Bulletin of the Technical Committee on

Data Engineering

September 2000 Vol. 23 No. 3 **IEEE Computer Society**

Letters

Special Issue on Next-Generation Web Search

Announcements and Notices

Editorial Board

Editor-in-Chief

David B. Lomet
Microsoft Research
One Microsoft Way, Bldg. 9
Redmond WA 98052-6399
lomet@microsoft.com

Associate Editors

Luis Gravano
Computer Science Department
Columbia University
1214 Amsterdam Avenue
New York, NY 10027

Alon Levy
University of Washington
Computer Science and Engineering Dept.
Sieg Hall, Room 310
Seattle, WA 98195

Sunita Sarawagi
School of Information Technology
Indian Institute of Technology, Bombay
Powai Street
Mumbai, India 400076

Gerhard Weikum
Dept. of Computer Science
University of the Saarland
P.O.B. 151150, D-66041
Saarbrücken, Germany

The Bulletin of the Technical Committee on Data Engineering is published quarterly and is distributed to all TC members. Its scope includes the design, implementation, modelling, theory and application of database systems and their technology.

Letters, conference information, and news should be sent to the Editor-in-Chief. Papers for each issue are solicited by and should be sent to the Associate Editor responsible for the issue.

Opinions expressed in contributions are those of the authors and do not necessarily reflect the positions of the TC on Data Engineering, the IEEE Computer Society, or the authors' organizations.

Membership in the TC on Data Engineering is open to all current members of the IEEE Computer Society who are interested in database systems.

The Data Engineering Bulletin web page is http://www.research.microsoft.com/research/db/debull.

TC Executive Committee

Chair

Betty Salzberg
College of Computer Science
Northeastern University
Boston, MA 02115
salzberg@ccs.neu.edu

Vice-Chair

Erich J. Neuhold
Director, GMD-IPSI
Dolivostrasse 15
P.O. Box 10 43 26
6100 Darmstadt, Germany

Secretry/Treasurer

Paul Larson
Microsoft Research
One Microsoft Way, Bldg. 9
Redmond WA 98052-6399

SIGMOD Liason

Z.Meral Ozsoyoglu
Computer Eng. and Science Dept.
Case Western Reserve University
Cleveland, Ohio, 44106-7071

Geographic Co-ordinators

Masaru Kitsuregawa (**Asia**)
Institute of Industrial Science
The University of Tokyo
7-22-1 Roppongi Minato-ku
Tokyo 106, Japan

Ron Sacks-Davis (**Australia**)
CITRI
723 Swanston Street
Carlton, Victoria, Australia 3053

Svein-Olaf Hvasshovd (**Europe**)
ClustRa
Westermannsveita 2, N-7011
Trondheim, NORWAY

Distribution

IEEE Computer Society
1730 Massachusetts Avenue
Washington, D.C. 20036-1992
(202) 371-1013
nschoultz@computer.org

Letter from the Editor-in-Chief

Please Read: Bulletin Formats

In an effort to simplify the preparation of the Bulletin, and to reduce disk storage on our server, I would like to eliminate some of the versions of the Bulletin that are available on this web site (and via ftp). Specifically, I propose to *eliminate the draft versions* of the Bulletin. These are the versions in which many figures are not included. The purpose of the draft version was to increase the probability that something could be successfully printed, even if the figure could not. I have heard no complaints from anyone about their ability to display or print the Bulletin. The current issue (September, 2000) will include draft versions on the web site. But I intend to discontinue them for subsequent issues. Indeed, I intend to remove all draft versions from the Bulletin web site. If you believe you need draft versions, please send me mail at `lomet@microsoft.com`. I am willing to re-consider this should there be sufficient concern expressed about this. But please, only send me comments if you actually depend upon the availability of the draft versions.

About the Current Issue

For those of us (most of us) who are regularly "on-line", web searching has become an everyday part of our lives. And it is surely not only the technical community, but people everywhere who are benefitting from web search technology. Indeed, an important part of the usefulness of the web lies exactly in the ability to find information that very frequently derives from many sources from all over the world. So the impact of effective web search has been enormous, and will continue to grow in importance over time, as even more people access ever more information made available at web sites around the world.

Of course, the challenging nature of web search and its commercial opportunities have not escaped the attention of both the research and the industrial database communities. Indeed, the distinction between research and industrial communities has all but disappeared in many cases as researchers frequently join startups in the exploitation and improvement of web searching. The field is moving very rapidly, and those with database skills are playing a substantial role in the development of the field. After all, search is one of the things that database folks have been doing for over thirty years.

Luis Gravano is the editor for the current issue. He has assembled an issue that includes articles describing work going on in both research labs and industrial enterprises. Much of this work is an effort to move beyond "vanilla" key word search by considering "meta" information derived from the way that information is organized on the web, both for the semantic relevance of the result and for assessing the quality of the information. Luis has done a fine job in assembling this very interesting and timely collection of articles on web search, and I thank him for this. I am confident that readers will find this issue to be both interesting and useful.

David Lomet
Microsoft Corporation

Letter from the Special Issue Editor

Web search engines have made substantial progress in helping users find information effectively. Nowadays a few select search engines answer tens of millions of queries a day over roughly a billion web pages, taking on average under a second to process a query. At the same time, overall user satisfaction for some of these engines is high. However, web search technology is far from mature, and there is still plenty of room for improvement. More specifically, users are often overwhelmed with query results that include many irrelevant pages. An important on-going challenge for search engines is to guess what users are after given only very few words as input. Also, current search engines ignore the often valuable contents of search-only text databases available on the web, since crawlers cannot download and index the contents of such databases. Recent studies have estimated the size of this "hidden web" to be several orders of magnitude that of the "crawlable web." On a related note, web pages sometimes include relatively structured information that would be most useful if extracted and made available via specialized interfaces that exploited this structure. The papers in this issue of the Bulletin address the research challenges in building next-generation web search engines with intuitive, expressive interfaces for all kinds of web-accessible information.

The first paper in this issue, by Monika Henzinger (Google, Inc.), surveys recent work on exploiting the link structure of the web to improve the quality of search results. Before the web existed, traditional information retrieval research studied for decades how to identify the text documents that are topically most relevant to a given query. A key assumption underlying this research was that the collection of available text documents was reasonably uniform in quality, value, and intended audience. This assumption does not hold over the web, hence the importance of adding other factors (e.g., popularity) when producing query results. The second paper, by Alberto Mendelzon (U. of Toronto) and Davood Rafiei (U. of Alberta), shows a novel use of the topology of the web to compute the set of topics on which a given web page has high "reputation." The third paper, by William Cohen, Andrew McCallum, and Dallan Quass (WhizBang! Labs–Research), discusses how to build topic-specific search engines that identify, extract, and integrate structured information present on web pages. Such specialized search engines can then let users ask complex queries that are appropriate for the domain and topic of the engines. The fourth paper, by Steve Lawrence (NEC Research), argues that next-generation search engines should consider the user's "context" to choose what resources (e.g., specialized search engines) to return and exploit. The fifth paper, by Jamie Callan (Carnegie Mellon U.), describes distributed search models for search engines, where the evaluation of queries potentially spans a number of autonomous, searchable databases, rather than a single large, monolithic, centralized collection as with current search engines. Finally, the paper by Marti Hearst (U. of California, Berkeley) discusses how to build effective interfaces for the different flavors of web resources.

Web search is a unique research area in its potential to directly impact the way hundreds of millions of users interact with information. I hope you will find the six papers in this issue of the Bulletin to be a good sample of the exciting research that is being pursued in this area in academia, research labs, and start-up companies.

Luis Gravano
Columbia University

Link Analysis in Web Information Retrieval

Monika Henzinger
Google Incorporated
Mountain View, California
monika@google.com

Abstract

The analysis of the hyperlink structure of the web has led to significant improvements in web information retrieval. This survey describes two successful link analysis algorithms and the state-of-the art of the field.

1 Introduction

The goal of information retrieval is to find all documents relevant for a user query in a collection of documents. Decades of research in information retrieval were successful in developing and refining techniques that are solely word-based (see e.g., [2]). With the advent of the web new sources of information became available, one of them being the *hyperlinks* between documents and records of user behavior. To be precise, *hypertexts* (i.e., collections of documents connected by hyperlinks) have existed and have been studied for a long time. What was new was the large number of hyperlinks created by independent individuals. Hyperlinks provide a valuable source of information for web information retrieval as we will show in this article. This area of information retrieval is commonly called *link analysis*.

Why would one expect hyperlinks to be useful? A hyperlink is a reference of a web page B that is contained in a web page A. When the hyperlink is clicked on in a web browser, the browser displays page B. This functionality alone is not helpful for web information retrieval. However, the way hyperlinks are typically used by authors of web pages can give them valuable information content. Typically, authors create links because they think they will be useful for the readers of the pages. Thus, links are usually either navigational aids that, for example, bring the reader back to the homepage of the site, or links that point to pages whose content augments the content of the current page. The second kind of links tend to point to high-quality pages that might be on the same topic as the page containing the link.

Based on this motivation, link analysis makes the following simplifying assumptions:

- A link from page A to page B is a recommendation of page B by the author of page A.
- If page A and page B are connected by a link the probability that they are on the same topic is higher than if they are not connected.

Copyright 2000 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.
Bulletin of the IEEE Computer Society Technical Committee on Data Engineering

Link analysis has been used successfully for deciding which web pages to add to the collection of documents (i.e., which pages to *crawl*), and how to order the documents matching a user query (i.e., how to *rank* pages). It has also been used to categorize web pages, to find pages that are related to given pages, to find duplicated web sites, and various other problems related to web information retrieval.

The idea of studying "referrals" is, however, not new. A subfield of classical information retrieval, called bibliometrics, analyzed citations (see, e.g., [19, 14, 29, 15]). The field of sociometry developed algorithms [20, 25] very similar to the PageRank and HITS algorithms described below. Some link analysis algorithms can also be seen as collaborative filtering algorithms: each link represents an opinion and the goal is to mine the set of opinions to improve the answers to individuals.

This paper is structured as follows. We first discuss graph representations for the web (Section 2). In Section 3 we discuss two types of connectivity-based ranking schemata: a *query-independent* approach, where a score measuring the intrinsic quality of a page is assigned to each page without a specific user query, and a *query-dependent* approach, where a score measuring the quality and the relevance of a page to a given user query is assigned to some of the pages. In Section 4 other uses of link analysis in web information retrieval are described.

2 A Graph Representation for the Web

In order to simplify the description of the algorithms below we first model the web as a graph. This can be done in various ways. Connectivity-based ranking techniques usually assume the most straightforward representation: The graph contains a node for each page u and there exists a directed edge (u, v) if and only if page u contains a hyperlink to page v. We call this directed graph the *link graph* G.

Some algorithms make use of the undirected *co-citation graph*: As before each page is represented by a node. Nodes u and v are connected by an undirected edge if and only if there exists a third node x linking to both u and v.

The link graph has been used for ranking, finding related pages, and various other problems. The co-citation graph has been used for finding related pages and categorizing pages.

3 Connectivity-Based Ranking

3.1 Query-Independent Connectivity-Based Ranking

In *query-independent* ranking a score is assigned to each page without a specific user query with the goal of measuring the intrinsic quality of a page. At query time this score is used with or without some query-dependent criteria to rank all documents matching the query.

The first assumption of connectivity based techniques immediately leads to a simple query-independent criterion: The larger the number of hyperlinks pointing to a page the better the page. The main drawback of this approach is that each link is equally important. It cannot distinguish between the quality of a page pointed to by a number of low-quality pages and the quality of a page that gets pointed to by the same number of high-quality pages. Obviously, it is therefore easy to make a page appear to be high-quality – just create many other pages that point to it.

To remedy this problem Brin and Page [5, 26] invented the PageRank measure. The PageRank of a page is computed by weighting each hyperlink proportionally to the quality of the page containing the hyperlink. To determine the quality of a referring page, they use its PageRank recursively. This leads to the following definition of the PageRank $R(p)$ of a page p:

$$R(p) = \epsilon/n + (1-\epsilon) \cdot \sum_{(q,p)\in G} R(q)/outdegree(q),$$

where

- ϵ is a dampening factor usually set between 0.1 and 0.2;
- n is the number of nodes in G; and
- $outdegree(q)$ is the number of edges leaving page p, i.e., the number of hyperlinks on page q.

Alternatively, the PageRank can be defined to be the stationary distribution of the following infinite random walk $p_1, p_2, p_3, \ldots$, where each p_i is a node in G: Each node is equally likely to be the first node p_1. To determine node p_{i+1} with $i > 0$ a biased coin is flipped: With probability ϵ node p_{i+1} is chosen uniformly at random from all nodes in G, with probability $1 - \epsilon$ node p_{i+1} is chosen uniformly at random from all nodes q such that edge (p_i, q) exists in G.

The PageRank is the dominant eigenvector of the probability transition matrix of this random walk. This implies that when PageRank is computed iteratively using the above equation, the computation will eventually converge under some weak assumptions on the values in the probability transition matrix. No bounds are known on the number of iterations but in practice roughly 100 iterations suffice.

The PageRank measure works very well in distinguishing high-quality web pages from low-quality web pages and is used by the Google[1] search engine.

The PageRank algorithm assigns a score to each document independent of a specific query. This has the advantage that the link analysis is performed once and then can be used to rank all subsequent queries.

3.2 Query-Dependent Connectivity-Based Ranking

In *query-dependent* ranking a score measuring the quality and the relevance of a page to a given user query is assigned to some of the pages.

Carriere and Kazman [11] proposed an indegree-based ranking approach to combine link analysis with a user query. They build for each query a subgraph of the link graph G limited to pages on the query topic. More specifically, they use the following query-dependent *neighborhood graph*. A *start set* of documents matching the query is fetched from a search engine (say the top 200 matches). This set is augmented by its *neighborhood*, which is the set of documents that either point to or are pointed to by documents in the start set. Since the indegree of nodes can be very large, in practice a limited number of predecessors (say 50) of a document are included. The neighborhood graph is the subgraph of G induced by the documents in the start set and its neighborhood. This means that each such document is represented by a node u and there exists an edge between two nodes u and v in the neighborhood graph if and only if there is a hyperlink between them. The indegree-based approach then ranks the nodes by their indegree in the neighborhood graph. As discussed before this approach has the problem that each link counts an equal amount.

To address this problem, Kleinberg [21] invented the *HITS* algorithm. Given a user query, the algorithm first iteratively computes a *hub* score and an *authority* score for each node in the neighborhood graph[2]. The documents are then ranked by hub and authority scores, respectively.

Nodes, i.e., documents that have high authority scores are expected to have relevant content, whereas documents with high hub scores are expected to contain hyperlinks to relevant content. The intuition is as follows. A document which points to many others might be a good hub, and a document that many documents point to might be a good authority. Recursively, a document that points to many good authorities might be an even better hub, and similarly a document pointed to by many good hubs might be an even better authority. This leads to the following algorithm.

(1) Let N be the set of nodes in the neighborhood graph.

[1] `http://www.google.com/`

[2] In the HITS algorithm the neighborhood graph is slightly modified to exclude edges between nodes on the same host. The reason is that hyperlinks within the same host might be by the same author and hence might not be a recommendation.

```
(2) For every node $n$ in $N$, let $H[n]$ be its hub score and
    $A[n]$ its authority score.
(3) Initialize $H[n]$ to 1 for all $n$ in $N$.
(4) While the vectors $H$ and $A$ have not converged:
(5)         For all $n$ in $N$, $A[n] := \sum_{(n',n) \in N} H[n']$
(6)         For all $n$ in $N$, $H[n] := \sum_{(n,n') \in N} A[n']$
(7)         Normalize the $H$ and $A$ vectors.
```

Since this algorithm computes the dominant eigenvectors of two matrices, the H and A vectors will eventually converge, but no bound on the number of iterations is known. In practice, the vectors converge quickly.

Note that the algorithm does not claim to find *all* valuable pages for a query, since there may be some that have good content but have not been linked to by many authors or that do not belong to the neighborhood graph.

There are two types of problems with this approach: First, since it only considers a relatively small part of the web graph, adding edges to a few nodes can potentially change the resulting hubs and authority scores considerably. Thus it is easier for authors of web pages to manipulate the hubs and authority scores than it is to manipulate the PageRank score. See [23] for a more extensive discussion of this problem. A second problem is that if the neighborhood graph contains more pages on a topic different from the query, then it can happen that the top authority and hub pages are on this different topic. This problem was called *topic drift*. Various papers [7, 8, 4] suggest the use of edge weights and content analysis of either the documents or the anchor text to deal with these problems. In a user study [4] it was shown that this can considerably improve the quality of the results.

A recent paper by Lempel and Moran [23] gives anecdotal evidence that a variant of the indegree-based approach achieves better results than the HITS algorithm. They compute the stationary distribution of a random walk on an auxiliary graph. This corresponds to scaling the indegree of a node u in the link graph by the relative size of u's connected component in the co-citation graph and the number of edges in u's component in the auxiliary graph. Basically, each link is weighted and the quality of a page is the sum of the weights of the links pointing to it. However, more experimental work is needed to evaluate this approach.

3.3 Evaluation of Query-Dependent Rankings

Amento, Terveen, and Hill [1] evaluated different link-based ranking criteria on a graph similar to the neighborhood graph. They start from a seed-set of relevant pages for a given query and their goal is to rank them by quality using various criteria.

The seed-set has the property that no url in the seed-set is the prefix of another one. They consider these urls to be *root urls* of *sites*: all pages which contain the root url as prefix belong to the site of this root url. Then they perform a neighborhood expansion using link and text similarity heuristics and restricting the expansion to pages on the above sites. For their analysis they use either this graph or a *site graph*, where all pages on a site are collapsed to one node. Note that the set of nodes in the site graph is fully determined by the seed-set and the neighborhood expansion is used only to determine the edges in the site graph.

They use five link-based metrics (in-degree, out-degree, HITS authority score, HITS hub score, and PageRank) and some other metrics to rank the root urls by either using the score assigned to the root url (in the pages-based graph) or to the site (in the site graph). Interestingly, the ranking on the site graph outperformed the ranking on the pages-based graph. Furthermore, there is a large overlap and correlation in the rankings of the in-degree, HITS authority score, and PageRank metric and these three metrics perform roughly equally well. They also outperform the other metrics together with another simple metric that counts the number of pages on a site that belong to the graph.

Note, however, that they perform the PageRank computation on a small graph, while the PageRank computation described before was performed on the whole link graph and the resulting PageRank values will most likely differ considerably.

4 Other Uses of Link Analysis in Web Information Retrieval

Apart from ranking, link analysis can also be used for deciding which web pages to add to the collection of web pages, i.e., which pages to crawl. A *crawler* (or *robot* or *spider*) performs a traversal of the web graph with the goal of fetching high-quality pages. After fetching a page, it needs to decide which page out of the set of uncrawled pages to fetch next. One approach is to crawl the pages with highest number of links from the crawled pages first. Cho et al. propose to visit the pages in the order of PageRank [10].

Link analysis was also used for a search-by-example approach to searching: given one relevant page find pages related to it. Kleinberg [21] proposed using the HITS algorithm for this problem and Dean and Henzinger [12] show that both the HITS algorithm and a simple algorithm on the co-citation perform very well. The idea behind the latter algorithm is that frequent co-citation is a good indication of relatedness and thus the edges with high weight in the co-citation graph tend to connect nodes with are related.

