services also provide "dead link" detection, but this feature is typically turned off by default or does not return results until all pages are checked.

Document Display

Figure 6 (on p. 45) shows a sample document from



Engine	Response	Total	Retrieved	Processed	Duplicate
AltaVista	Yes	628	70	51	12
Excite	Yes	50+	50	33	17
HotBot	Yes	813	25	24	10
Infoseek	Yes	792	65	40	12
Lycos	Yes	43	43	28	1
Northern Light	Yes	830	125	94	18
Yahoo	Yes	1	1	1	1
Total		3157	379	271	71
More documents were found but maximum number of hits was reached					

Figure 5. Additional information, including duplicate context strings, results clustered by site, and pages that could not be downloaded, are displayed after the query is complete.

the "digital watermark" search. The links at the top jump to the first occurrence of the query terms in the document, and indicate the number of occurrences. Each query term within the text also links to the next use of the term. Such linking and highlighting helps users quickly identify page relevance. The NECI engine can also track query results and page contents, automatically informing users when new matching documents are found or when a given page has been modified ("Track page").

Currently, the NECI engine uses two forms of caching. The engine caches all downloaded pages for a limited time period, and query terms and links are added on demand. The engine also caches the top 20 relevance-ranked results from each query. If a user repeats the query, these pages are the first displayed—if they still exist and contain the query terms.

IMPROVING USER QUERIES

Our analysis of 9,000 queries during the second half of 1997 showed that most queries contained only a few terms (Figure A shows the total distribution). Because simple queries often generate thousands of matching documents and poor precision in the results, we built the NECI engine to suggest query improvements to the user. For example,

- for queries that do not specify phrases, the engine looks for combinations of the query terms appearing as phrases and suggests the use of a phrase if a threshold is exceeded;
- for multi-term queries where no terms are required, the engine suggests the use of "+" or "and" to require terms; and
- the engine stems the query terms and searches the pages for morphological variants. If any are found it suggests them as terms that can be added to the query.

The first two suggestions are aimed at improving precision; the third, at improving recall.

Suggesting that users introduce phrases and term requirements may seem counterintuitive from a traditional information retrieval viewpoint, as these suggestions can exclude many relevant documents. However, the Web poses different information retrieval problems from those posed by traditional databases, because it is larger and more diverse, with a lower signal-to-noise ratio. As a result, in Web searches it is often useful to trade recall (the number of documents returned) for improved precision.

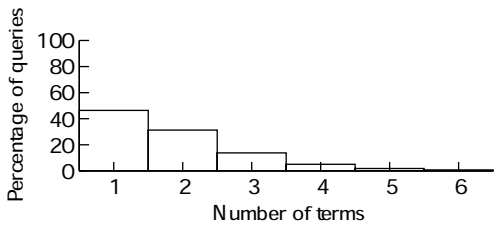


Figure A. The distribution of the number of terms contained in queries.

IMPLEMENTATION

The NECI metasearch engine is currently implemented for server operation at NEC Research Institute, where it serves about 100 users. A client implementation could also be created, which would improve scalability. The disadvantages are increased processing and memory requirements, and the need to update all clients when modifications are made to the metasearch engine. A client implementation would also decrease the caching benefits.

Resource Requirements

The NECI search engine uses roughly an order of magnitude more bandwidth than other search engines. These bandwidth requirements could limit the number of users that can simultaneously use a server-based implementation.

However, these requirements are not as great as those required by other Web developments, such as the increasing use of audio and video, and bandwidth and access times on the Internet continue to improve.[9] Also, the engine will not necessarily need to analyze more pages per query as the Web grows (though precise queries will become more important).

The prototype engine runs on a Pentium Pro 200 PC, is written in Perl, and is not optimized for efficiency. When only a few queries are executed at a time using our prototype engine, the analysis does not typically slow the response (network response time is the limiting factor).

Performance

We analyzed the response time of the following six search engines: AltaVista, Excite, HotBot, Infoseek, Lycos, and Northern Light. The median response time from 3,000 queries to these engines during November-December 1997 was 1.9 seconds. However, if queries are made to all of the engines simultaneously, then the median time for the first engine to respond was 0.7 seconds. A similar advantage is gained by downloading the Web pages corresponding to the hits in parallel, resulting in a median time for the NECI engine to receive the first page being 1.3 seconds. On average, the parallel architecture of

Jump to: **digital watermark** (4) http://www.comm.toronto.edu/~deepa/wtmk.html [Track Page] [No Images]

Watermarking Links

This page is currently under MASSIVE construction! There are many more links to come… and much more organizing to do… The links on this page are in no specific order, they give sites to companies, people, projects and articles related to digital watermarking.

