Steve Lawrence. *Context in Web Search*, IEEE Data Engineering Bulletin, Volume 23, Number 3, pp. 25–32, 2000.

Context in Web Search

Steve Lawrence
NEC Research Institute, 4 Independence Way, Princeton, NJ 08540
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 However, the major web search engines have significant limitations – they are often out-of-date, they only index a fraction of the publicly indexable web, they do not index documents with authentication requirements and many documents behind search forms,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.

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. For example, 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 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 specialized search engines already exist (see *www.completeplanet.com* and *www.invisibleweb.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 *researchindex.org*), 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 index information on 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 database, 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 plaintext 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-László 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. Garcı́a-Molina, L. Gravano, and N. Shivakumar. Exploiting goegraphical location information of web pages. In *Proceedings 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 Garcı́a-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 *Proceedings 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 Aritifical 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.