Extensions of the HITS and PageRank approaches were used by Rafiei and Mendelzon to compute the reputation of a web page [27] and by Sarukkai to predict personalized web usage [28].

Almost completely mirrored web hosts cause problems for search engines: they waste space in the index data structure and might lead to duplicate results. Bharat et al. [3] showed that a combination of IP address analysis, URL pattern analysis, and link structure analysis can detect many near-mirrors. The idea is that near-mirrors exhibit as very similar link structure within the host as well as to the other hosts.

Chakrabarti et al. [9] made first steps towards using the link structure for web page categorization.

In [17, 18] PageRank-like random walks were performed on the web to sample web pages almost according to the PageRank distribution and the uniformly distribution, respectively. The goal was to compute various statistics on the web pages and to compare the quality, respectively the number, of the pages in the indices of various commercial search engines.

Buyukkokten et al. [6] and Ding et al. [13] classify web pages based on their geographical scope by analyzing the links that point to the pages.

5 Conclusions

The main use of link analysis is currently in ranking query results. Other areas were link analysis has been shown to be useful are crawling, finding related pages, computing web page reputations and geographic scope, prediction link usage, finding mirrored host, categorizing web pages, and computing statistics of web pages and of search engines.

However, research of the hyperlink structure of the web is just at its beginning and a much deeper understanding needs to be gained.

References

[1] B. Amento, L. Terveen, and W. Hill. Does authority mean quality? Predicting expert quality ratings of web documents. In *Proceedings of the 23rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'00)*, pages 296–303.

[2] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley, 1999.

[3] K. Bharat, A. Z. Broder, J. Dean, and M. Henzinger. A comparison of techniques to find mirrored hosts on the World Wide Web. *Workshop on Organizing Web Space (WOWS)* in conjunction with *ACM Digital Library '99*. To appear in the *Journal of the American Society for Information Science,* 2000.

[4] K. Bharat and M. Henzinger. Improved algorithms for topic distillation in hyperlinked environments. In *Proceedings of the 21st International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'98)*, pages 111–104.

[5] S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine. In *Proceedings of the Seventh International World Wide Web Conference* 1998, pages 107–117.

[6] O. Buyukkokten, J. Cho, H. García-Molina, L. Gravano, and N. Shivakumar. Exploiting geographical location information of Web pages. *Proc. of the ACM SIGMOD Workshop on the Web and Databases (WebDB'99)*, 1999.

[7] S. Chakrabarti, B. Dom, D. Gibson, S. R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. Experiments in topic distillation. In *ACM–SIGIR'98 Post-Conference Workshop on Hypertext Information Retrieval for the Web*.

[8] S. Chakrabarti, B. Dom, P. Raghavan, S. Rajagopalan, D. Gibson and J. Kleinberg. Automatic resource compilation by analyzing hyperlink structure and associated text. In *Proceedings of the Seventh International World Wide Web Conference* 1998, pages 65–74.

[9] S. Chakrabarti, B. Dom, and P. Indyk. Enhanced hypertext categorization using hyperlinks. in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 1998, pages 307–318.

[10] J. Cho, H. García-Molina, and L. Page. Efficient crawling through URL ordering. In *Proceedings of the Seventh International World Wide Web Conference* 1998, pages 161–172.

[11] J. Carriere and R. Kazman. Webquery: Searching and visualizing the web through connectivity. In *Proceedings of the Sixth International World Wide Web Conference* 1997, pages 701–711.

[12] J. Dean and M. R. Henzinger. Finding related Web pages in the World Wide Web. In *Proceedings of the 8th International World Wide Web Conference* 1998, pages 389–401.

[13] J. Ding, L. Gravano, and N. Shivakumar. Computing geographical scopes of Web resources. *Proceedings of the 26th International Conference on Very Large Databases (VLDB'00)*, 2000.

[14] E. Garfield. Citation analysis as a tool in journal evaluation. *Science*, 178, 1972.

[15] E. Garfield. *Citation Indexing.* ISI Press, 1979.

[16] T. Haveliwala. Efficient computation of PageRank. Technical Report 1999-31, Stanford University, 1999.

[17] M. R. Henzinger, A. Heydon, M. Mitzenmacher, and M. Najork. Measuring search engine quality using random walks on the Web. In *Proceedings of the 8th International World Wide Web Conference* 1999, pages 213–225.

[18] M. R. Henzinger, A. Heydon, M. Mitzenmacher, and M. Najork. On near-uniform URL sampling. In *Proceedings of the Ninth International World Wide Web Conference* 2000, pages 295–308.

[19] M. M. Kessler. Bibliographic coupling between scientific papers. *American Documentation*, 14, 1963.

[20] L. Katz. A new status index derived from sociometric analysis. *Psychometrika*, 18(1):39-43, March 1953.

[21] J. Kleinberg. Authoritative sources in a hyperlinked environment. In *Proceedings of the 9th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 668–677, January 1998.

[22] J. Kleinberg, S. R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. The Web as a graph: Measurements, models and methods. Invited survey at the *International Conference on Combinatorics and Computing*, 1999.

[23] R. Lempel and S. Moran. The stochastic approach for link-structure analysis (SALSA) and the TKC effect. In *Proceedings of the Ninth International World Wide Web Conference* 2000, pages 387–401.

[24] D. S. Modha and W. S. Spangler. Clustering hypertext with applications to Web searching. *Proceedings of the ACM Hypertext 2000 Conference, San Antonio, TX*, 2000. Also appears as IBM Research Report RJ 10160 (95035), October 1999.

[25] M. S. Mizruchi, P. Mariolis, M. Schwartz, and B. Mintz. Techniques for disaggregating centrality scores in social networks. In N. B. Tuma, editor, *Sociological Methodology*, pages 26–48. Jossey-Bass, San Francisco, 1986.

[26] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank citation ranking: Bringing order to the Web. *Stanford Digital Library Technologies*, Working Paper 1999-0120, 1998.

[27] D. Rafiei, and A. Mendelzon. What is this page known for? Computing Web page reputations. In*Proceedings of the Ninth International World Wide Web Conference* 2000, pages 823–836.

[28] R. Sarukkai. Link prediction and path analysis using Markov chains. In*Proceedings of the Ninth International World Wide Web Conference* 2000, pages 377–386.

[29] H. Small. Co-citation in the scientific literature: A new measure of the relationship between two documents. *J. Amer. Soc. Info. Sci.*, 24, 1973.

What do the Neighbours Think?
Computing Web Page Reputations

Alberto O. Mendelzon
Department of Computer Science
University of Toronto
mendel@cs.toronto.edu

Davood Rafiei
Department of Computing Science
University of Alberta
drafiei@cs.ualberta.ca

Abstract

The textual content of the Web enriched with the hyperlink structure surrounding it can be a useful source of information for querying and searching. This paper presents a search process where the input is the URL of a page, and the output is a ranked set of topics on which the page has a reputation. For example, if the input is www.gamelan.com, then a possible output is "Java." We describe a simple formulation of the notion of reputation of a page on a topic, and report some experiences in the use of this formulation.

1 Introduction

The idea of exploiting the "reputation" of a Web page when searching has attracted research attention recently and even been incorporated into some search engines [16, 6, 11, 2, 3]. The idea is that pages with good reputations should be given preferential treatment when reporting the results of a search; and that link structure can be mined to extract such reputation measures, on the assumption that a link from page a to page b is, to some degree, an endorsement of the contents of b by the creator of a. The question that needs to be answered to use page reputation in Web search is: given a topic, what pages have the highest reputation on this topic? We consider a different question in this paper: given a page (or a Web site), on what topics is this page considered an authority by the Web community?

There are many potential applications for such computations. For example, a company may wish to know how its Web site is categorized by other pages that point to it, for several reasons: to assess its popularity, to check whether it is projecting the right image, or to detect problems signaled by unflattering links. Statistics of web access, such as the unique monthly visitor counts maintained by Web ranking services, are notoriously controversial [18] and do not provide insight on the topics that a Web site is perceived to be relevant to. A link-based reputation measuring service could be used not only by the site owners, but by anyone who needs to evaluate a site before using it as a source of information, or before transacting business with it.

Another example is the use of the reputation measure of a Web site listing a researcher's publications to help assess the impact of the researcher's work, with the obvious caveats against depending too heavily on any one measure of impact– caveats that, for that matter, also apply to more traditional methods such as print or online citation indexes.

Copyright 2000 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

Bulletin of the IEEE Computer Society Technical Committee on Data Engineering

There are difficulties in formalizing the notion of “reputation” effectively. The assumption that links are endorsements suggests that the number of incoming links of a page indicates its reputation. But in practice, links represent a wide variety of relationships such as navigation, subsumption, relatedness, refutation, justification, etc. In addition, we are interested not just in the overall reputation of a page, but in its reputation on specific topics.

In this paper, we describe a search process where the input is the URL of a page, and the output is a ranked set of *topics* on which the page has a reputation. (For our purposes, a topic is simply a term or a phrase, an admittedly simplistic definition.) For example, if the input is `www.informatik.uni-trier.de/~ley/db`, then among the outputs we would like to see with high rank topics such as “DBLP Bibliography”, “database systems,” etc. We present a simple formulation of the notion of reputation with the following two goals in mind: (1) it must be easy to compute in the setting of the Web; (2) it must be effective in measuring the reputations of pages. We point out that this formulation is a rough approximation to a more complex measure, based on random walk models of Web browsing behaviour. Finally, we report some test results on the effectiveness of our computations.

1.1 Related Work

Recent analyses of the linkage structure of the Web suggest that hyperlinks between pages often represent relevance [16, 6] or endorse some authority [11, 2, 3].

In a method incorporated into the Google search engine, Brin and Page [3] compute the importance of a Web page as the sum of the importance of its incoming links. The computation simulates the behaviour of a “random surfer” who either selects an outgoing link uniformly at random, or jumps to a new page chosen uniformly at random from the entire collection of pages. The PageRank of a page corresponds to the number of visits the “random surfer” makes to the page.

Kleinberg [11] proposes an algorithm that, given a topic, finds pages that are considered authorities on that topic. The algorithm, known as HITS, is based on the hypothesis that for broad topics, authority is conferred by a set of *hub pages*, which are recursively defined as a set of pages with a large number of links to many relevant authorities.

In earlier work [17], we developed two formulations of the notion of reputation based on random walk models of simple Web browsing behaviours. The first model, based on the idea of one-level influence propagation, generalizes PageRank; the main difference is that the ranking is performed with respect to a specific topic instead of computing a universal rank for each page. The second model is a probabilistic formulation of a model similar to Hubs and Authorities; this formulation allows us to invert the search process, i.e., given the URL of a page, we can find the topics the page is an authority on. For differences between HITS and probabilistic approaches, see the work by Lempel and Moran [15].

The notion of adjusting link weights for HITS was studied by Chakrabarti et al. [4] and Bharat and Henzinger [2]. Based on the hub-and-authority structure of a community, Kumar et al. [13] show that a large number of such communities can be identified from their signatures in the form of complete bipartite subgraphs of the Web. Dean and Henzinger [6] suggest algorithms to find related pages of a given page solely based on the linkage structure around the page. Finally, Henzinger et al. [9, 10] use random walks on the Web to do URL sampling and also to measure the quality of pages stored in an index.

2 Our Approach

Given a page, we want to identify a ranked list of topics the page has a reputation on. This requires a ranking function and a collection (of pages and topics) over which the ranks can be computed.

We first define two ratios that relate a page *p* and a topic *t*. The *penetration* of page *p* on topic *t*, $P_p(t)$, is the fraction of pages on topic *t* that point to page *p*. (For our purposes, a page is *on* topic *t* simply when it contains the term or phrase *t*.) The *focus* of page *p* on topic *t*, $F_t(p)$, is the fraction of pages pointing to *p* that are on topic *t*. That is, if $I(p,t)$ is the number of pages that contain *t* and point to *p*, $In(p)$ is the number of pages that point to *p*, and *N(t)* is the number of pages that contain *t*, then

$$P_p(t) = I(p,t)/N(t)$$

and

$$F_t(p) = I(p,t)/In(p).$$

Note that these quantities can be interpreted as conditional probabilities: $P_p(T)$ is the conditional probability that a page points to *p*, given that it contains *t*, and $F_t(p)$ is the conditional probability that a page contains *t*, given that it points to *p*.

Using one of these ratios as a measure of the reputation of *p* on *t* is problematic. For example, $P_p(t)$ may overestimate the reputation on any topic of pages that have large in-degrees, while $F_t(p)$ may overestimate the reputation of those whose incoming links are narrowly concentrated on topic *t*. It is more appropriate to consider, not just the conditional probability that a page points to *p* given that it contains *t*, but how much larger (or smaller) is this probability than the unconditional probability that an arbitrary page points to *p*. Note that the latter probability is given by $L(p) = In(p)/N_w$, where N_w is the number of pages on the web. Let us define the *reputation measure* of *p* on *t*, $RM(p,t)$, as

$$RM(p,t) = \ (P_p(t) - L(p))/L(p).$$

We could equally define $RM(p,t)$ in terms of $F_t(p)$, letting $M(t) = N(t)/N_w$ be the probability that an arbitrary page contains term *t*:

$$RM(p,t) = \ (F_t(p) - M(t))/M(t) = \ (P_p(t) - L(p))/L(p).$$

If we now define $LM(t,p)$ as the probability that a page contains term *t* and points to page *p*, it is easily seen that

$$RM(p,t) = \ [LM(t,p)/(L(p) \times M(t))] \ - \ 1.$$

That is, $RM(p,t)$ measures how far from independent are the events “a page contains *t*” and “a page points to *p*.” Brin *et al.* [1] propose a similar measure, called a *dependence rule*, as an alternative to association rules (which use confidence ratios similar to $P_p(t)$ and $F_t(p)$), and show how to use standard statistical tests to evaluate its significance; we do not pursue the latter topic in this paper.

Given a search engine that can compute estimates of $I(p,t)$, $N(t)$ and N_w, $RM(P,t)$ can be readily estimated as

$$RM(p,t) = \ (N_w I(p,t)/N(t) In(p)) - \mathit{1}.$$

We next show that the proposed ranking provides both an effective and easy to compute reputation measure.

2.1 Ranking Effectiveness

We now provide some justifications on the effectiveness of the proposed ranking. When a page is created, it has no incoming links (except possibly some links from the same site, which we ignore in our computation). As other users become aware of the page, based on their judgments of the content of the page and its relevance to their topics of interest, they start including links to the page within the pages they create or maintain. After a while, if a large fraction of pages on a specific topic point to the page, it is natural to expect that the page has secured a reputation on that topic.

A similar interesting phenomenon can be seen in link navigation. Users frequently search the Web by alternating between the two modes of (1) searching for pages that contain some terms (using a search engine), and (2) following outgoing links from those pages. If a large fraction of pages on a specific topic points to a page, the page will be most likely visited by the Web surfers searching for pages on that topic. Specifically, for a given page *p* and topic *t*, $RM(p,t)$ is, to some degree, a rough approximation to the number of visits made to page *p* by a "random surfer" who wanders the Web by following the links and searching for pages on topic *t* (see [17] for details).

2.2 Rank Computations

Consider the rank computation in the simple case where both a page *p* and a word or phrase *t* are given. We can use a search engine to estimate the values of $In(p)$, $I(p,t)$ and $N(t)$, and then plug these estimates in to compute $RM(p,t)$. For example, we can estimate $In(p)$, $I(p,t)$ and $N(t)$ by respectively sending queries "+link:p," "+link:p +t" and "+t" to AltaVista (*www.altavista.com*) and retrieving the counts returned by the engine. The value of N_w, a constant which doesn't affect the ordering, can be estimated by the number of pages in the search engine collection; this is often publicly announced (e.g., [14]).

However, we are often interested in the case where the topics that a page has a reputation on are not known in advance. Thus the problem is: how to compute for a given page *p* the set of topics *t* such that $RM(p,t)$ is highest? A solution is to compute $RM(p,t)$ for every word or phrase that appears in page p. Although this is easy to compute, it is not good enough because, as pointed out in the literature, often a page is an authority on some term that is not mentioned in the page. For example, the IBM Almaden Research Center (*www.almaden.ibm.com*) has a reputation on "data mining," but this phrase does not appear anywhere in its home page. [1]

Another solution is to compute $RM(p,t)$ for every possible word or phrase that appears in a page that points to *p*. (We call such pages "incoming links" of *p*.) In general, this is infeasible in the setting of the Web since there can be tens or hundreds of thousands of incoming links, and examining all those links is a cumbersome process.

We now describe the solution adopted by TOPIC [19], a prototype developed at the University of Toronto for computing Web page reputations. Given a page, the system compiles a set of incoming links of the page. This is currently done using AltaVista and Lycos, but it can be equally done using other search engines. The size of the result set is limited by the maximum number of entries returned by the search engine; search engines often return no more than 1000 entries. Clearly the accuracy of the ranks will depend on the fraction of incoming links returned. Then, for each incoming link, the system extracts words and phrases from the "snippet" returned by the search engine, rather than the page itself. This avoids the additional overhead of downloading the page, under the expectation that the snippet of a page, to some degree, is representative of the topic of the page.

There are other issues which need to be dealt with. First, links within the same site are often created for navigation purposes; as mentioned above, these links are ignored. Second, it is quite likely to find one or more near-duplicate copies of the same document in the search engine collection, even though search engines usually try to avoid storing duplicates. To address this problem, duplicate snippets are removed. Third, stop words such as "the," "for," "in," etc. and rare terms such as "BBAAA" usually convey no specific meanings (for the purpose of computing page reputations) and are removed.

3 Examples

In this section, we report our experiences with TOPIC, a prototype that uses the proposed method for reputation measurements. The input to the prototype is the URL of a page, and the output is a ranked list of topics. The default value for the number of links to download is 300, but the user can change it to a smaller number (to get better speed) or to a larger number (to get better precision). The user can enter an optional term or phrase, in which case the reputation of the page is measured only on that particular topic, and the result is displayed within a list of top 10 authorities identified by Google [8] for the same topic, providing a comparative ranking. The

[1] Disclaimer: all the examples are current as of 00/08/28, but things change fast on the Web.

default search engine for downloading the incoming links and estimating the query counts is AltaVista, but the user can change it to another engine (currently Lycos being the only alternative).

3.1 Known Authoritative Pages

For our first experiment, we selected the home pages of a set of U.S. database research groups, whose reputations we knew, from the DBLP bibliography [5]. [2] As shown in Figure 1, the results look quite reasonable. Note the high reputation of the *Microsoft Research* home page on the phrase "Data Engineering Bulletin," due to the fact that the site hosts the online version of this Bulletin.

URL : www-db.research.bell-labs.com   192 links examined (out of 193 available) **Topics**: Database Systems, Data Mining, ACM, Databases, Computer Science
URL : www.research.microsoft.com/research/db   212 links examined (out of 349 available) **Topics**: Technical Committee on Data Engineering, IEEE Data, Active Databases, Data Engineering Bulletin, Database Research, SIGMOD, SIGMOD Record, DBLP, VLDB, Database Systems
URL : www.almaden.ibm.com   999 links examined (out of 7815 available) **Topics**: IBM Almaden Research Center, Search Engines, Data Mining, Microscopy, Visualization
URL : www-db.stanford.edu   1000 links examined (out of 5637 available) **Topics**: Database research, Data Warehousing, Database Systems, Data Mining, Stanford
URL : db.cs.berkeley.edu   73 links examined (out of 130 available) **Topics**: Computer Science, Berkeley, Database, Research
URL : www.db.ucsd.edu   43 links examined (out of 78 available) **Topics**: XML, Database, Research, Project, Information

Figure 1: Selected database research group home pages and their reputations

3.2 Unregulated Web Sites

For our second experiment, we selected the home pages of a number of Canadian departments of Computer Science. The main characteristic of these sites is that they are unregulated, in the sense that users store any documents they desire in their own pages. Our goal was to find out how a site is perceived overall, without singling out specific pages stored on the site.