If you are currently working in the area of digital watermarking, please feel free to let me know by e-mail at deepa@comm.toronto.edu and I'll be happy to put your name/site on the list.

[…section deleted…]

Figure 6. Sample page view for the NECI metasearch engine. Query terms are highlighted; links take users to the first occurrence of the query term.

the NECI engine allows it to find, download, and analyze the first page faster than the standard search engines can respond, even though the standard engines do not download and analyze the current contents of the pages.

In May 1998 we analyzed the time for the engine to display the first five and first 10 relevant results from 200 queries. The median time for the first five relevant results was 2.7 seconds, and the median time for the first 10 relevant results was 3.2 seconds (these figures do not include queries that did not return the target number of results).

CONCLUSION

The NECI metasearch engine demonstrates that real-time downloading and analysis of the pages that match a query is possible. In fact, by calling the Web search engines and downloading Web pages in parallel, the NECI metasearch engine can, on average, display the first result quicker than a standard search engine.

Like other metasearch engines and various Web tools, the NECI metasearch engine relies on the underlying search engines for important and valuable services. Wide use of this or any metasearch engine requires an amiable arrangement with the underlying search engines; such arrangements might include passing through ads or micro-payment systems.

There are numerous areas for future research. Because the NECI engine collects the full text of matching documents, it is a good test bed for information retrieval research. Areas we are working on include clustering, query expansion, and relevance feedback. Because the query-sensitive summaries let users better assess relevance without having to view pages, implicit feedback should be more successful and might be useful for improved relevance measures, automatic relevance feedback, and learning specific expressive forms. Other areas we are looking at include page classification and extending the specific-expressive-forms search technique. ■

ACKNOWLEDGMENTS

We thank Eric Baum, Kurt Bollacker, Adam Grove, Bill Horne, Bob Krovetz, Roy Lipski, John Oliensis, Steve Omohundro, Maximilian Ott, James Philbin, Majd Sakr, Lance Williams, the employees of NECI, and the anonymous reviewers for useful comments and suggestions.

REFERENCES

1. E. Selberg and O. Etzioni, "Multi-Service Search and Comparison Using the MetaCrawler," *Proc. 1995 WWW Conf.*, 1995; available online at http://draz.cs.washington.edu/papers/www4/html/Overview.html.
2. S. Lawrence and C.L. Giles, "Searching the World Wide Web," *Science*, Vol. 280, No. 5360, 1998, p. 98.
3. D. van Eylen, "AltaVista Ranking of Query Results," available at http://www.ping.be/dirk_van_eylen/avrank.html, 1998.
4. D. Dreilinger and A. Howe, "An Information Gathering Agent for Querying Web Search Engines," Tech. Report CS-96-111, Computer Science Dept., Colorado State Univ., Fort Collins, Colo., 1996.
5. E. Selberg and O. Etzioni, "The MetaCrawler Architecture for Resource Aggregation on the Web," *IEEE Expert*, Jan.-Feb. 1997, pp. 11-14; also available at http://www.cs.washington.edu/homes/speed/papers/ieee/ieee-metacrawler.ps.

RELATED WORK

The idea of querying and collating results from multiple databases is not new. Companies such as PLS (http://www.pls.com), Lexis-Nexis (http://www.lexis-nexis.com), Dialog (http://www.dialog.com), and Verity (http://www.verity.com) long ago created systems that integrated search results from multiple heterogeneous databases.[1] There are many existing Web metasearch services, including MetaCrawler, SavvySearch, Inference Find, Fusion, ProFusion, Highway 61, Mamma, Quarterdeck WebCompass, Metabot, Symantec Internet FastFind, and WebSeeker (for a quick review of metasearch engines, see Notess[2]).