The results, as shown in Figure 2, can be surprising. The Computer Science Department at the University of Toronto has a high reputation on "Russian history" and "travel" mainly because a Russian graduate student of the department has put online a large collection of resources on Russia, and many pages on Russia link to it. The high reputation on "hockey" is due to a former student who used to play on the Canadian national women's hockey team. The Department of Computer Science at the University of British Columbia has a high reputation on "periodic table" because the site keeps an online version of the periodic table of chemical elements. The site also has a high reputation on "Anime" and "Manga," Japanese animation and comic art, because a staff member has put online a collection of pages on the subject and many other pages link to this collection. The reputation

[2] Disclaimer: the selection is arbitrary and not intended to be a complete list of groups of high reputation.

of the Department of Computing Science at the University of Alberta on “virtual reality” and “chess” and the reputation of the Department of Computing Science at Simon Fraser University on “data mining” and “reasoning” are to be expected. The high reputation of the CS Department at Simon Fraser University on “whales” is due to a 3D animation project being carried out on whales.

The results reported here, even though some are surprising, turn out to be consistent with other data once we examine the pages in question. For example, the Russian page at the University of Toronto reported over one million hits since 1997 for a period of two years; the number of unique visits (visits with different host names) to the page was slightly over 333,000. Similarly, Andria Hunter’s hockey page, for the period of two weeks starting from July 19, 2000 showed 2,700 average daily visits. The total number of visits since the creation of the page in March 1995 was about 1.8 million.

URL : www.cs.toronto.edu 1000 links examined (out of 8400 available) **Topics**: Russian History, Neural, Travel, Hockey
URL : www.cs.ualberta.ca 1000 links examined (out of 10557 available) **Topics**: University of Alberta, Virtual Reality, Language, Chess, Artificial
URL : www.cs.ubc.ca 999 links examined (out of 17958 available) **Topics**: Confocal, Periodic Table, Anime, Computer Science, Manga, Mathematics
URL : www.cs.sfu.ca 963 links examined (out of 2055 available) **Topics**: Whales, Simon Fraser University, Data Mining, Reasoning

Figure 2: Selected Computer Science Department home pages and their reputations

3.3 News Sites

So far, we have only ranked topics for a given site. For this purpose it does not matter whether we use $RM(p, t)$ or the penetration $P_p(t)$, since they produce the same ordering of topics for a fixed p. In this experiment, we evaluated a set of sites, all of them news providers, on a predetermined set of topics. For a fixed topic t, ranking a set of sites by their value of $RM(p, t)$ amounts to ordering them by their focus $F_t(p)$. Since we are interested in comparing both within and across sites, we show in Figure 3, both the penetration and focus for each combination of site and topic, revealing interesting patterns. For example, *CNN* has the largest penetration of any site on every topic, but relatively low focus, showing that it is well-known on all the topics but not specifically known for any single one. On the other hand, *wired.com*, while ranking a close second in penetration to *CNN* on “technology,” has substantially higher focus on this topic than any other site.

3.4 Personal Home Pages

In another experiment, we selected a set of personal home pages and used our system to find the reputation topics for each page. We expected this to describe in some way the reputation of the owner of the page. The results, as shown in Figure 4, can be revealing, but need to be interpreted with some care. Tim Berners-Lee’s reputation on the “History of the Internet,” Don Knuth’s fame on “TeX” and “Latex,” Jeff Ullman’s reputation on “database systems” and “programming languages” and Jim Gray’s reputation on “database,” “research” and “systems” are to be expected. The Comprehensive TeX Archive Network (CTAN) is frequently cocited with Don Knuth’s home page mainly within TeX information pages. Alberto Mendelzon’s high reputation on “data warehousing” and

	CNN	BBC	ABC	Wired.com
	(www.cnn.com)	(www.bbc.co.uk)	(www.abcnews.go.com)	
International News	[0.0277 , 0.0170]	[0.0121 , 0.0201]	[0.0021 , 0.0279]	[0.0057 , 0.0090]
Weather	[0.0096 , 0.2228]	[0.0029 , 0.1845]	[0.0005 , 0.2338]	[0.0013 , 0.0744]
Sports	[0.0043 , 0.1724]	[0.0012 , 0.1265]	[0.0003 , 0.2531]	[0.0017 , 0.1698]
Entertainment	[0.0074 , 0.1632]	[0.0049 , 0.2913]	[0.0003 , 0.1516]	[0.0023 , 0.1294]
Travel	[0.0040 , 0.1421]	[0.0016 , 0.1548]	[0.0003 , 0.2264]	[0.0012 , 0.1082]
Technology	[0.0041 , 0.1957]	[0.0011 , 0.1390]	[0.0002 , 0.2234]	[0.0038 , 0.4560]
Business	[0.0034 , 0.2831]	[0.0018 , 0.3946]	[0.0002 , 0.3399]	[0.0022 , 0.4635]

Figure 3: [P , F] ranks of news provider sites on a set of topics

“OLAP,” on the other hand, is due to an online research bibliography he maintains on these topics, and not to any merits of his own.

URL : www.w3.org/People/Berners-Lee   789 links examined (out of 1334 available)

Topics: Tim Berners-Lee, History Of The Internet, Hypertext, Pioneers, Brief, W3C

URL : www-cs-faculty.stanford.edu/˜ knuth   1000 links examined (out of 1543 available)

Topics: Don Knuth, TeX Users, LaTex, Linux, CTAN, Donald, Computer Science

URL : www-db.stanford.edu/˜ ullman   294 links examined (out of 423 available)

Topics: Ullman, Database Management Systems, Database Systems, Database Design, Data Mining, Programming Languages

URL : www.research.microsoft.com/˜ gray   57 links examined (out of 74 available)

Topics: Database, Research, Systems, Information

URL : www.cs.toronto.edu/˜ mendel   161 links examined (out of 162 available)

Topics: Data Warehousing, OLAP, Data Mining, Bibliography, Computer Science, Database, Research

Figure 4: Selected personal home pages and their reputations

4 Conclusion

We have presented a method for evaluating the reputation of a page which is both easy to compute and, according to our preliminary tests, effective. There are some limitations to our method that should be mentioned. First, we don’t exploit relationships among terms such as synonyms, generalization, specialization, etc. Second, our computation is affected by the fraction of incoming links returned by search engines and the unpredictable order in which they are returned: this affects both the choice of relevant topics and the estimation of the ratios. We are currently working on refinements of the method to overcome these limitations, as well as on more systematic evaluation of the method’s effectiveness.

Acknowledgements

This research was supported by the Natural Sciences and Engineering Research Council Canada and Communications Information Technology Ontario.

References

[1] S. Brin, R. Motwani, and C. Silverstein, Beyond market baskets: generalizing association rules to dependence rules. *Data Mining and Knowledge Discovery* 2(1), pages 39–68, 1998.

[2] K. Bharat and M. R. Henzinger. Improved algorithms for topic distillation in hyperlinked environments. In *Proc. of the ACM SIGIR Conference*, pages 104–111, 1998.

[3] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In *Proc. of the WWW7 Conference*, pages 107–117, Brisbane, Australia, April 1998. Elsevier Science.

[4] S. Chakrabarti, B. Dom, P. Raghavan, S. Rajagopalan, D. Gibson, and J. Kleinberg. Automatic resource compilation by analyzing hyperlink structure and associated text. In *Proc. of the WWW7 Conference*, Brisbane, Australia, April 1998. Elsevier Science.

[5] DBLP Bibliography. www.informatik.uni-trier.de/˜ley/db.

[6] J. Dean and M. R. Henzinger. Finding related pages on the Web. In *Proc. of the WWW8 Conference*, pages 389–401, Toronto, Canada, May 1999. Elsevier Science.

[7] D. Florescu, A. Levy, and A. Mendelzon. Database techniques for the World Wide Web : a survey. *ACM SIGMOD Record*, 27(3):59–74, September 1998.

[8] Google. www.google.com.

[9] M. R. Henzinger, A. Heydon, M. Mitzenmacher, and M. Najork. Measuring index quality using random walks on the Web. In *Proc. of the WWW8 Conference*, pages 213–225, Toronto, Canada, May 1999. Elsevier Science.

[10] M. R. Henzinger, A. Heydon, M. Mitzenmacher, and M. Najork. On near-uniform url sampling. In *Proc. of the WWW9 Conference*, pages 295–308, Amsterdam, May 2000. Elsevier Science.

[11] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. In *Proc. of ACM-SIAM Symposium on Discrete Algorithms*, pages 668–677, January 1998.

[12] R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. Extracting large-scale knowledge bases from the Web. In *Proc. of the VLDB Conference*, pages 639–650, September 1999.

[13] R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. Trawling the Web for emerging cyber-communities. In *Proc. of the WWW8 Conference*, pages 403–415, Toronto, May 1999. Elsevier Science.

[14] S. Lawrence and C.L. Giles. Accessibility of information on the Web. *Nature*, 400:107–109, 1999.

[15] R. Lempel and S. Moran. The stochastic approach for link-structure analysis (SALSA) and the TKC effect. In *Proc. of the WWW9 Conference*, pages 387–401, Amsterdam, May 2000. Elsevier Science.

[16] Netscape Communications Corporation. What's related. Web page, www.netscape.com/escapes/related/faq.html.

[17] D. Rafiei and A. O. Mendelzon. What is this page known for? Computing Web page reputations. In *Proc. of the WWW9 Conference*, pages 823–835, Amsterdam, May 2000. Elsevier Science.

[18] Jon Swartz. Net rankings vex dot-coms. *USA Today*, June 2000. www.usatoday.com/life/cyber/invest/in794.htm.

[19] TOPIC. www.cs.toronto.edu/db/topic.

Learning to Understand the Web

William Cohen[‡] wcohen@whizbang.com

Andrew McCallum[§] mccallum@whizbang.com

Dallan Quass[¶] dquass@whizbang.com

WhizBang! Labs–Research

Abstract

In a traditional information retrieval system, it is assumed that queries can be posed about any topic. In reality, a large fraction of web queries are posed about a relatively small number of topics, like products, entertainment, current events, and so on. One way of exploiting this sort of regularity in web search is to build, from the information found on the web, comprehensive databases about specific topics. An appropriate interface to such a database can support complex structured queries which are impossible to answer with traditional topic-independent query methods. Here we discuss three case studies for this "data-centric" approach to web search. A common theme in this discussion is the need for very robust methods for finding relevant information, extracting data from pages, and integrating information taken from multiple sources, and the importance of statistical learning methods as a tool for creating such robust methods.

1 Introduction

In a traditional information retrieval system, it is assumed that queries can be posed about any topic. In reality, a large fraction of web queries are posed about a relatively small number of topics, like products, entertainment, current events, and so on. One way of exploiting this sort of regularity in web search is to build, from the information found on the web, comprehensive databases about specific topics. An appropriate interface to such a database can support complex structured queries (*e.g.*, "bed and breakfast units within one mile of the beach in Hawaii costing less than $150 per night") which are impossible to answer with a topic-independent query method.

In this paper, we will motivate this "data-centric" approach to web search, and then discuss three case studies: namely WHIRL, CORA, and FLIPDOG. In a typical WHIRL application, data from a few dozen web sites is merged together to form a database; however, the extracted data is generally "dirty," or approximate, in several key respects. WHIRL uses textual similarity metrics from information retrieval to approximately answer structured queries to this "approximate" database of web information. CORA and FLIPDOG extract information about specific types of entities (research papers and job postings, respectively) from thousands of different web sites.

Copyright 2000 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

Bulletin of the IEEE Computer Society Technical Committee on Data Engineering

[‡] Mailing address: WhizBang! Labs, East, 4616 Henry Street, Pittsburgh PA 15213

[§] Mailing address: WhizBang! Labs, East, 4616 Henry Street, Pittsburgh PA 15213

[¶] Mailing address: WhizBang! Labs, 3210 N. Canyon Road, Suite 300, Provo Utah 84604

Both CORA and FLIPDOG rely heavily on machine learning methods to discover relevant information sources, extract information about entities, and organize entities into categories. A common theme in our discussion of these systems is the need for very robust methods for finding and classifying relevant information, extracting data from pages, and integrating information taken from multiple sources, and the importance of statistical learning methods as a tool for creating such robust methods.

2 Data-Centric Web Search

Traditional web search methods have certain fundamental limitations. Topic-independent search methods rely heavily on matching terms in a user's natural language query with terms appearing in a document. This approach is broadly applicable, but complex information needs (*e.g.*, the sample query of the introduction) cannot be expressed easily in a topic-independent system. Another limitation is that returning a single list of documents, ranked by estimated relevance, may not satisfy a user's query. For instance, the answer to the query "find technical job postings in companies whose stock has increased 50% or more in the last two years" may require combining information from two different sources—one which contains information about stock prices, and one which contains job postings.

We believe that such complex information needs are real (even if they are not often formulated by users of existing search engines). One approach to handling these queries is to compile the textual information on the web into some formalism that supports structured queries—for example, a relational database. In this "data-centric" approach to web information access, one begins by focusing on a particular topic: this is an important first step, as this makes it possible to build a database with deep, semantically meaningful schema. One then constructs a "topic-specific search engine": an information access system that allows access to all the information on the web that is relevant to a particular topic. We will assume here that a "topic-specific search engine" includes some sort of facility for answering structured queries, such as a relational database interface, but other organizational schemes (like a Yahoo-style topic hierarchy for documents) may also be used.

For example, suppose that a topic-specific search engine for travel were to be constructed. If I were interested in vacationing in Hawaii, I might browse the topic hierarchy to "Lodging," at which point I could either continue to drill down into the topic hierarchy, or I could narrow my search more quickly through a relational-style query over the information in lodging. Suppose that "type of lodging," "location," and "room rate" had been extracted from each lodging document and stored in a relational database. A search form could be created at the lodging node of the topic hierarchy that would allow me to query over these fields. I could specify the criteria "bed and breakfast units within one mile of the beach in Hawaii costing less than $150 per night." The result would be a list of bed and breakfast units that matched my search criteria, including their names, addresses, and phone numbers (also extracted from the documents), along with links to the actual source documents.

3 Building Topic-Specific Search Engines

While a topic-specific search engine is conceptually quite simple, building one is not. The essential problem involves classifying web pages relevant to the topic and extracting structured information that can be stored in a database. Web information on a topic may be generated by thousands of different content providers in thousands of different formats. This information is continually changing and growing, and the scale of the web is such that only an automatic or nearly-automatic approach can hope to maintain a database for a reasonably broad topic. Furthermore, most of the web data available is presented as simple text—a format easy for people to generate and understand, but extremely difficult for computers to process. These factors make it extremely difficult to automatically convert web information into a format suitable for storing in a single coherent database.

One possible solution to this problem is to require that providers generate information in some database-compatible format, such as XML. Unfortunately, while XML is a great way to share information coming out of databases, XML is far more difficult for end users to generate than simple textual documents. We cannot expect end-users to consistently tag their documents with the relevant pieces of information and to conform to standard DTDs. Nor is it reasonable to expect that programmers will mount a large-scale effort to convert the world's existing text to XML data. Therefore, although we believe that XML will be useful for exchanging information, we do *not* believe that most information will be created originally in XML. For the foreseeable future, web information will be presented primarily as text.

We believe that the most feasible approach for collecting web information into topic-specific search systems is to automatically extract information from the web using learned procedures. For example, one type of learning system might allow users to create topic hierarchies "by example." A user would create a topic hierarchy and place a few example documents into that hierarchy. The system would then gather additional documents from the web similar to these examples, and present them to the user for verification, essentially asking "is this the kind of thing you are interested in?" After such a dialog, the system would learn to automatically classify documents into the hierarchy. Analogously, a user could show examples of the relational-database fields that should be extracted from a document. After training, the system would learn to extract these fields automatically and populate a topic-specific relational database.

Although it is not necessary for such a tool to incorporate learning or programming-by-example, there are several reasons for believing that learning is the "right" approach to this problem. One advantage is that no programming is necessary—all the user need do is provide and verify examples. This greatly lowers the cost of developing a topic-specific search engine. Another advantage is that learned classifiers and extractors are often more robust than hand-written rules, in the sense that they are less likely to break down on unusual inputs.

Below we will present several examples of data-centric web search systems, and discuss the ways in which learning and other robust, statistical methods are used. First, however, we will give some additional background on the main techniques used in a topic-specific search engine: focused crawling, text classification, and information extraction.

4 Background and Related Work

Research in the areas of focused crawling, text classification, and information extraction has been on-going for many years. Early research methods suffered from low accuracy or required a significant amount of hand-tuning. Only recently have methods been developed that achieve sufficient accuracy without extensive hand-tuning as to make topic-specific search engines feasible.

As the Web continues to grow exponentially, the idea of crawling the entire Web on a regular basis becomes less and less feasible. Since only a relatively small portion of the Web is relevant to a topic-specific search engine, it can use the notion of focused crawling (*e.g.,* [4, 3, 6, 19]) to find those pages quickly, and bypass most pages that are not related to the topic. Crawling the web in general involves gathering up the links on each visited page and putting them into a queue of links to be followed. Focused crawling reorders the links in the queue as to their predicted likelihood to lead to pages that are relevant to a particular topic. By following high-likelihood links first, pages that are relevant to a particular topic are found more quickly. Furthermore, links leading to irrelevant pages tend to fall to the bottom of the queue, and can be ignored once the crawler has determined that the rate of finding new relevant pages from following links has fallen below some given threshold.

Text classification is used in topic-specific search engines in at least two areas. First, it is used during crawling to classify web pages as to whether they are relevant to the given topic. Second, it can be used to classify relevant web pages or content extracted from web pages into a Yahoo-style topic hierarchy. Much research has been done in the area of text classification (*e.g.,* [7, 14]). The general idea is to first convert the text to a multidimensional vector representation, where dimensions correspond to words in the text, then to use machine-learning or statistical techniques (*e.g.,* decision trees, naive Bayes) to classify the vectors in the multidimensional space. The hypertext structure of the Web provides an additional advantage for web page classification that is not available for classifying text in general: One can use the text in links leading to the page and also the classification of nearby pages to make classification of a particular web page more accurate than if classification were performed on the text of the page alone.

The key to creating a topic-specific search engine is to extract values for specific fields from the web pages, storing the values in a database so that structured queries can be performed over the extracted information. A primary venue for research in information extraction has been the Message Understanding Conference (MUC) series. The MUC conferences were a series of contests where researchers would build working systems based upon their research and test their systems on real-world tasks. One set of tests involved extracting "named entities" such as locations, company names, or person names from flat text files such as newspaper articles. Early systems were typically built using a large set of hand-coded extraction rules. Later, some of the systems were built using machine-learning techniques (*e.g.*, [18]).