Work in the area of "collection fusion" is reported in the Text Retrieval Conference (TREC) and the Special Interest Group for Information Retrieval (SIGIR) conference proceedings. Several other researchers have also used relevance measures including term proximity.[3,4]

Research search engines that promise improved results ranking include Laser[5] (http://laser.cs.cmu.edu/) and Google[6] (http://google.stanford.edu). These engines use the structure of HTML pages and hyperlink information to help determine page relevancy. For example, Google uses the text in links to a particular page as descriptors of that page (links often contain better descriptions of the page than the pages themselves). Google also uses a ranking algorithm called PageRank, which bases rankings on analysis of the number of pages pointing to each page. Although most of the benefits of metasearch apply to these improved search engines, displaying query term context may become less important for determining page relevancy as results rankings improve.

REFERENCES

1. E. Selberg and O. Etzioni, "Multi-Service Search and Comparison Using the MetaCrawler," *Proc. 1995 WWW Conf.*, 1995; available online at http://draz.cs.washington.edu/papers/www4/html/Overview.html.
2. G.R. Notess, "Internet 'Onesearch' With the Mega Search Engines," *Online*, Vol. 20, No. 6, 1996, pp. 36-39.
3. E. Keen, "Term Position Ranking: Some New Test Results," *Proc. 15th Int'l ACM SIGIR Conf. Research and Development in Information Retrieval*, 1992, pp. 66-76; available online at http://www.acm.org/pubs/citations/proceedings/ir/133160/p66-keen/.
4. D. Hawking and P. Thistlewaite, "Proximity Operators—So Near and Yet So Far," *Proc. Fourth Text Retrieval Conf.*, D.K. Harman, ed., 1995; available online at http://web.soi.city.ac.uk/~andym/PADRE/trec4.ps.Z.
5. J. Boyan, D. Freitag, and T. Joachims, "A Machine-Learning Architecture for Optimizing Web Search Engines," *Proc. AAAI Workshop Internet-Based Information Systems*, 1996; available online at http://www.lb.cs.cmu.edu/afs/cs/project/reinforcement/papers/boyan.laser.ps.
6. S. Brin and L. Page, "The Anatomy of a Large-Scale Hypertextual Web Search Engine," *Proc. 1998 WWW Conf.*, 1998; available online at http://google.stanford.edu/~backrub/google.html.
6. A. Tombros, *Reflecting User Information Needs Through Query Biased Summaries*, doctoral thesis, Dept. Computer Science, Univ. of Glasgow, 1997.
7. S.T. Kirsch, "Document Retrieval over Networks Wherein Ranking and Relevance Scores are Computed at the Client for Multiple Database Documents," US Patent #5,659,732, US Patent Office, 1997.
8. G. Salton, *Automatic Text Processing: The Transformation, Analysis and Retrieval of Information by Computer*, Addison Wesley Longman, Reading, Mass., 1989.
9. "1997 Internet Performance Wrap-Up—How Well Did It Perform? Keynote Systems Shows All," *Business Wire*, Jan. 9, 1998; available at http://www.keynote.com/measures/business/business40.html.

Steve Lawrence is a scientist at the NEC Research Institute in Princeton, N.J. His research interests include information retrieval, machine learning, neural networks, face recognition, speech recognition, time series prediction, and natural language. His awards include an NEC Research Institute excellence award, ATERB and APRA priority scholarships, a QUT university medal and award for excellence, QEC and Telecom Australia Engineering prizes, and three successive prizes in the annual Australian Mathematics Competition. Lawrence received a BSc in computing and a BEng in electronic systems from the Queensland University of Technology, Australia, and a PhD in electrical and computer engineering from the University of Queensland, Australia.

C. Lee Giles is a senior research scientist in computer science at NEC Research Institute, Princeton, N.J., and an adjunct professor at the Institute for Advanced Computer Studies at the University of Maryland, College Park. His research interests are in novel applications of neural computing, machine learning, agents, and AI in all areas of computing. He is on the editorial boards of *IEEE Intelligent Systems*, *IEEE Transactions on Knowledge and Data Engineering*, *IEEE Transactions on Neural Networks*, *J. of Computational Intelligence in Finance*, *J. of Parallel and Distributed Computing*, *Neural Networks*, *Neural Computation*, and *Applied Optics*. Giles is a Fellow of the IEEE and a member of AAAI, ACM, the International Neural Network Society, the Optical Society of America, and the Center for Discrete Mathematics and Theoretical Computer Science, Rutgers University.

Contact Lawrence and Giles at NEC Research Institute, 4 Independence Way, Princeton, NJ 08540; {lawrence, giles}@research.nj.nec.com.