In comparison to extracting information from flat text files, it is possible to get increased accuracy when extracting information from web page by taking advantage of the HTML tag structure. One common approach to extracting information from web pages is to write site-specific "wrappers" that extract information based upon regularities of the HTML tag structure that is typically present in a single web site (*e.g.*, [10, 12]). For example, if a bed and breakfast web site listed all their bed and breakfast units in a list, it would be possible to extract information for each of those units by simply extracting the text following each "`<LI>`" tag. The disadvantage of this approach is that as web sites change over time, the tags the wrappers are dependent upon tend to change as well, requiring the wrappers to be rewritten.

Another method for extracting information from the Web is to use a bootstrapping approach (*e.g.*, [1, 2]). In this approach one begins with a small relation and searches the Web for additional tuples to add to the relation. New tuples are added to the relation using a bootstrapping technique, where new tuples are added if they appear on web pages in the same contexts as existing tuples in the relation.

5 Three Case Studies

Below we review two research prototypes and a commercial system that use the methods discussed above as well as additional research methods to create topic-specific search engines.

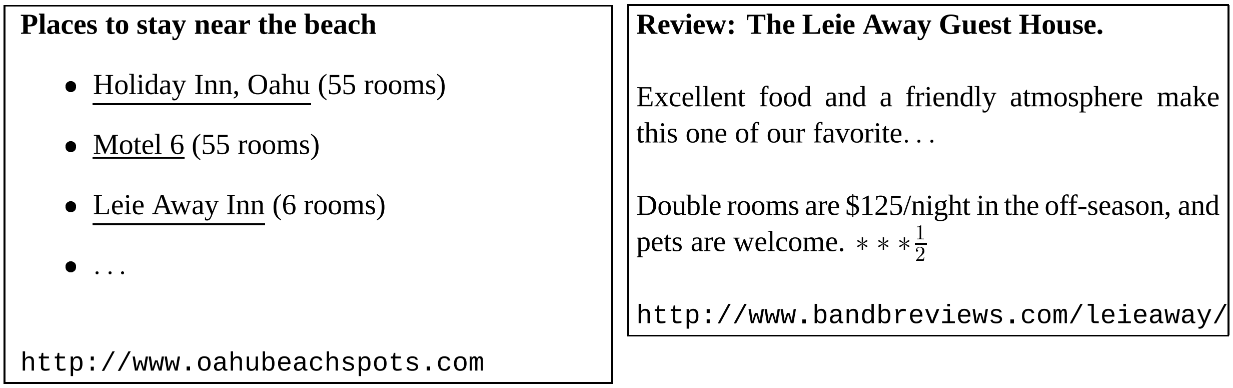
5.1 A Research Paper Search Engine: CORA

The CORA system [16, 15] automatically spiders, classifies and extracts computer science research papers from the Web. The papers in CORA are organized into a taxonomy with 75 leaves, and various fields such as author and title are extracted from each paper. Additionally, bibliographic information is extracted from each paper, allowing bibliometric analysis to be performed. It currently contains over 50,000 papers and is publically available at *www.cora.whizbang.com*.

The creation and maintenance of CORA relies heavily on artificial intelligence and machine learning techniques. The tasks can be broken down into four components: spidering, extraction, reference matching and classification.

Research papers are gathered from the web using an efficient, topic-directed spider based on reinforcement-learning. The spider uses the words in the context of a hyperlink to assign it an "expected future discounted reward," and follows the high-reward links with higher priority. Training data for the language model that makes the assignment is easily obtained by exhaustively spidering a few training sites, and counting the minimum number of hops from each hyperlink to a target page. Experiments show that our directed spider is three times more efficient than a spider based on breadth-first search, and also more efficient than other smart spiders that do not explicitly model future reward.

After spidering, the papers' titles, authors, and references are automatically extracted using hidden Markov models—a type of probabilistic finite state machine often used in speech recognition. Certain states are associated



Places to stay near the beach

- Holiday Inn, Oahu (55 rooms)
- Motel 6 (55 rooms)
- Leie Away Inn (6 rooms)
- ...

http://www.oahubeachspots.com

Review: The Leie Away Guest House.

Excellent food and a friendly atmosphere make this one of our favorite...

Double rooms are $125/night in the off-season, and pets are welcome. $* * *\frac{1}{2}$

http://www.bandbreviews.com/leieaway/

Figure 1: Typical travel-oriented web data

with different database fields (such as *author* or *title*), and after calculating the most likely state path using the Viterbi algorithm, the word emissions associated with those particular states are said to be associated with their associated fields. Training data consists of a combination of hand-labeled data and data from large bibliography files found on the web. After learning the state-transition structure and the parameters from this training data, automatic extraction achieves over 90% accuracy.

Next, the extracted references are matched against the papers, so that the complete citation graph is built. This is done efficiently by a two-stage clustering process that uses two different distance metrics—first a cheap, approximate distance metric based on an inverted index, then an expensive, detailed distance metric based on a tuned string-edit distance [17]. The quadratic-time string-edit distance is only performed on the small subset of reference-pairs that are within some distance threshold according to the cheap distance metric. Once the citation graph is complete, we run a bibliometric analysis based on principal components analysis to automatically find "seminal" and "survey" papers.

Finally, the papers are automatically categorized into the topic hierarchy using probabilistic text classification. A small number of human-provided keywords, a large amount of unlabeled data, and a statistical technique for taking advantage of the hierarchy called "empirical Bayes" are all combined in a Bayesian classifier that provides near-human accuracy.

The CORA system can be adapted to a new domain by labeling the additional examples needed to learn a new set of classifiers—in fact, a second technical-paper search engine for statistics research papers was created in this way after just two days work. NEC's CiteSeer system [9] is another topic-specific search engine that is more specifically geared to research papers, but uses less machine learning.

5.2 Information Integration Systems and WHIRL

One set of data-centric web systems are "information integration systems" (*e.g.*, [13, 8, 11]), which collect into a single database the information from several heterogeneous, database-like web sites. As an example, consider the sample query above, and consider the web pages shown in Figure 1 (the second web page is one example of several reviews stored on the same site). A typical information integration system might extract information from these pages, converting the first web page into a relation of the form `nearBeach(hotel)` and the second web site into a relation `perNight(hotel,cost)`. The sample query above might be answered by joining these two relations and then filtering the result by cost.

One of the problems in doing information integration is that many database operations are very sensitive to errors in extracted data. As an example, if automatic, learned methods are used to extract data from the web pages above, it is quite possible that the `nearBeach` relation will contain the tuple ⟨`''Leie Away Inn''`⟩ and the `perNight` relation will contain the tuple ⟨`''Leie Away Guest House'',$125`⟩. (It is even possible that the `perNight` relation will contain a less-useful tuple, such as ⟨`''Review: The Leie Away Guest House''`,`$125`⟩.) This inconsistency means that neither version of this name will appear in the join of two relations.

The WHIRL integration system [5] is based on a novel representation scheme, which allows more extracted information to be stored as raw text. By making use of robust similarity metrics for text developed in the information retrieval community, many database-style operations can be approximated on this representation, even if the data is somewhat misaligned.

For instance, one might use the following WHIRL query to find inexpensive housing near the beach:

```
SELECT nearBeach.∗, perNight.∗
FROM   nearBeach, perNight
WHERE  nearBeach.hotel SIM perNight.hotel AND
       perNight.cost ≤ 150
```

The output of this query will be k tuples from the cross-product of `perNight` and `nearBeach` that have the most similar `hotel` fields, subject to the constraint that `perNight.cost` is less than $150; thus the join of the relations `nearBeach` and `perNight` is approximated by finding tuples with similar hotel names. The number of tuples k returned by the query is fixed by the user, and similar approximations can be used for more complex queries.[1]

WHIRL views the process of finding the k best answers to a query as an optimization problem. For this query, WHIRL searches through the space of possible pairings of `nearBeach` and `perNight` tuples to find the pairings that maximize the given similarity condition ("`nearBeach.hotel SIM perNight.hotel`") using an artificial-intelligence optimization method called A$*$ search. To make this search efficient, WHIRL relies heavily on indexing methods used in information retrieval. In information retrieval, the similarity of two documents is a function of the set of *terms* those documents share, and the weight assigned to these shared terms. (For the purpose of this discussion, a term can be considered to be a single word.) For this query, WHIRL might use inverted indices to find `perNight` tuples that are guaranteed to share some "important" (highly-weighted) term in the `hotel` field of a candidate `nearBeach` tuple. More generally, appropriate use of inverted indices ensures that the most plausible pairings are explored early in the search.

Using robust versions of database operations eliminates the need for certain data-cleaning operations—in this example, for instance, it is not necessary to normalize the names of hotels. This often greatly simplifies the process of extracting data. Experimentally, WHIRL was used to provide interfaces to 50 different sites with approximately four man-months development time—a vast improvement over earlier systems—by relying on simple, approximate extraction schemes.

The sites included in the experiments with WHIRL were primarily on two topics—birds of North America, and educational computer games for children—and were primarily sites containing large amounts of data-like material. One site for the North American bird topic was `http://www2.math.sunysb.edu/~tony/birds` which contains hundreds of sound files containing bird calls; one site for the computer game domain was `http://www.kidsdomain.com` which contains hundreds of reviews of recent computer games. For each such site, a separate extraction routine was hand-written which spidered the site, retrieved the appropriate data, and converted it into WHIRL's internal format.

Because of WHIRL's robustness, many of these extraction routines could be quite simple (most were a single line in a special-purpose language). However, extracting data was still a bottleneck for WHIRL, because a routine had to be manually written and maintained for each site. We conjecture that coupling a WHIRL-like database system with CORA-style learned extraction methods would further reduce the cost of data collection.

[1] The current implementation of WHIRL supports a fairly large subset of SQL: specifically, any disjunction of conjunctive SQL queries has an analog in WHIRL.

5.3 A Commercial System: FLIPDOG.COM

Our organization, WhizBang! Labs, has developed a set of tools that facilitate the creation of topic-specific search engines. Recently, we have fielded a commercial system using these tools: FLIPDOG.COM, an on-line job board based on a database constructed by automatically extracting job postings directly from corporate Web pages. The FLIPDOG database was created primarily by applying general-purpose machine learning techniques—techniques that could well be used to extract other sorts of databases from the Web.

FLIPDOG contains more than 600,000 jobs gathered from over 50,000 different corporate web sites, making it the largest commercial job board on the Web. Operationally, the process of constructing the FLIPDOG database is much like the process used in CORA. Sites are automatically spidered to find pages that contain job postings. Individual job postings are then extracted from these pages, and augmented with automatically-extracted fields such as job title, job description, location, and application email address. Job postings are also organized into a taxonomy to facilitate browsing.

However, construction of the FLIPDOG database also required addressing a number of issues not solved by previous research prototypes. Unlike research papers, job postings are frequently accessible only by accessing forms, which means that the job-posting spider must be able to understand how to fill in these forms. The "location" field for a job posting is also harder to extract than, say, the title of a paper, because a job's location is often *not* explicitly mentioned in the job posting itself: *e.g.*, frequently a single Web page will name a location, and then list several jobs at that location.

FLIPDOG also embodies a new solution to the problem of errors in automatically-extracted data. In FLIPDOG, all automatically-made decisions about a job posting are associated with a "confidence"—a numeric measure of the system's certainty that the decision is correct—and human beings verify any low-confidence decisions. This means that the learning algorithms must not only provide accurate predictions, but accurate assessments of confidence. Adopting this approach means that FLIPDOG's job database is quite "clean," containing very few erroneous entries, but can still be refreshed on a weekly basis.

6 Summary

In general, constructing a topic-specific search engine requires solving many different problems, including identifying relevant information sources, extracting and classifying information, and integrating information taken from different sources. However, solutions to many of these problems are in hand, and have been implemented in various research prototypes and commercial systems.

We believe in the near future, toolkits consisting of various machine-learning-based techniques will make it much easier to classify and extract information from text. As the underlying technology matures, data-centric, topic-specific search engines will become more prevalent.

It is likely that many such search engines will be developed, each specializing in a different topic. Topic-specific engines are also likely to vary in their depth of coverage, with some systems electing to impose a rich schema on a smaller subset of the web, and others imposing a weak schema on a large subset of the web. Ultimately the vast majority of queries that focus on common topics will be answered by one of a few dozen general-purpose databases; and of the remaining, special-purpose queries, most will be answered by one of a few thousand more specialized databases, much like CORA. The information systems in wide use today will persist: however, traditional broad-coverage keyword-search based IR systems (like GOOGLE and ALTAVISTA) and highly-focused topic specific databases (like `imdb.com`, which contains only information about movies and TV) will simply be two ends of densely-populated spectrum of data-centric information systems, each providing a database-like view of a different part of the web.

References

[1] Eugene Agichtein and Luis Gravano. *Snowball:* extracting relations from large plain-text collections. In *Proceedings of the 5th ACM International Conference on Digital Libraries*, June 2000.

[2] Sergey Brin. Extracting patterns and relations from the World Wide Web. In *The World Wide Web and Databases, International Workshop Web'98*, Valencia, Spain, March 1998.

[3] Soumen Chakrabarti, Martin van den Berg, and Byron Dom. Focused crawling: A new approach to topic-specific web resource discovery. In *Proceedings of The Eighth International World Wide Web Conference (WWW-99)*, Toronto, 1999.

[4] Junghoo Cho, Hector Garcia-Molina, and Lawrence Page. Efficient crawling through URL ordering. In *Proceedings of the 7th World Wide Web Conference (WWW7)*, Brisbane, Australia, April 1998.

[5] William W. Cohen. WHIRL: A word-based information representation language. *Artificial Intelligence*, 118:163–196, 2000.

[6] M. Diligenti, F. M. Coetzee, S. Lawrence, C. L. Giles, and M. Gori. Focused crawling using context graphs. In *Proceedings of the 26th International Conference on Very Large Databases (VLDB-2000)*, Cairo, Egypt, September 2000.

[7] Susan Dumais, John Platt, David Heckerman, and Mehran Sahami. Inductive learning algorithms and representations for text categorization. In *Proceedings of the ACM Conference on Information and Knowledge Management (CIKM)*, November 1998.

[8] H. Garcia-Molina, Y. Papakonstantinou, D. Quass, A. Rajaraman, Y. Sagiv, J. Ullman, and J. Widom. The TSIMMIS approach to mediation: Data models and languages (extended abstract). In *Next Generation Information Technologies and Systems (NGITS-95)*, Naharia, Israel, November 1995.

[9] Lee Giles, Kurt Bollaker, and Steve Lawrence. CiteSeer: An automatic citation indexing system. In *Proceedings of the Third ACM Conference on Digital Libraries*, 1998.

[10] Ashish Gupta, Venky Harinarayan, and Anand Rajaraman. Virtual database technology. *SIGMOD Record*, 26(4):57—61, 1997.

[11] Craig A. Knoblock, Steven Minton, Jose Luis Ambite, Naveen Ashish, Pragnesh Jay Modi, Ion Muslea, Andrew G. Philpot, and Sheila Tejada. Modeling web sources for information integration. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI-98)*, Madison, WI, 1998.

[12] Nicholas Kushmerick, Daniel S. Weld, and Robert Doorenbos. Wrapper induction for information extraction. In *Proceedings of the 15th International Joint Conference on Artificial Intelligence*, Osaka, Japan, 1997.

[13] Alon Y. Levy and Rachel Pottinger. A scalable algorithm for answering queries using views. In *Proceedings of the 26th International Conference on Very Large Databases (VLDB-2000)*, Cairo, Egypt, September 2000.

[14] Andrew McCallum and Kamal Nigam. A basic introduction to the two flavors of naive Bayes document classification. In *AAAI Workshop on Learning for Text Categorization*, 1998.

[15] Andrew McCallum, Kamal Nigam, Jason Rennie, and Kristie Seymour. Cora Research Paper Search. At `http://www.cora.whizbang.com`.

[16] Andrew McCallum, Kamal Nigam, Jason Rennie, and Kristie Seymour. Automating the construction of internet portals. To appear in *Information Retrieval*, 2000.

[17] Andrew McCallum, Kamal Nigam, and Lyle Ungar. Efficient clustering of high-dimensional data sets with application to reference matching. In *KDD-2000: Proceedings of the Sixth International Conference on Knowledge Discovery and Data Mining*, 2000.

[18] Scott Miller, Michael Crystal, Heidi Fox, Lance Ramshaw, Richard Schwartz, Rebecca Stone, Ralph Weischedel, and the Annotation Group. Algorithms that learn to extract information BBN: description of the sift system as used for MUC-7. In *Proceedings of the 7th Message Understanding Conference (MUC-7)*, April 1998.

[19] Jason Rennie and Andrew McCallum. Using reinforcement learning to spider the web efficiently. In *Proceedings of the International Conference on Machine Learning (ICML)*, 1999.

Context in Web Search

Steve Lawrence
NEC Research Institute
Princeton, New Jersey
http://www.neci.nec.com/˜lawrence
lawrence@research.nj.nec.com

Abstract

Web search engines generally treat search requests in isolation. The results for a given query are identical, independent of the user, or the context in which the user made the request. Next-generation search engines will make increasing use of context information, either by using explicit or implicit context information from users, or by implementing additional functionality within restricted contexts. Greater use of context in web search may help increase competition and diversity on the web.

1 Introduction

As the web becomes more pervasive, it increasingly represents all areas of society. Information on the web is authored and organized by millions of different people, each with different backgrounds, knowledge, and expectations. In contrast to the databases used in traditional information retrieval systems, the web is far more diverse in terms of content and structure.

Current web search engines are similar in operation to traditional information retrieval systems [57] – they create an index of words within documents, and return a ranked list of documents in response to user queries. Web search engines are good at returning long lists of *relevant* documents for many user queries, and new methods are improving the ranking of search results [8, 10, 21, 36, 41]. However, few of the results returned by a search engine may be *valuable* to a user [6, 50]. Which documents are valuable depends on the context of the query – for example, the education, interests, and previous experience of a user, along with information about the current request. Is the user looking for a company that sells a given product, or technical details about the product? Is the user looking for a site they previously found, or new sites?

Search engines such as Google and FAST are making more information easily accessible than ever before and are widely used on the web. A GVU study showed that about 85% of people use search engines to locate information [31], and many search engines consistently rank among the top sites accessed on the web [48]. However, the major web search engines have significant limitations – they are often out-of-date, they only index a fraction of the publicly indexable web, they do not index documents with authentication requirements and many documents behind search forms, and they do not index sites equally [42, 43]. As more of the population goes online, and as more tasks are performed on the web, the need for better search services is becoming increasingly important.

Copyright 2000 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.
Bulletin of the IEEE Computer Society Technical Committee on Data Engineering

2 Understanding the Context of Search Requests

Web search engines generally treat search requests in isolation. The results for a given query are identical, independent of the user, or the context in which the user made the request. Context information may be provided by the user in the form of keywords added to a query, for example a user looking for the homepage of an individual might add keywords such as “home” or “homepage” to the query. However, providing context in this form is difficult and limited. One way to add well-defined context information to a search request is for the search engine to specifically request such information.

2.1 Adding Explicit Context Information

The Inquirus 2 project at NEC Research Institute [29, 30] requests context information, currently in the form of a category of information desired. In addition to providing a keyword query, users choose a category such as “personal homepages”, “research papers”, or “general introductory information”. Inquirus 2 is a metasearch engine that operates as a layer above regular search engines. Inquirus 2 takes a query plus context information, and attempts to use the context information to find relevant documents via regular web search engines. The context information is used to select the search engines to send queries to, to modify queries, and to select the ordering policy.

For example, a query for research papers about “machine learning” might send multiple queries to search engines. One of these queries might be transformed with the addition of keywords that improve precision for finding research papers (e.g., “abstract” and “references”). Another query might be identical to the original query, in case the transformations are not successful. Inquirus 2 has proven to be highly effective at improving the precision of search results within given categories. Recent research related to Inquirus 2 includes learning methods that automatically learn query modifications [18, 28].

2.2 Automatically Inferring Context Information

Inquirus 2 can greatly improve search precision, but requires the user to explicitly enter context information. What if search context could be automatically inferred? This is the goal of the Watson project [11, 12, 13]. Watson attempts to model the context of user information needs based on the content of documents being edited in Microsoft Word, or viewed in Internet Explorer. The documents that users are editing or browsing are analyzed with a heuristic term weighting algorithm, which aims to identify words that are indicative of the content of the documents. Information such as font size is also used to weight words. If a user enters an explicit query, Watson modifies the query based on the content of the documents being edited or viewed, and forwards the modified query to web search engines, thus automatically adding context information to the web search.

In addition to allowing explicit queries, Watson also operates in the background, continually looking for documents on the web related to documents that users are editing or viewing. This mode of operation is similar to the Remembrance Agent [54, 56]. The Remembrance Agent indexes specified files such as email messages and research papers, and continually searches for related documents while a user edits a document in the Emacs editor. Other related projects include: Margin Notes [55], which rewrites web pages to include links to related personal files; the Haystack project [1], which aims to create a community of interacting “haystacks” or personal information repositories; and Autonomy’s Kenjin program (*www.kenjin.com*), which automatically suggests content from the web or local files, based on the documents a user is reading or editing. Also related are agents that learn user interest profiles for recommending web pages such as Fab [4], Letizia [47], WebWatcher [3], and Syskill and Webert [51].

2.3 Personalized Search

The next step is complete personalization of search – a search engine that knows all of your previous requests and interests, and uses that information to tailor results. Thus, a request for "Michael Jordan" may be able to rank links to the professor of computer science and statistics highly amongst links to the famous basketball player, for an individual with appropriate interests.

Such a personalized search engine could be either server or client-based. A server-based search engine like Google could keep track of a user's previous queries and selected documents, and use this information to infer user interests. For example, a user that often searches for computer science related material may have the homepage of the computer scientist ranked highly for the query "Michael Jordan", even if the user has never searched for "Michael Jordan" before.

A client-based personalized search service can keep track of all of the documents edited or viewed by a user, in order to obtain a better model of the user's interests. However, these services do not have local access to a large scale index of the web, which limits their functionality. For example, such a service could not rank the homepage of the computer scientist highly for the query "Michael Jordan", unless a search service returns the page within the maximum number of results that the client retrieves. The clients may modify queries to help retrieve documents related to a given context, however this is difficult for the entire interests of a user. Watson and Kenjin are examples of client-based personalized web search engines. Currently, Watson and Kenjin extract context information only from the current document that a user is editing or viewing.

With the cost of running a large scale search engine already very high, it is likely that server-based full-scale personalization is currently too expensive for the major web search engines. Most major search engines (Northern Light is an exception) do not even provide an alerting service that notifies users about new pages matching specific queries. However, advances in computer resources should make large scale server-based personalized search more feasible over time. Some Internet companies already devote a substantial amount of storage to individual users. For example, companies like DriveWay (*www.driveway.com*) and Xdrive (*www.xdrive.com*) offer up to 100Mb of free disk storage to each user.

One important problem with personalized search services is that users often expect consistency – they would like to receive the same results for the same queries, whereas a personalized search engine may return different results for the same query, both for different users, and also for the same user as the engine learns more about the user. Another very important issue, not addressed here, is that of privacy – many users want to limit the storage and use of personal information by search engines and other companies.

2.4 Guessing What the User Wants

An increasingly common technique on the web is guessing the context of user queries. The search engines Excite (*www.excite.com*), Lycos (*www.lycos.com*), Google (*www.google.com*), and Yahoo (*www.yahoo.com*) provide special functionality for certain kinds of queries. For example, queries to Excite and Lycos that match the name of an artist or company produce additional results that link directly to artist or company information. Yahoo recently added similar functionality, and provides specialized results for many different types of queries – e.g., stock symbols provide stock quotes and links to company information, and sports team names link to team and league information. Other examples for Yahoo include car models, celebrities, musicians, major cities, diseases and drug names, zodiac signs, dog breeds, airlines, stores, TV shows, and national parks. Google (*www.google.com*) identifies queries that look like a U.S. street address, and provides direct links to maps. Similarly, Google keeps track of recent news articles, and provides links to matching articles when found, effectively guessing that the user might be looking for news articles.

Rather than explicitly requiring the user to enter context information such as "I'm looking for a news article" or "I want a stock quote", this technique guesses when such contexts may be relevant. Users can relatively easily identify contexts of interest. This technique is limited to cases where potential contexts can be identified based on the keyword query. Improved guessing of search contexts could be done by a personalized search engine. For example, the query “Michael Jordan” might return a link to a list of Prof. Michael Jordan’s publications in a scientific database for a user interested in computer science, guessing that such a user may be looking for a list of publications by Prof. Jordan.

Clustering of search results, as performed by Northern Light for example, is related. Northern Light dynamically clusters search results into categories such as “current news” and “machine learning”, and allows a user to narrow results to any of these categories.

3 Restricting the Context of Search Engines

Another way to add context into web search is to restrict the context of the search engine, i.e., to create specialized search engines for specific domains. Thousands of these search engines already exist (see *www.invisibleweb.com* and *www.completeplanet.com*). Many of these services provide similar functionality to regular web search engines, either for information that is on the publicly indexable web (only a fraction of which may be indexed by the regular search engines), or for information that is not available to regular search engines (e.g., the New York Times search engine). However, an increasing number of specialized search engines are appearing which provide functionality far beyond that provided by regular web search engines, within their specific domain.

3.1 Information Extraction and Domain-Specific Processing

ResearchIndex (also known as CiteSeer) [40, 44, 45] is a specialized search engine for scientific literature. ResearchIndex is a free public service (available at *www.researchindex.com*), and is the world’s largest free full-text index of scientific literature, currently indexing over 300,000 articles containing over 3 million citations. It incorporates many features specific to scientific literature. For example, ResearchIndex automates the creation of citation indices for scientific literature, provides easy access to the context of citations to papers, and has specialized functionality for extracting information commonly found in research articles.

Other specialized search engines that do information extraction or domain-specific processing include DEADLINER [37], which parses conference and workshop information from the web, newsgroups and mailing lists; FlipDog (*www.flipdog.com*), which parses job information from employee sites; HPSearch (*http://hpsearch.uni-trier.de/hp/*), which indexes the homepages of computer scientists; and GeoSearch [14, 23], which uses information extraction and analysis of link sources in order to determine the geographical location and scope of web resources. Northern Light also provides a service called GeoSearch, however Northern Light’s GeoSearch only attempts to extract addresses from web pages, and does not incorporate the concept of the geographical scope of a resource (for example, the New York Times is located in New York but is of interest in a larger geographical area, whereas a local New York newspaper may be of less interest outside New York).

Search engines like ResearchIndex, DEADLINER, FlipDog, HPSearch, and GeoSearch automatically extract information from web pages. Many methods have been proposed for such information extraction, see for example [2, 9, 20, 38, 39, 40, 58, 59].

3.2 Identifying Communities on the Web

Domain-specific search engines that target the publicly indexable web need a method of locating the subset of the web within their domain. Flake et al. [25] have recently shown that the link structure of the web self-organizes such that communities of highly related pages can be efficiently identified based purely on connectivity. A web *community* is defined as a collection of pages where each member has more links (in either direction) inside the community than outside of the community (the definition may be generalized to identify communities of various sizes and with varying levels of cohesiveness). This discovery is important because there is no central authority or process governing the formation of links on the web. The discovery allows identification of communities on the web independent of, and unbiased by, the specific words used. An algorithm for efficient identification of these communities can be found in [25].

Several other methods for locating communities of related pages on the web have been proposed, see for example [7, 15, 16, 17, 22, 27, 36, 53].

3.3 Locating Specialized Search Engines

With thousands of specialized search engines, how do users locate those of interest to them? More importantly, perhaps, how many users will go to the effort of locating the best specialized search engines for their queries? Many queries that would be best served by specialized services are likely to be sent to the major web search engines because the overhead in locating a specialized engine are too great.

The existence of better methods for locating specialized search engines can help, and much research has been done in this area. Several methods of selecting search engines based on user queries have been proposed, for example GlOSS [33, 34] maintains word statistics on available databases, in order to estimate which databases are most useful for a given query. Related research includes [19, 24, 26, 32, 46, 49, 61, 62].

It would be of great benefit if the major web search engines attempted to direct users to the best specialized search engine where appropriate, however many of the search engines have incentives not to provide such a service. For example, they may prefer to maximize use of other services that they provide.

4 One Size Does Not Fit All, and May Limit Competition

Typical search engines can be viewed as "one size fits all" – all users receive the same responses for given queries. As argued earlier, this model may not optimally serve many queries, but are there larger implications?

An often stated benefit of the web is that of equalizing access to information. However, not much appears to be equal on the web. For example, the distribution of traffic and links to sites is extremely skewed and approximates a power law [5, 35], with a disproportionate share of traffic and links going to a small number of very popular sites. Evidence of a trend towards "winners take all" behavior can be seen in the market share of popular services. For example, the largest conventional book retailer (Barnes & Noble) has less than 30% market share, however the largest online book retailer (Amazon) has over 70% market share [52].

Search engines may contribute to such statistics. Prior to the web, consumers may have located a store amongst all stores listed in the phone book. Now, an increasing number of consumers locate stores via search engines. Imagine if most web searches for given keywords result in the same sites being ranked highly, perhaps with popularity measures incorporated into the selection and ranking criteria [43]. Even if only a small percentage of people use search engines to find stores, these people may then create links on the web to the stores, further enhancing any bias towards locating a given store. More generally, the experience of locating a given item on the web may be more of a common experience amongst everyone, when compared with previous means of locating items (for example, looking in the phone book, walking around the neighborhood, or asking a friend). Note that this is different to another trend that may be of concern – namely the trend towards less common experiences watching TV, for example, where increasing numbers of cable channels, and increasing use of the web, mean that fewer people watch the same programs.

Biases in access to information can be limited by using the appropriate search service for each query. While searches for stores on the major web search engines may return biased results, users may be able to find less biased listings in online Yellow Pages phone directories. As another example, when searching with the names of the U.S. presidential candidates in February 2000, there were significant differences between the major web search engines in the probability of the official candidate homepages being returned on the first page of results [60]. Similar searches at specialized political search engines may provide less biased results. However, the existence of less biased services does not prevent bias in information access if many people are using the major web search engines. Searches at directory sites like Yahoo or the Open Directory may also be less biased, although there may be significant and unequal delays in listing sites, and many sites are not listed in these directories.

The extent of the effects of such biases depends on how often people use search engines to locate items, and on the kinds of search engines that they use. New search services that incorporate context, and further incorporation of context into existing search services, may increase competition, diversity, and functionality, and help mitigate any negative effects of biases in access to information on the web.

5 Summary

Search engines make an unprecedented amount of information quickly and easily accessible – their contribution to the web and society has been enormous. However, the "one size fits all" model of web search may limit diversity, competition, and functionality. Increased use of context in web search may help. As web search becomes a more important function within society, the need for even better search services is becoming increasingly important.

References

[1] E. Adar, D. Karger, and L. Stein. Haystack: Per-user information environments. In *Proceedings of the 1999 Conference on Information and Knowledge Management, CIKM*, 1999.

[2] E. Agichtein and L. Gravano. Snowball: Extracting relations from large plain text collections. In *Proceedings of the 5th ACM International Conference on Digital Libraries*, 2000.

[3] R. Armstrong, D. Freitag, T. Joachims, and T. Mitchell. WebWatcher: A learning apprentice for the World Wide Web. 1995.

[4] Marko Balabanovic. An adaptive web page recommendation service. In *Proceedings of the First International Conference on Autonomous Agents*, pages 378–385. ACM Press, New York, 1997.

[5] Albert-Lászlo Barabasi and Réka Albert. Emergence of scaling in random networks. *Science*, 286:509–512, 1999.

[6] Carol L. Barry. *The Identification of User Criteria of Relevance and Document Characteristics: Beyond the Topical Approach to Information Retrieval*. PhD thesis, Syracuse University, 1993.

[7] K. Bharat and M.R. Henzinger. Improved algorithms for topic distillation in a hyperlinked environment. In *SIGIR Conference on Research and Development in Information Retrieval*, 1998.

[8] J. Boyan, D. Freitag, and T. Joachims. A machine learning architecture for optimizing web search engines. In *Proceedings of the AAAI Workshop on Internet-Based Information Systems*, 1996.

[9] S. Brin. Extracting patterns and relations from the World Wide Web. In *WebDB Workshop at EDBT 98*, 1998.

[10] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In *Seventh International World Wide Web Conference*, Brisbane, Australia, 1998.

[11] J. Budzik and K.J. Hammond. User interactions with everyday applications as context for just-in-time information access. In *Proceedings of the 2000 International Conference on Intelligent User Interfaces*, New Orleans, Louisiana, 2000. ACM Press.

[12] J. Budzik, K.J. Hammond, C. Marlow, and A. Scheinkman. Anticipating information needs: Everyday applications as interfaces to Internet information servers. In *Proceedings of the 1998 World Conference of the WWW, Internet and Intranet*, Orlando, Florida, 1998. AACE Press.

[13] Jay Budzik, Kristian J. Hammond, Larry Birnbaum, and Marko Krema. Beyond similarity. In *Proceedings of the 2000 Workshop on Artificial Intelligence and Web Search.* AAAI Press, 2000.

[14] O. Buyukkokten, J. Cho, H. García-Molina, L. Gravano, and N. Shivakumar. Exploiting geographical location information of web pages. In *Proc. of the ACM SIGMOD Workshop on the Web and Databases, WebDB*, 1999.

[15] S. Chakrabarti, B. Dom, D. Gibson, J. Kleinberg, P. Raghavan, and S. Rajagopalan. Automatic resource list compilation by analyzing hyperlink structure and associated text. In *Proceedings of the 7th International World Wide Web Conference*, 1998.

[16] Soumen Chakrabarti, Martin van den Berg, and Byron Dom. Focused crawling: A new approach to topic-specific web resource discovery. In *8th World Wide Web Conference*, Toronto, May 1999.

[17] Junghoo Cho, Héctor García-Molina, and Lawrence Page. Efficient crawling through URL ordering. In *Proceedings of the Seventh World-Wide Web Conference*, 1998.

[18] Frans Coetzee, Eric Glover, Steve Lawrence, and C. Lee Giles. Feature selection in web applications using ROC inflections. In *Symposium on Applications and the Internet, SAINT*, San Diego, CA, January 8–12 2001.

[19] N. Craswell, P. Bailey, and D. Hawking. Server selection on the World Wide Web. In *Proceedings of the Fifth ACM Conference on Digital Libraries*, pages 37–46, 2000.

[20] M. Craven, D. DiPasquo, D. Freitag, A. McCallum, T. Mitchell, K. Nigam, and S. Slattery. Learning to extract symbolic knowledge from the World Wide Web. In *Proceedings of Fifteenth National Conference on Artificial Intelligence, AAAI 98*, pages 509–516, 1998.

[21] B. D. Davison, A. Gerasoulis, K. Kleisouris, Y. Lu, H. Seo, W. Wang, and B. Wu. DiscoWeb: Applying link analysis to web search. In *Proceedings of the Eighth International World Wide Web Conference*, page 148, Toronto, Canada, 1999.

[22] Michelangelo Diligenti, Frans Coetzee, Steve Lawrence, C. Lee Giles, and Marco Gori. Focused crawling using context graphs. In *26th International Conference on Very Large Databases, VLDB 2000*, Cairo, Egypt, 10–14 September 2000.

[23] Junyan Ding, Luis Gravano, and Narayanan Shivakumar. Computing geographical scopes of web resources. In *26th International Conference on Very Large Databases, VLDB 2000*, Cairo, Egypt, September 10–14 2000.

[24] D. Dreilinger and A. Howe. Experiences with selecting search engines using meta-search. *ACM Transactions on Information Systems*, 15(3):195–222, 1997.

[25] Gary Flake, Steve Lawrence, and C. Lee Giles. Efficient identification of web communities. In *Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 150–160, Boston, MA, August 20–23 2000.

[26] Susan Gauch, Guihun Wang, and Mario Gomez. ProFusion: Intelligent fusion from multiple, distributed search engines. *Journal of Universal Computer Science*, 2(9), 1996.

[27] D. Gibson, J. Kleinberg, and P. Raghavan. Inferring web communities from link topology. In *Proceedings of the 9th ACM Conference on Hypertext and Hypermedia*, 1998.

[28] Eric Glover, Gary Flake, Steve Lawrence, William P. Birmingham, Andries Kruger, C. Lee Giles, and David Pennock. Improving category specific web search by learning query modifications. In *Symposium on Applications and the Internet, SAINT*, San Diego, CA, January 8–12 2001.

[29] Eric Glover, Steve Lawrence, William Birmingham, and C. Lee Giles. Architecture of a metasearch engine that supports user information needs. In *Eighth International Conference on Information and Knowledge Management, CIKM 99*, pages 210–216, Kansas City, Missouri, November 1999.

[30] Eric J. Glover, Steve Lawrence, Michael D. Gordon, William P. Birmingham, and C. Lee Giles. Web search – your way. *Communications of the ACM*, 2000. Accepted for publication.

[31] Graphic, Visualization, and Usability Center. GVU's tenth WWW user survey (conducted October 1998), 1998.

[32] L. Gravano, C. Chang, H. García-Molina, and A. Paepcke. STARTS: Stanford proposal for Internet meta-searching. In *Proc. of the 1997 ACM SIGMOD International Conference on Management of Data*, pages 207–218, 1997.

[33] L. Gravano, H. García-Molina, and A. Tomasic. GlOSS: Text-source discovery over the Internet. *ACM Transactions on Database Systems*, 24(2), 1999.

[34] Luis Gravano and Héctor García-Molina. Generalizing GlOSS to vector-space databases and broker hierarchies. In *International Conference on Very Large Databases, VLDB*, pages 78–89, 1995.

[35] B.A. Huberman, P.L.T. Pirolli, J.E. Pitkow, and R.M. Lukose. Strong regularities in World Wide Web surfing. *Science*, 280:95–97, 1998.

[36] J. Kleinberg. Authoritative sources in a hyperlinked environment. In *Proceedings ACM-SIAM Symposium on Discrete Algorithms*, pages 668–677, San Francisco, California, 25–27 January 1998.

[37] Andries Kruger, C. Lee Giles, Frans Coetzee, Eric Glover, Gary Flake, Steve Lawrence, and Cristian Omlin. DEADLINER: Building a new niche search engine. In *Ninth International Conference on Information and Knowledge Management, CIKM 2000*, Washington, DC, November 6–11 2000.

[38] N. Kushmerick. Wrapper induction: Efficiency and expressiveness. In *AAAI-98 Workshop on AI and Information Integration*, 1998.

[39] N. Kushmerick, D. Weld, and R. Doorenbos. Wrapper induction for information extraction. In *IJCAI 97*, pages 729–735, Nagoya, Japan, 1997.

[40] Steve Lawrence, Kurt Bollacker, and C. Lee Giles. Indexing and retrieval of scientific literature. In *Eighth International Conference on Information and Knowledge Management, CIKM 99*, pages 139–146, Kansas City, Missouri, November 1999.

[41] Steve Lawrence and C. Lee Giles. Context and page analysis for improved web search. *IEEE Internet Computing*, 2(4):38–46, 1998.

[42] Steve Lawrence and C. Lee Giles. Searching the World Wide Web. *Science*, 280(5360):98–100, 1998.

[43] Steve Lawrence and C. Lee Giles. Accessibility of information on the web. *Nature*, 400(6740):107–109, 1999.

[44] Steve Lawrence and C. Lee Giles. Searching the web: General and scientific information access. *IEEE Communications*, 37(1):116–122, 1999.

[45] Steve Lawrence, C. Lee Giles, and Kurt Bollacker. Digital libraries and autonomous citation indexing. *IEEE Computer*, 32(6):67–71, 1999.

[46] D. Leake, R. Scherle, J. Budzik, and K. Hammond. Selecting task-relevant sources for just-in-time retrieval. In *Proceedings of the AAAI-99 Workshop on Intelligent Information Systems*, Menlo Park, CA, 1999. AAAI Press.

[47] H. Lieberman. Letizia: An agent that assists web browsing. In *1995 International Joint Conference on Artificial Intelligence*, Montreal, CA, 1995.

[48] Media Metrix. Media Metrix announces top 25 digital media/web properties and sites for January 1999, 1999.

[49] W. Meng, K. Liu, C. Yu, W. Wu, and N. Rishe. Estimating the usefulness of search engines. In *15th International Conference on Data Engineering, ICDE*, Sydney, Australia, 1999.

[50] Stefano Mizzaro. Relevance: The whole history. *Journal of the American Society for Information Science*, 48(9):810–832, 1997.

[51] M. Pazzani, J. Muramatsu, and D. Billsus. Syskill & Webert: Identifying interesting web sites. In *Proceedings of the National Conference on Artificial Intelligence, AAAI*, 1996.

[52] Ivan Png. The competitiveness of on-line vis-a-vis conventional retailing: A preliminary study. In *11th NEC Research Symposium*, Stanford, CA, 2000.

[53] J. Rennie and A. McCallum. Using reinforcement learning to spider the web efficiently. In *Proceedings of the Sixteenth International Conference on Machine Learning (ICML-99)*, 1999.

[54] Bradley Rhodes. *Just-in-Time Information Retrieval*. PhD thesis, Massuchesetts Institute of Technology, 2000.

[55] Bradley J. Rhodes. Margin Notes: Building a contextually aware associative memory. In *Proceedings of the International Conference on Intelligent User Interfaces, IUI 00*, 2000.

[56] Bradley J. Rhodes and Thad Starner. Remembrance Agent: A continuously running automated information retrieval system. In *Proceedings of the First International Conference on the Practical Application of Intelligent Agents and Multi Agent Technology*, pages 487–495, 1996.

[57] G. Salton. *Automatic text processing: the transformation, analysis and retrieval of information by computer*. Addison-Wesley, 1989.

[58] Kristie Seymore, Andrew McCallum, and Roni Rosenfeld. Learning hidden Markov model structure for information extraction. In *AAAI 99 Workshop on Machine Learning for Information Extraction*, 1999.

[59] S. Soderland. Learning information extraction rules for semi-structured and free text. *Machine Learning*, 34(1):233–272, 1999.

[60] D. Sullivan. Can you find your candidate? Search Engine Watch, February 29 2000.

[61] J. Xu and J. Callan. Effective retrieval with distributed collections. In *Proceedings of the 21st International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 112–120, 1998.

[62] J. Zobel. Collection selection via lexicon inspection. In *Proceedings of the 1997 Australian Document Computing Symposium*, pages 74–80, Melbourne, Australia, 1997.

Searching for Needles in a World of Haystacks

Jamie Callan
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15213-3890, USA
callan+@cs.cmu.edu

Abstract

Current Web search engines are based on a single database *model of information retrieval. This paper argues that next generation Web search will be based on a* multi-database *model of information retrieval. The strengths, weaknesses, open research problems, and prospects of several multi-database retrieval models are discussed briefly.*

1 Introduction

Web search engines provide access to HTML documents that are stored on computers around the world. The distributed nature of the document collection provides the illusion of a complex information access task, but the first generation of Web search engines is actually based on a simple retrieval paradigm. Documents are copied from their original locations to a central site, where they are added to a central database, indexed, and made available for search. The first generation of Web search engines is based on a *single database* model of information retrieval.

The single database model of information retrieval was successful because initially i) the Web was not very large, ii) the Web was not a commercial medium, and iii) HTML pages were not very complex. A large, but not *too large*, amount of information was available for copying, freely, in a format that was easy to interpret. As the Web evolved, these conditions were violated. There was too much information for it all to be copied, access to some of it became restricted, and a large amount of information was in searchable databases or was generated dynamically in response to a request or user action. Web search engines today provide access to only a small portion of the information on the Web.

The single database approach to Web search is a poor foundation upon which to build next generation systems. New, multi-database architectures provide a more solid foundation, because they explicitly model the multi-site, multi-resource nature of the Web. Traditional Web search engines based on the single database model can still play a valuable role in a multi-database environment, but it will no longer be the central role.

Copyright 2000 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

Bulletin of the IEEE Computer Society Technical Committee on Data Engineering

2 Multi-Database Retrieval Models

Multi-database retrieval models can be classified as appropriate for *small-scale* or *large-scale* environments. A *small-scale* environment might be one that contains a few hundred text databases, perhaps under the control of a single organization. A *large-scale* environment might be one that contains tens of thousands of text databases, perhaps under the control of multiple organizations.

We consider a multi-database retrieval model appropriate for a small-scale environment if human effort is required to maintain a database hierarchy, a knowledge base [13, 4], or training data [14] used for database selection. The determining factor is the ease with which a newly-discovered text database can be made accessible. We also consider a multi-database retrieval model appropriate for a small-scale environment if each query requires some degree of communication with every database [9].

Multi-database retrieval models for small-scale environments have their advantages. For example, manual database organization or the use of training data can enable very accurate database selection for typical queries. This approach to database selection is most likely to be used by commercial information services and in customer support centers, for example, where there is a premium on accuracy and less concern about the rapid introduction of new text databases on a regular basis.

We consider a multi-database retrieval model appropriate for a large-scale environment, such as the Web, if i) new text databases can be introduced quickly, easily, and automatically; and ii) only limited communication is required to decide which text databases a particular query should be sent to. Retrieval models that satisfy these requirements can be grouped into two classes: message-passing, and centralized selection.

3 Message-passing

A *message-passing* multi-database retrieval model is one in which a search request is repeatedly passed from one node in a computer network to adjacent nodes until either an answer is found or it reaches a search horizon. If each node passes requests to n neighbors, the search horizon at distance h contains n^h nodes, so search requests can be propagated rapidly to many computers in a distributed manner. Matching objects can be passed back along the search path (e.g., Freenet [5]), or a direct connection can be established between the original search node and the matching node(s) (e.g., Gnutella [10]).

Message-passing models are a truly distributed solution to searching across multiple databases. There is no central site that can be attacked or otherwise shut down [3]. Each node decides for itself whether it has content that matches the query, which makes it easy to integrate diverse information sources, dynamic information content, and resources that charge for content. It is also possible to offer a search client complete anonymity in a message-passing model [5], although doing so may entail significant communication costs for every node between the original search client and nodes with matching content.

Some characteristics of message-passing retrieval models are likely to limit their effectiveness as models for next-generation Web search systems. Perhaps the most serious is one of scale. If a message-passing model is applied on a large scale, the amount of additional Internet traffic and unnecessary computation is staggering. For example, a *single query* with fanout 4 and search horizon 10 would involve $\Sigma_{i=1}^{10} 4^i = 1,398,100$ nodes. It is difficult to imagine applying this model to millions of queries per day.

Even on a small-scale, message-passing models are *non-deterministic*, because answers depend upon the topology of the network at a particular time, and *non-optimal*, because only a portion of the network is searched for each query. Message-passing models are also prone to *spoofing*, in which a node claims to have matching information when it does not.

The limitations of the message-passing model are opportunities for interesting research. For example, issues of scale, non-determinism, and non-optimality might all be addressed by less random message-passing, while spoofing might be handled by holding each node responsible for its past actions. Whatever the solution, these problems must be addressed before message passing architectures are a viable, large-scale solution to multi-database text retrieval.

4 Centralized Resource Selection

Perhaps the most fully developed multi-database retrieval models are based on centralized database selection. A central site gathers information about available databases, organizes it into an index, and then provides a resource selection service to search clients. A simple resource selection service might just return a ranked list of resources (searchable text databases) to the client. A more complex service might select a set of resources, submit the query to those resources, and return to the search client a single, merged list of documents obtained from multiple resources.

The best-known models for centralized database selection are based on the GlOSS family of algorithms [8], the CORI algorithm [1], and the Cue Validity Variance (CVV) algorithm [16]. The three algorithms differ in their details, but all are based on matching queries to "bag of words" representations of text databases. The "bag of words" resource representation is created automatically, making it relatively easy to add new resources to an existing resource index. The algorithms are general, relatively robust, and do not necessarily require the cooperation of the resource provider [1]. Unlike message-passing architectures, the centralized selection model is computationally efficient, is deterministic, and considers all resources for all queries.

Centralized services can be attacked, legally or illegally, and do not provide the anonymity that message-passing retrieval models provide. They also raise interesting research issues that are less likely to arise with message-passing retrieval models and retrieval models for small-scale environments.

Resource identification: Techniques for crawling the Web are well-defined, but it can be difficult to identify the type of resource at the other end of a hyperlink. For example, the Common Gateway Interface (CGI) hyperlink type indicates an executable resource, but the resource capabilities are not described. The resource might be a text database with search capabilities, a relational database, or a counter. Text associated with the link might or might not indicate something about its capabilities. A crawler that encounters a link to a CGI resource usually has few clues about the type of resource it points to.

Resource identification can be viewed as a document categorization task, for example assigning documents into categories such as "text database", "relational database", and "other" based on multiple forms of uncertain evidence.

Interoperability: Multi-database models designed for small-scale environments and message-passing retrieval models assume that all resources conform to a common interoperability standard. Multi-database retrieval models for large-scale environments are likely to find this assumption violated more often than satisfied. Rather than excluding such resources, it may be possible learn how to communicate with the resource via its Web page, for example, by probing it and examining its responses. Information can be extracted from returned Web pages using *wrapper induction* techniques [11]. Current wrapper induction algorithms are based on supervised learning, but large-scale interoperability requires wrapper induction algorithms based on unsupervised learning.

Resource representation: Content-based resource selection is based on knowing what subject area(s) each resource covers. For example, if the resource is an archive of Wall Street Journal newspaper articles, it might be described as covering the domains of stocks, bonds, international finance, politics, and similar subjects.

If the resource is a text database, and is controlled by a trusted and cooperative party, the STARTS protocol [7] is an effective method of discovering its contents. The STARTS protocol enables a search service and an information resource to exchange information about the contents and capabilities of the resource. If the resource is controlled by an uncooperative party, the contents of the resource can be discovered by running queries and analyzing the documents that are returned (*query-based sampling*) [1].

Both STARTS and query-based sampling are sufficient for bag-of-words resource representations. More complex resource representations, for example, representations based on statistical language models [15], may require new approaches.

Resource selection: Given a set of information resources, content-based resource selection algorithms rank them by the likelihood that they will satisfy the information need described by a query. The GlOSS family of algorithms [8], CORI [1], and Cue Validity Variance (CVV) [16] are three examples that are based on variations of algorithms for ranking documents. The current generation of algorithms can be effective, but there is also evidence that they are fragile when expectations about query length, document length, and resource description quality are not met [2]. Even under ideal conditions, accuracy is considerably lower than theoretical limits [6].

Result merging: After a set information resources is searched, it may be desirable to merge the results into a single display or representation. This problem is difficult when resources are uncooperative. The state-of-the-art is either to rerank the documents at the search client, to merge the rankings using heuristics [1], or to reverse-engineer the ranking functions of the various search engines [12]. Each of these solutions has weaknesses, hence this remains an important, if often overlooked, research problem.

Robust, large-scale centralized resource selection is arguably the most complex of the multi-database retrieval models discussed above, because it requires solutions to several independent problems. Each of the problems outlined above is interesting and challenging because solutions must be general, automatic, and robust, and because they cannot rely on cooperation from information providers. The lack of cooperation, in particular, makes large-scale centralized resource selection both an appealing solution and an interesting research problem.

5 Forecast

All of the multi-database retrieval models surveyed in this paper are likely to play a role in next generation networked information systems. Models optimized for accuracy in small-scale environments will be used in corporate environments, and on the Web in specific subject areas (e.g., to select among databases covering medical research). Research efforts on this class of multi-database retrieval models will focus on improving accuracy and on reducing the amount of manual intervention necessary.

There are two adoption scenarios for message-passing retrieval models. Message-passing is a good choice for dynamic and diverse information content, because the information provider, which knows its content best, determines what information it can provide for each request. However, this same characteristic causes problems such as spoofing; centralized resource selection may provide adequate access and greater reliability. Message-passing is also a good choice when it is important to present no single point of information access, and when anonymity is important. The most obvious examples are theft of intellectual property such as music and software, and exchange of politically sensitive information.

Centralized resource selection is the retrieval model most likely to be the basis for the next generation of large-scale Web search systems. It scales well, it is effective, it provides consistent results, it supports heterogeneous text resources, and it supports heterogeneous query streams. There are research issues that must be addressed before multi-database retrieval models start becoming common, but they appear to be “near term” research opportunities, rather than “long term” research challenges.

Acknowledgements

This material is based on work supported in part by the National Science Foundation under grants IIS-9873009 and EIA-9983253. Any opinions, findings, conclusions or recommendations expressed in this material are the authors', and do not necessarily reflect those of the sponsor(s).

References

[1] J. Callan. Distributed information retrieval. In W.B. Croft, editor, *Advances in information retrieval*, chapter 5, pages 127–150. Kluwer Academic Publishers, 2000.

[2] J. Callan, A. L. Powell, J. C. French, and M. Connell. The effects of query-based sampling on automatic database selection algorithms. Technical Report IR-181, Center for Intelligent Information Retrieval, Department of Computer Science, University of Massachusetts, 1999.

[3] A. E. Cha. E-power to the people. In *The Washington Post*, page A01, May 18 2000.

[4] A. S. Chakravarthy and K. B. Haase. NetSerf: Using semantic knowledge to find Internet information archives. In *Proceedings of the Eighteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 4–11, Seattle, 1995. ACM.

[5] I. Clarke. Freenet - Front page. http://freenet.sourceforge.net/, August 2000.

[6] J. French, A. Powell, J. Callan, C. Viles, T. Emmitt, K. Prey, and Y. Mou. Comparing the performance of database selection algorithms. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 238–245. ACM, 1999.

[7] L. Gravano, K. Chang, H. García-Molina, and A. Paepcke. STARTS Stanford proposal for Internet meta-searching. In *Proceedings of the ACM-SIGMOD International Conference on Management of Data*, 1997.

[8] L. Gravano, H. García-Molina, and A. Tomasic. GlOSS: Text-Source Discovery over the Internet. *ACM Transactions on Database Systems*, 24(2):229–264, 1999.

[9] D. Hawking and P. Thistlewaite. Methods for information server selection. *ACM Transactions on Information Systems*, 17(1):40–76, 1999.

[10] Wego.com Incorporated. Welcome to gnutella. http://gnutella.wego.com/, August 2000.

[11] N. Kushmerick, D. S. Weld, and R. Doorenbos. Wrapper induction for information extraction. In *Proceedings of the 15th International Joint Conference on Artificial Intelligence (IJCAI-97)*, 1997.

[12] K. Liu, W. Meng, C. Yu, and N. Rishe. Discovery of similarity computations of search engines. In *Proceedings of the 9th International Conference on Information and Knowledge Management (CIKM)*. ACM, 2000.

[13] R. S. Marcus. An experimental comparison of the effectiveness of computers and humans as search intermediaries. *Journal of the American Society for Information Science*, 34:381–404, 1983.

[14] E. M. Voorhees, N. K. Gupta, and B. Johnson-Laird. Learning collection fusion strategies. In *Proceedings of the Eighteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 172–179, Seattle, 1995. ACM.

[15] J. Xu and W. B. Croft. Cluster-based language models for distributed retrieval. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 254–261, Berkeley, 1999. ACM.

[16] B. Yuwono and D. L. Lee. Search and ranking algorithms for locating resources on the World Wide Web. In S. Y. W. Su, editor, *Proceedings of the 12th International Conference on Data Engineering*, pages 164–171, New Orleans, 1996.

Next Generation Web Search: Setting Our Sites

Marti A. Hearst
School of Information Management and Systems
University of California Berkeley
hearst@sims.berkeley.edu

Abstract

The current state of web search is most successful at directing users to appropriate web sites. Once at the site, the user has a choice of following hyperlinks or using site search, but the latter is notoriously problematic. One solution is to develop specialized search interfaces that explicitly support the types of tasks users perform using the information specific to the site. A new way to support task-based site search is to dynamically present appropriate metadata that organizes the search results and suggests what to look at next, as a personalized intermixing of search and hypertext.

1 Introduction

Surveys indicate that search engine user satisfaction has risen recently. According to one survey, 80% of search engine users say they find what they want all or most of the time [21]. This is a surprising result given average query length is still quite short – about 2 words. How can people be finding exactly what they want given such short queries?

Since no published large-scale analysis exists, we currently have to make guesses. I think the answer is that, as a gross generalization, most people use web search engines to find good starting points – home pages of web sites that discuss a topic of interest. A query on "horseradish" is very general; it does not indicate what it is the user wants to know about or do with horseradish, so the best thing a search engine can do is bring up sources of general information about horseradish which the user can then peruse in more detail.

This multi-stage process of search, starting with a general query and then getting more specific, is well-documented in non-web search [14]. Users of old search systems like Dialog and Lexis-Nexis were taught to first write a general query, look at how many (thousands) of results were returned, and then refine the query with additional terms until a reasonable number of documents resulted. In these older systems the user had to first select a collection, or source, to search. Many of these searchers were professionals, who knew a great deal about which sources were available after years of experience.

By contrast, in web search the purpose of the initial query seems primarily to be to choose the source – a web site of interest. Once at the source, the user has a choice of using hyperlinks to navigate through the many pages of information available, or using site search.

Copyright 2000 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

Bulletin of the IEEE Computer Society Technical Committee on Data Engineering

In this article, I use the term “site” to mean a collection of information that has some kind of unified theme. This can be a collection of items such as architectural images, recipes, or biomedical texts, or the catalog of an e-commerce company, or the intranet of a company or a university (the theme being the various kinds of work done and information used by people in the organization), or what’s begun to be called a “vortal” for vertical portal, which offers a wealth of different sources about one topic. FindLaw is an example of a vortal, providing search over dozens of different legal sources, including law journals, US Supreme court decisions, and legal news.

Major search engine companies are now offering site-specific search as part of their product suites. For example, Inktomi has announced plans to develop an architecture that allows flexible partitioning of information sets, allowing the assignment of weights to documents depending on which parts of the collection they occur in and what kinds of context they are associated with [19]. To best utilize this potentially powerful facility, an understanding is needed of how to combine this information effectively.

In the early days of web search, a query like “horseradish” would most likely have retrieved what felt like random pages. But two things have changed since then. First, more high-quality content has become available on a vast array of topics, and second, search engines now focus on returning definitive *sites*, rather than pages, for such general queries. For instance, the top hits for a search on “horseradish” on Lycos finds a link to the Horseradish Information Council, listed as part of the

Recreation > Food
Business > Industries > Food and Related Products > Fruit and Vegetables

web directory categories, and hits on encyclopedia entries for horseradish are shown alongside the

Reference > Encyclopedia > Microsoft Encarta > H
Reference > Encyclopedia > Encyclopedia.com > H

categories. These are followed by the home pages of various food products companies. (It is a pity that we cannot run this same query over the 1996 web using 1996 search engines for comparison purposes.)

The fact that search engines show hits on category labels is significant, because web directory categories, and their associated links, are manually selected to be representative starting points for search. The Google search engine is known for using hyperlink inlink information for ranking pages, based on the idea that if many pages link to a page, that linked-to page is likely to be of higher quality (recursively) because it is in effect “recommended” by the authors of the other pages. Others have also documented the merit of using inlink information for assessing page quality [12, 1], and other web search engines are making use of this information. However, Google also now incorporates category information into its search results, listing category labels beneath a search hit if that category had been assigned to it. An informal test on short general queries brought up on average 2.7 categories alongside the top 10 hits.[1]

A recent study [1] presents intriguing evidence that the *number of pages in the site* is a good predictor for the inlink-based ranking, at least for popular entertainment topics. In other words, the most popular sites on a topic according to inlink information are those sites that have a lot of information on the topic; the good sources. This bolsters the argument that what web search is good at doing is getting people to the right site or collection, after which the complicated information seeking begins.

2 Site Search and Tasks

In a well-designed web site, hyperlinks provide helpful hints about what is behind them and where to go next, a characteristic also known as “scent” [5, 16]. One usability expert claims that as a general rule, users do not object so much to following many links as they object to having to follow links that do not clearly support their task.

[1] The queries were horseradish, election, colon cancer, sex, berkeley, affirmative action, china, chat, recipes, united airlines.

If the user is unsure where to go next or has to resort to the back button, the usability of the site can decrease dramatically [20].

When the user has to resort to using the search facility, the results on a site are usually disorderly and shown out of context, and do not indicate what role the various hits play in the use or structure of the site. Usability gurus and ecommerce researchers alike lament the poor quality of site search, and claim that billions in business are lost each year due to poor site design and site search [20, 11].

What is a solution? Consider the following analogy. Following hyperlinks is like taking the train, whereas using site search is like driving an all-terrain four-wheel drive vehicle. On the train, there are a fixed number of choices of where to go and how to get there. To get from Topeka to Santa Fe you have to go through Frostbite Falls whether this make sense to you or not. On the other hand, you are unlikely to get lost – if you look at where you are on the map, all is clear. By contrast, a four-wheel drive Land Cruiser will take you anywhere, but you may get wedged between two boulders on the side of a cliff and be completely disoriented as to your whereabouts.

An ideal search interface adopts the best aspects of each technique. We would like a train system that magically lays down new track to suggest useful directions to go based on where we have been so far and what we are trying to do. The tracks follow the lay of the land, but cross over the crevasses and get to every useful part of the earth. They also allow us to back up at any time and take a different route at each choice point.

2.1 The Importance of the Task

Figuring out the best way to lay these tracks is nontrivial, because just as there is a huge variety of information available on these sites, there are also a huge variety of ways people use that information. To help cut down on the number of possible routes, the search interface, as well as the site structure, should reflect what is it people do with the site, that is, what *tasks* they attempt to accomplish on the site.

A recent study by Jared Spool's research firm uncovered the importance of task completion in user satisfaction in web site use [18]. The study compared 10 different sites; each participant conducted searches for information of interest to them on each site. Afterwards, Spool's team asked the participants to rate the web sites according to how fast they thought they were. Surprisingly, there was no correlation between the page download speeds and the perceived speed ratings. In fact, participants perceived the site with the fastest download speed to be the slowest, and vice versa.

However, there was a strong correlation between perceived speed and how successful the participants were at achieving their goals on the site. This was also correlated with how strongly the participant thought they usually "knew what to do next" at any given point in their task. Spool concludes that the correlational evidence suggests that if the goal is to improve perceived speed, it is more important to focus on designing web sites to help users complete their tasks, rather than focusing on task-neutral features like download speed.

I have now said that site search should reflect the tasks that a user would like to accomplish on the site, and I have suggested a metaphor about a magical train that lays tracks according to where you want to go and what you have done so far, a blending of the best features of hypertext and search. In the remainder of this section I will describe this idea in more detail and outline our research efforts in this direction.

2.2 Metadata

One more ingredient is needed before we can cook up this new idea. That is the notion of metadata. Metadata is commonly glossed as meaning "data about data". Most documents have some kinds of meta-information associated with them – that is, information that characterizes the external properties of the document, that help identify it and the circumstances surrounding its creation and use. These attributes include author(s), date of publication, length of document, publisher, and document genre.

Additionally, content-oriented subject or category metadata has become more prevalent in the last few years, and many people are interested in standards for describing content in various fields. Web directories such as

Yahoo and looksmart are familiar examples, and as seen above, web search engines have begun to interleave search hits on category labels with other search results.

Collections such as medical documents and architectural images have richer metadata available; some items have a dozen or more content attributes attached to them. It can be useful to think of category metadata as being composed of *facets*: orthogonal sets of categories, which together can be used to describe a topic. In the medical domain, the different facets are Disease type, Drug Type, Physiology, Surgery Type, Patient Type, and so on. Each article is a complex combination of several of these types of facets, each of which has a hierarchical structure. For example, a MedLine article entitled "Inhaled and systemic corticosteroid therapies: Do they contribute to inspiratory muscle weakness in asthma?" is assigned the MeSH categories *Steroidal Anti-Inflammatory Agents, Asthma, Muscular Diseases, Respiratory Muscles, Inhalation Administration, Adult, Beclomethasone, Case-Control Studies, Prednisone,* and *Risk Factors,* among others.

Researchers have long reported that using metadata in search is problematic, because the labels assigned often mismatch user expectations [15, 6]. Furthermore, category metadata is often inconsistent. Nevertheless, I believe the full potential of metadata in search results is underexplored and could yield significant improvements, especially for supporting task-based search over large collections of similar-style items (such as biomedical articles, architectural images, and recipes).

Metadata can be used as a counterpoint to free text, since free text and metadata can both be searched, but whereas free text queries must usually be subject to relevance ranking, metadata can be retrieved much as in a standard database query. It is much easier to accurately implement the query "Find all documents that have been assigned the category label Affirmative Action" than it is to implement "Find all documents about affirmative action".

2.3 An Example: epicurious

As a consequence of writing this paper, a website was brought to my attention that exhibits a good subset of the ideas I think can be useful for using metadata to improve task-oriented site search.[2] In this case the collection is recipe information; actually a database problem (fuzzy matching is not required) but many of the ideas can transfer to the fuzzier needs of information search.

Recipes are examples of information for which hierarchical faceted metadata is familiar to everyone. The facets used by epicurious are Main Ingredient, Cuisine, Preparation Method, Season/Occasion, and Course/Dish. Each of these has subcategories; for example, subcategories for Course/Dish include Appetizers, Bread, Desserts, Sandwiches, Sauces, Sides and Vegetables. The collection has over 11,000 recipes.

A standard search interface for recipes requires the user to either do a keyword search or drill down a category hierarchy. For example, on a different recipe site (called SOAR[3]), selecting Main Dishes > Poultry results in 57 recipes. To further refine these choices, some additional hyperlinked categories are shown:

> Poultry: Chicken Recipes; Poultry: Duck Recipes; Poultry: Game Hens;
> Poultry: Game Recipes; Poultry: Goose; Poultry: Turkey

Selecting Chicken retrieves a list of about 40 recipes and two more subcategories:

> Chicken Appetizers
> Diabetic Chicken Recipes

I could have also found the Chicken Appetizers category if I'd begun with the Appetizers category. However, if I'd wanted to see Italian recipes with chicken as a main dish, I have to first select Region > Italian, and then look

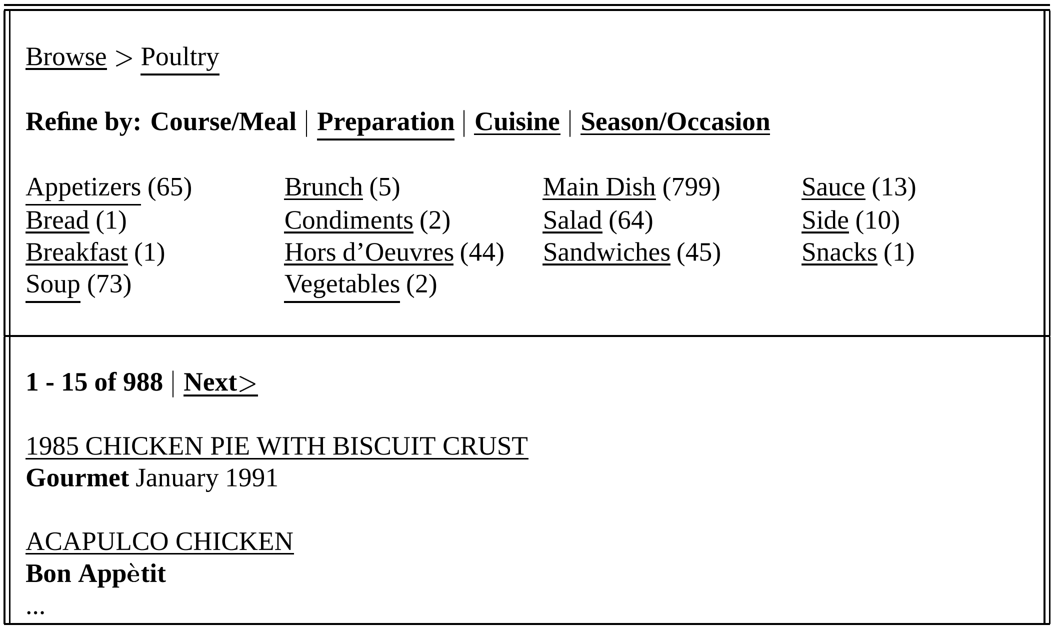
[2] http://www.epicurious.com/e_eating/e02_recipes/browse_main.html

[3] http://soar.berkeley.edu/recipes/

through all 665 Italian recipes. The site also allows a search for a particular term over the category, so I can do a search on “chicken” in the Italian section and hope for the best.

One problem with this interface is that I do not know how many recipes have a given attribute until after I follow the link. It would be much more useful to know this information when having to make a decision about where to go next. Researchers in the human-computer interaction community advocate the importance of information previews in this kind of situation [17]. Another flaw is the irregularity of what kinds of subcategories occur under a given category. Why are only appetizers shown as a special category, but not main dishes?

Recipe finding is in most cases a combination of a browsing and a search task, and so is a prime candidate for our ideas about combining the best of both techniques. The epicurious site does an excellent job of this. On this site, after selecting Main Ingredient > Poultry, the information of Figure 1 is shown.



Browse > Poultry

Refine by: Course/Meal | **Preparation** | **Cuisine** | **Season/Occasion**

Appetizers (65)	Brunch (5)	Main Dish (799)	Sauce (13)
Bread (1)	Condiments (2)	Salad (64)	Side (10)
Breakfast (1)	Hors d’Oeuvres (44)	Sandwiches (45)	Snacks (1)
Soup (73)	Vegetables (2)		

1 - 15 of 988 | **Next>**

1985 CHICKEN PIE WITH BISCUIT CRUST
Gourmet January 1991

ACAPULCO CHICKEN
Bon Appètit
...

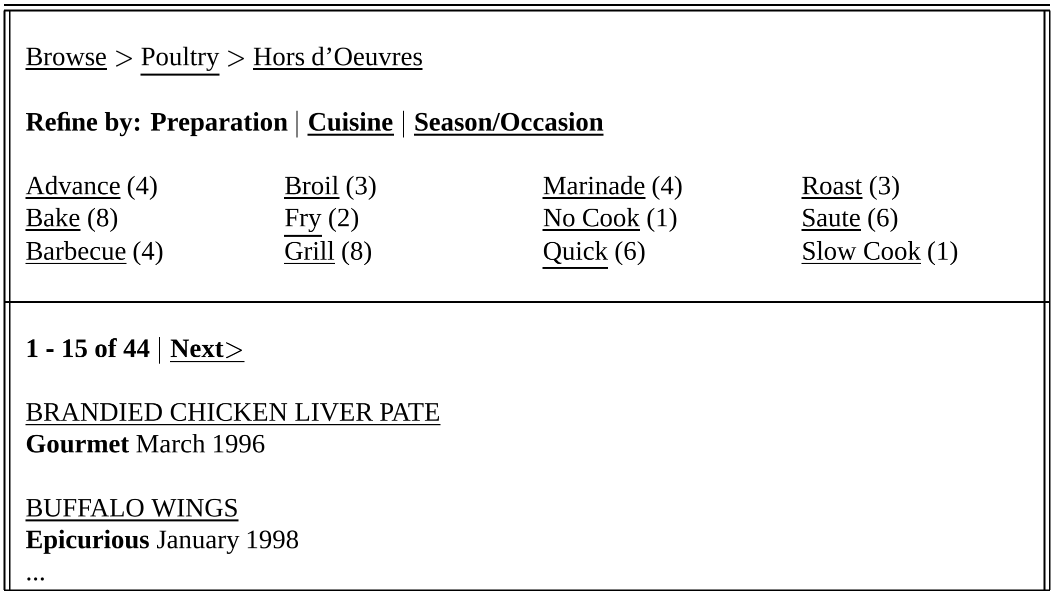
Figure 1: A sketch of the epicurious site’s method for browsing dynamic metadata describing a collection of recipes. After selecting a main dish type (poultry), the user can refine the results using four other types of metadata.

This view allows me to reduce the set of recipes along any of the other metadata facets. It also tells me how many recipes will result if I refine the current metadata facet (Course/Meal) according to one if the terms in its subhierarchy. If I select Hors d’Oeuvres, the view of Figure 2 results.

Note that I have created this combination of categories on the fly. I could have started with Course Dish > Hors d’Oeuvres, and then refined by Main Ingredient > Poultry and ended up at the same point.

Now I can further refine the set of recipes by selecting a preparation type, the subhierarchies for which are shown above. Alternatively, I can select Cuisine and the system will show the same set of 44 chicken Hors d’Oeuvres recipes according to cuisine type, such as Caribbean, Italian, Low Fat, and Kid-Friendly.

The site also allows search over document subsets. In the example above, I can search over the 44 chicken appetizer recipes to isolate those that include avocado or some other ingredient. Search is also allowed over the entire dataset. However, the results of search are shown as a long unordered list of recipes, and thus loses the benefits of the browsing interface. There is, however, an advanced search form that allows the user to select sets of main ingredients along with the other metadata types. It suffers in comparison to the browsing facility, however, in not helping users avoid empty or large results sets.



Browse > Poultry > Hors d'Oeuvres

Refine by: Preparation | **Cuisine** | **Season/Occasion**

Advance (4)
Bake (8)
Barbecue (4)
Broil (3)
Fry (2)
Grill (8)
Marinade (4)
No Cook (1)
Quick (6)
Roast (3)
Saute (6)
Slow Cook (1)

1 - 15 of 44 | **Next>**

BRANDIED CHICKEN LIVER PATE
Gourmet March 1996

BUFFALO WINGS
Epicurious January 1998
...

Figure 2: Result of revising the results of Figure 1 by selecting the Hors d'Oeuvres metadata type.

This interface supports information seeking for several different kinds of recipe-related tasks, including, e.g., "Help me find a summer pasta," (ingredient type with event type), "How can I use an avocado in a salad?" (ingredient type with dish type), and "How can I bake sea-bass" (preparation type and ingredient type). It does not support other tasks such as menu planning and organizing by customer reviews.

2.4 Example: Yahoo

The epicurious site provides a nice example of how to incorporate metadata into the search process, acting as a kind of dynamically-determined hyperlink. This is different from a setup like the directory structure at Yahoo, in which the paths are determined in advance. The Yahoo directory does combine certain types of metadata – most notably Region types are intermixed with other categories – but usually only up to two types are combined, and the combinations are not tailed to what the user wants to see. For example, to find UC Berkeley following links, the links that I clicked on were College and University > Colleges and Universities > United States > U > University of California > Campuses > Berkeley. This makes use of crosslinks (symbolic links), so the official category once I finally see a link for UC Berkeley is: U.S. States > California > Education > College and University > Public > University of California > Campuses. After clicking on Berkeley, the new category label that is actually associated with the information about UC Berkeley is U.S. States > California > Cities > Berkeley > Education > College and University > Public > UC Berkeley. This is an entirely different path than the one I traversed via hyperlinks. To handle the fact that most categories are best reached by multiple kinds of metadata, Yahoo does a great deal of crosslinking that causes the actual category labels traversed to change beneath the user. A system that lets the user choose which kinds of metadata to view next should be more effective.

By contrast to this clumsy use of metadata, Yahoo has a nice way of dynamically linking metadata in its restaurant selection site. At the top level it presents links that aid in tasks involving selection of restaurants, including maps and telephone directories. Before any searching can take place, the user must navigate a region metadata hierarchy to select a city. The user can then either select a cuisine link or issue a query over restaurant names or cuisine types, resulting in a list of restaurants that meet these criteria. After selecting a hyperlink for a particular restaurant, the user sees a summary of information about the restaurant which includes a set of links grouped under the label of "Find Nearby". These links include movies, bars and clubs, and cafes that can be found in the geographic neighborhood of the selected restaurant. (Researchers are also providing this type of functionality [4].) There is also a link labeled "More A&E" (A&E indicates Arts and Entertainment, the supercategory for Restaurants and for Movies), but this breaks the conceptual model by showing a listing of all entertainment choices in the city, not those limited just to be near the selected restaurant.

In essence, Yahoo has assumed that many of those users searching the restaurant collection are actually engaged in a larger task which can be paraphrased as the stereotypical "find evening entertainment" task: dinner and a nearby movie. Two metadata facets are combined here: the region facet and the entertainment facet, and within this, two subhierarchies with the entertainment facet have been linked together – restaurant (with a cuisine attribute) and movies.

2.5 Integrating Search

The epicurious site does a nice job of dynamically suggesting metadata to help the user reduce the set of documents in an organized manner, making use of information previews to show how many documents would result after each choice, and allowing the user to easily back up to earlier states in the search process by clicking on the hyperlinks indicating the path taken so far. However, the interface does not interweave the search results into the category structure, and a straightforward improvement would be to organize the search results according to the same category metadata layout that is used for the browsing interface.

However, a large text collection such as the MedLine collection of biomedical abstracts is more difficult to search and organize than something like recipes. Additional facilities may be needed to apply this kind of dynamic metadata interface to something as complex and voluminous as medical text, and information retrieval-style ranking is probably necessary along with keyword search to help sort through results. Furthermore, the metadata is in some cases more hierarchical than in the recipe example, and more types of metadata are available, so only a subset should be shown at any given time. The system should dynamically determine which *types* of metadata to show, based on what the user has done so far and their past history (this idea has been pursued in other contexts [10, 13]). For example, a clinician prescribing medications for a patient may want to be able to always see categories associated with this patient's particular allergies.

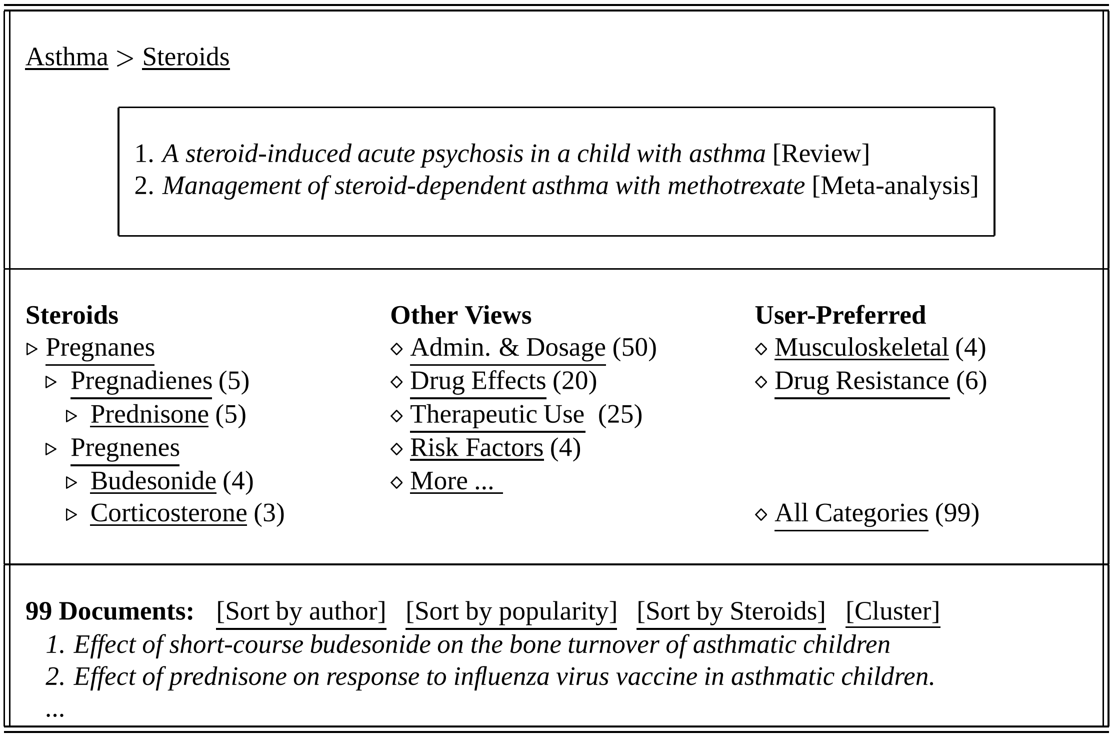
2.6 Example: BioMedical Text

Consider the following example. Say a medical clinician named Dr. Care needs to find information about the use of cortisone shots as a treatment for asthma for a particular patient. Using our proposed system, Dr. Care can begin either by selecting an initial category label or by typing in some terms directly. Assume she already knows the MeSH category labels for the high-level concepts *Asthma* and *Steroids* but does not want to have to remember the category labels for more specific terms. By specifying these labels directly, the equivalent of a conjunctive Boolean search is run over the collection.[4] This returns 99 articles, which have a total of 2000 MeSH subject headings, 577 of which are unique.

Figure 3 shows an example of what an interface that incorporates the ideas discussed above might look like. The system provides Dr. Care with a way to get started in dealing with this large result set. It indicates the path taken to get to this point, the titles of two of the articles, suggestions for next steps below this, and the full document list at the bottom.

At the top is shown a hyperlinked path indicating the two choices made so far. The links allow the user to easily go back to an earlier stage in the navigation process; by selecting the Asthma link, Dr. Care would see the

[4] The data for this example was generated by running queries over the Medline/Healthstar database for 1995-1999 as of July, 1999, using the California Digital Library interface to this system. The collection contains article abstracts from 8,400 journals. Article information was downloaded and processed by hand in order to obtain the numbers. In some cases the details are simplified for expository purposes.



Asthma > Steroids

1. *A steroid-induced acute psychosis in a child with asthma* [Review]
2. *Management of steroid-dependent asthma with methotrexate* [Meta-analysis]

Steroids	**Other Views**	**User-Preferred**
▷ Pregnanes	◇ Admin. & Dosage (50)	◇ Musculoskeletal (4)
▷ Pregnadienes (5)	◇ Drug Effects (20)	◇ Drug Resistance (6)
▷ Prednisone (5)	◇ Therapeutic Use (25)	
▷ Pregnenes	◇ Risk Factors (4)	
▷ Budesonide (4)	◇ More ...	
▷ Corticosterone (3)		◇ All Categories (99)

99 Documents: [Sort by author] [Sort by popularity] [Sort by Steroids] [Cluster]

1. Effect of short-course budesonide on the bone turnover of asthmatic children
2. Effect of prednisone on response to influenza virus vaccine in asthmatic children.
...

Figure 3: Sketch of a proposed method for browsing and searching biomedical text.

documents associated with this term, independent of Steroids.

Below this appears a box showing a few selected document titles. These are a review article and a meta-analysis, which are appropriate given the high-level terms used so far in the navigation process. Other reviews exist in this result set but talk about specific steroids and so are not shown at this point. This illustrates another important idea: a search interface should match the level of generality of the retrieved documents to the generality of the current navigation state. For example, at the early stages of the search, overview articles should be shown, but as the search becomes more specific, so should the documents.

Below the title box are shown lists of categories. The lefthand side shows a subset of the Steroids portion of the chemicals facet of MeSH. Only those categories within the Steroids subtree that occur significantly within the 99 documents of this result set are shown. The steroid metadata hierarchy implicitly shows which branch of the steroid family they occupy, along with the number of documents in this subcollection that refer to the particular steroid. Thus the user sees a preview of what would happen if they added any of these terms to the query; the results would be reduced dramatically. But rather than actually issuing a query, the user can simply click on the category label, thereby navigating to a portion of the search space that contains the steroid term conjoined with the other terms used so far. The user can subsequently easily back up to the current state by following the hyperlinked path at the top of the screen.

The righthand column shows how user-preferred categories can be integrated into the interface. Say Dr. Care is concerned about musculoskeletal and drug resistance issues for this particular patient. These categories will appear on all views of the results collection, along with a preview of how many documents are in the set. These preferences could be specified directly by the user, inferred from previous selections, or based on citation structure or popularity in terms of page accesses. Finally, the user can elect to see all categories that correspond to documents in the current set, if they prefer to override the system's organization facilities.

The bottommost portion of the display allows Dr. Care to scan the retrieval results directly, and also allows her to reorder the titles in various ways. Sorting by popularity takes into account how often other users have viewed the documents. It also provides a “Sort by Steroids” option. This appears because Steroids is the most recently chosen category (which is also why it appears in the lefthand category column). The resulting ordering would make use of the structure of the Steroids subhierarchy to group documents with similar steroids together. Finally, users can invoke a clustering option to show a textual or graphical display of documents according to their overall commonality [9].

As these examples from disparate fields show, there are commonalities in supporting effective search strategies between domains. In our new research project, FLAMENCO, we are investigating these ideas of FLexible information Access using MEtadata in Novel COmbinations. Our end goal is to develop a general methodology for specifying task-oriented search interfaces across a wide variety of domains and tasks. We suggest that rich, faceted metadata be used in a flexible manner to give users information about where to go next, and to have these suggestions and hints reflect the users’ individual tasks.

3 Other Approaches to Site Search

3.1 Specialized Interfaces

Another way to improve web site search is to create a specialized interface that takes the structure of the information on the site into account. One of our research projects applies a variation of this idea in an attempt to improve Intranet search. Intranets contain the information associated with the internal workings of an organization, and our system, called Cha-Cha, organizes web search results in such a way as to reflect the underlying structure of the organization. An “outline” or “table of contents” is created by first recording the shortest paths in hyperlinks from root pages to every page within the web intranet. After the user issues a query, these shortest paths are dynamically combined to form a hierarchical outline of the context in which the search results occur [3]. For example, hits on the query “earthquake” will be shown to fall within the mechanical engineering department, a science education project, and administrative pages that indicate emergency evacuation plans.

This interface has been deployed as the UC Berkeley site search engine for the last two years, receiving on average about 200,000 queries a month. Based on user interviews and surveys, it is sometimes quite useful to see the context in which the search hit occurred, especially when the query does not produce a good hit. On the other hand, the extra information can be overwhelming and unnecessary if the search is relatively straightforward. Furthermore, an Intranet’s structure does not always reflect the user’s task; often some other kind of organization would be more appropriate. For example, a user trying to find out about research on the effects of old-growth logging probably cares about the different kinds of logging under consideration, but not which university department the results came from.

Another example of a search interface that follows the structure of the information is the CiteSeer interface [7] which focuses on a *type* of information – scientific references – rather than a content domain. The search structure reflects the structure of the underlying information: citations have hyperlinks to other papers by the same authors, and to paragraphs of text of articles in which the target article is cited. Search results of aggregates of users are exploited both for ranking and for producing informative statistics, such as how many articles cite a particular author’s papers. The interface shows special links that would not make sense in general search, and are tailored to what people searching research literature are interested in.

3.2 Question Answering

I believe a number of other approaches to web search will flourish as time goes on. These include rent-an-expert sites, where users are matched up with people who have expertise in a field and answer their questions for a fee.

Some services connect clients and experts via the phone, avoiding the necessity of typing and having the potential to spread easily to use via mobile devices (Keen.com and Exp.com are two examples).

Systems to automate question answering are also improving, and these will become important supplements for organizations' web sites, to handle customer questions more quickly and cheaply. Rather than generating an answer from scratch, these systems attempt to link a natural language query to the most pertinent sentence, paragraph, or page of information that has already been written. They differ from standard search engines in that they make use of the structure of the question and of the text from which the answers are drawn. For example, a question about what to do when a disk is full needs to be linked to an answer about compressing files or buying new disk. A user asking "Why does my computer keep hanging?" wants to find information about how to avoid this situation, whereas a user asking "How do I get my computer to print?" wants information about how to bring the situation about. The syntax of the question, as well as the content words, determines the kind of acceptable answer.

Question answering in a limited domain can be very powerful but it is much harder in a broad domain. Sophisticated question answering has not yet gotten far in web search aside from the well-known example of AskJeeves, in which question types are manually linked in advance to specific answer pages. However, researchers [2, 8] and companies (such as AnswerLogic and InQuizit) are developing domain-specific natural language processing algorithms and lexical resources that should greatly improve automated question answering in the next two to three years.

Technologists are becoming highly interested in what might be considered "real-world" metadata. The thinking is that a query of "Where is a good Mazda mechanic?" should automatically take note of the local time and location of the question asker, in order to make suggestions of car repair places that are both nearby and open at the time the question is asked. The need for such context-aware questions answering systems can be expected to grow along with the demand for networked mobile devices.

4 Integration with General Web Search

Returning to the original topic of this essay, what is the role of general search engines in the framework proposed above? General search engines should evolve to direct people to task-oriented solutions, instead of collection-oriented solutions as the modus operandi today. In other words, search engines will need to match user requests to task descriptions. One step in this direction is to interpret multi-word queries in terms of their implicit task. For example, to use document genre to determine which results to return. As a straightforward example, a query on "review" alongside another term such as the name of a play should bring back sites with theatre reviews. Currently the unstated default for most queries, for ad servers at least, is that the user's task is to buy something. Eventually I envision task directories supplementing directories, and search engines providing search over these descriptions.

Acknowledgements
I would like to thank Jan Pedersen for many helpful conversations about web search, Doug Cook for detailed information about Inktomi's current search results statistics, and Eric Brewer, Ame Elliott, Dan Glaser, Luis Gravano, Jason Hong, Rashmi Sinha, and Hal Varian for helpful comments on this article.
This research was supported an NSF CAREER grant, NSF9984741.

References

[1] Brian Amento, Loren Terveen, and Will Hill. Does 'authority' mean quality? Predicting expert quality ratings on web documents. In *Proceedings of the 23rd Annual International ACM/SIGIR Conference*, pages 296–303, Athens, Greece, 2000.

[2] Claire Cardie, Vincent Ng, David Pierce, and Chris Buckley. Examining the role of statistical and linguistic knowledge sources in a general-knowledge question-answering. In *Proceedings of the Sixth Applied Natural Language Processing Conference (ANLP-2000)*, pages 180–187. Association for Computational Linguistics/Morgan Kaufmann, May 2000.

[3] Michael Chen, Marti A. Hearst, Jason Hong, and James Lin. Cha-cha: A system for organizing intranet search results. In *Proceedings of the 2nd USENIX Symposium on Internet Technologies and Systems*, Boulder, CO, October 11-14 1999.

[4] Junyan Ding, Luis Gravano, and Narayanan Shivakumar. Computing geographical scopes of web resources. In *Proceedings of the Twenty-sixth International Conference on Very Large Databases (VLDB'00)*, Sept 2000.

[5] George W. Furnas. Effective view navigation. In *Proceedings of ACM CHI 97 Conference on Human Factors in Computing Systems*, volume 1 of *PAPERS: Information Structures*, pages 367–374, 1997.

[6] Fredric Gey, Hui-Min Chen, Barbara Norgard, Michael Buckland, Youngin Kim, Aitao Chen, Byron Lam, Jacek Purat, and Ray Larson. Advanced search technologies for unfamiliar metadata. In *Meta-Data '99 Third IEEE Meta-Data Conference*, Bethesda, MD, April 1999.

[7] C. Lee Giles, Kurt Bollacker, and Steve Lawrence. CiteSeer: An automatic citation indexing system. In *Digital Libraries 98 - The Third ACM Conference on Digital Libraries*, pages 89–98, Pittsburgh, PA, June 1998.

[8] Sanda Harabagiu, Marius Pasca, and Steven Maiorano. Experiments with open-domain textual question answering. In *Proceedings of the COLING-2000*. Association for Computational Linguistics/Morgan Kaufmann, Aug 2000.

[9] Marti A. Hearst, David Karger, and Jan O. Pedersen. Scatter/gather as a tool for the navigation of retrieval results. In Robin Burke, editor, *Working Notes of the AAAI Fall Symposium on AI Applications in Knowledge Navigation and Retrieval*, Cambridge, MA, November 1995. AAAI.

[10] Haym Hirsh, Chumki Basu, and Brian D. Davison. Learning to personalize. *Communications of the ACM*, 43(8), Aug 2000.

[11] Mark Hurst. Holiday '99 e-commerce. http://www.creativegood.com, Sept 1999.

[12] Jon Kleinberg. Authoritative sources in a hyperlinked environment. In *Proceedings of the 9th ACM-SIAM Symposium on Discrete Algorithms*, 1998.

[13] Henry Lieberman. Letizia: an agent that assists web browsing. In *Proceedings of 14th International Joint Conference on Artificial Intelligence (IJCAI-95)*, pages 924–929, 1995.

[14] Gary Marchionini. *Information Seeking in Electronic Environments*. Cambridge University Press, 1995.

[15] Karen Markey, Pauline Atherton, and Claudia Newton. An analysis of controlled vocabulary and free text search statements in online searches. *Online Review*, 4:225–236, 1982.

[16] Peter Pirolli. Computational models of information scent-following in a very large browsable text collection. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*, pages 3–10, Vancouver, Canada, May 1997. ACM.

[17] Catherine Plaisant, Ben Shneiderman, Khoa Doan, and Tom Bruns. Interface and data architecture for query preview in networked information systems. *ACM Transactions on Information Systems*, 17(3):320–341, 1999.

[18] Tara Scanlon, Will Schroeder, Richard Danca, Nina Gilmore, Matthew Klee, Lori Landesman, Amy Maurer, Paul Sawyer, and Jared Spool. *Designing Information-Rich Web Sites*. User Interface Engineering, 1999.

[19] Chris Sherman. Inktomi inside. http://websearch.about.com/internet/websearch/library/weekly/aa041900a.htm, April 2000.

[20] Jared Spool. *Web Site Usability: A Designer's Guide*. Morgan Kaufmann, 1998.

[21] Danny Sullivan. NPD search and portal site study. http://searchenginewatch.internet.com/reports/npd.html, July 6 2000. NPD's URL is http://www.npd.com.

ICDE 2001

in the Heart of Europe

The 17th International Conference on Data Engineering

April 2–6, 2001
Heidelberg, Germany

The 17th International Conference on Data Engineering (ICDE 2001) will be held in a beautiful old town – Heidelberg. This is at the center of Germany's most active high-tech region where you find world renowned companies, numerous software startups, many world-class research centers such as the ABB Research Center, the Deutsche Telekom Research and Development Center, the European Media Laboratory, the European Molecular Biology Laboratory, the Fraunhofer Gesellschaft Institute for Graphical Data Processing, the German Center for Cancer Research, the GMD National Center for Information Technology, the Computer Science Research Center, the Karlsruhe Research Center, and a number of universities with strong database research groups (amongst others Darmstadt University of Technology, International University in Germany, University of Karlsruhe, University of Mannheim and not far away the University of Stuttgart). Such combination of strong industry, ground breaking research institutions, economic prosperity, and a beautiful host town provide an ideal environment for a conference on Data Engineering.

With tutorials, panels and industrial program.

Website & Registration at
http://www.congress-online.de/ICDE2001

Early Registration until the 2nd of March 2001

Topics Include:

XML, METADATA, and SEMISTRUCTURED DATA
DATABASE ENGINES & ENGINEERING
QUERY PROCESSING
DATA WAREHOUSES, DATA MINING, AND KNOWLEDGE DISCOVERY
ADVANCED IS MIDDLEWARE
SCIENTIFIC AND ENGINEERING DATABASES
EXTREME DATABASES
E-COMMERCE and E-SERVICES
WORKFLOW and PROCESS-ORIENTED SYSTEMS
EMERGING TRENDS
SYSTEM APPLICATIONS AND EXPERIENCE

Conference Officers

General Chairs:	
Andreas Reuter	European Media Laboratory and International University in Germany
David Lomet	Microsoft Research, USA
Program Co-chairs:	
Alex Buchmann	University of Darmstadt, Germany
Dimitrios Georgakopoulos	Telcordia Technologies, USA
Panel Program Chair:	
Erich Neuhold	GMD-IPSI, Germany
Tutorial Program Chair:	
Guido Moerkotte	University of Mannheim, Germany
Eric Simon	INRIA, France
Industrial Program Co-chairs:	
Peter Lockemann	University of Karlsruhe, Germany
Tamer Ozsu	University of Alberta, Canada
Steering committee liaison:	
Erich Neuhold	GMD-IPSI, Germany
Marek Rusinkiewicz	MCC, USA
Organizing Chair:	
Isabel Rojas	European Media Laboratory, Germany
Demos & Exhibits:	
Wolfgang Becker	International University in Germany
Andreas Eberhart	
Local Arrangements:	
Bärbel Mack	European Media Laboratory, Germany
Claudia Spahn	
Public Relations:	
Peter Saueressig	European Media Laboratory, Germany

IEEE
COMPUTER
SOCIETY

Sponsored by the
IEEE Computer Society

Financial Support: Special support for travel expenses and conference fees will be available for participants from Eastern Europe.

Non-profit Org.
U.S. Postage
PAID
Silver Spring, MD
Permit 1398

IEEE Computer Society
1730 Massachusetts Ave, NW
Washington, D.C. 20036-1903