Searching the Workplace Web

Ronald Fagin Ravi Kumar Kevin S. McCurley
Jasmine Novak D. Sivakumar
John A. Tomlin David P. Williamson

IBM Almaden Research Center
650 Harry Road
San Jose, CA 95120

ABSTRACT

The social impact from the World Wide Web cannot be underestimated, but technologies used to build the Web are also revolutionizing the sharing of business and government information within *intranets.* In many ways the lessons learned from the Internet carry over directly to intranets, but others do not apply. In particular, the social forces that guide the development of intranets are quite different, and the determination of a "good answer" for intranet search is quite different than on the Internet. In this paper we study the problem of intranet search. Our approach focuses on the use of *rank aggregation*, and allows us to examine the effects of different heuristics on ranking of search results.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

General Terms

Algorithms, Measurement, Experimentation

1. INTRODUCTION

The corporate intranet is an organism that is at once very similar to and very unlike the Internet at large. A well-designed intranet, powered by a high-quality enterprise information portal (EIP), is perhaps the most significant step that corporations can make—and have made in recent years—to improve productivity and communication between individuals in an organization. Given the business of EIPs and their impact, it is natural that the *search problem* for intranets and EIPs is growing into an important business.

Despite its importance, however, there is little scientific work reported on intranet search, or intranets at all for that matter. For example, in the history of the WWW conference, there appear to have been only two papers that refer to an "intranet" at all in their title [28, 22], and as best we can determine, all previous WWW papers on intranets are case studies in their construction for various companies and universities [18, 22, 12, 4, 21, 29, 6]. It is not surprising that few papers are published on intranet search: companies whose livelihood depends on their intranet search products are unlikely to publish their research, academic researchers generally do not have access to corporate intranets, and most researchers with access to an intranet have access only to that of their own institution.

One might wonder why the problem of searching an intranet is in any way different from the problem of searching the web or a small corpus of web documents (e.g., site search). In Internet search, many of the most significant techniques that lead to quality results are successful because they exploit the reflection of social forces in the way people present information on the web. The most famous of these are probably the HITS [24, 10, 11] and the PageRank [7] algorithms. These methods are motivated by the observation that a hyperlink from a document to another is an implicit conveyance of authority to the target page. The HITS algorithm takes advantage of a social process in which authors create "hubs" of links to authoritative documents on a particular topic of interest to them.

Searching an intranet differs from searching the Internet because different social forces are at play, and thus search strategies that work for the Internet may not work on an intranet. While the Internet reflects the collective voice of many authors who feel free to place their writings in public view, an intranet generally reflects the view of the entity that it serves. Moreover, because intranets serve a different purpose than the Internet at large, the kinds of queries made are different, often targeting a single "right answer". The problem of finding this "right answer" on an intranet is very different from the problem of finding the best answers to a query on the Internet.

In this paper we study the problem of intranet search. As suggested by the argument above, part of such a study necessarily involves observations of ways in which intranets differ from the Internet. Thus we begin with a high-level understanding of the structure of intranets. Specifically, we postulate a collection of hypotheses that shed some light on how—and why—intranets are different from the Internet. These "axioms" lead to a variety of ranking functions or heuristics that are relevant in the context of intranet search.

We then describe an experimental system we built to study intranet ranking, using IBM's intranet as a case study. Our system uses a novel architecture that allows us to easily combine our various ranking heuristics through the use of a "rank aggregation" algorithm [16]. Rank aggregation algorithms take as input multiple ranked lists from the various heuristics and produces an ordering of the pages aimed at minimizing the number of "upsets" with respect to the orderings produced by the individual ranking heuristics.

Rank aggregation allows us to easily add and remove heuris-

WWW2003, May 20–24, 2003, Budapest, Hungary.
ACM 1-58113-680-3/03/0005.

tics, which makes it especially well-suited for our experimental purposes. We argue that this architecture is also well-suited for intranet search in general. When designing a general purpose intranet search tool (as opposed to a search tool for a *specific* intranet), we anticipate the architecture to be used in a variety of environments—corporations, small businesses, governmental agencies, academic institutions, etc. It is virtually impossible to claim an understanding of the characteristics of intranets in each of these scenarios. Therefore, it is crucial that the crafting of the final ranking method allows for a wide variety of ranking heuristics to be used in a "plug-and-play" mode. Each heuristic is typically based on one or more hypotheses about the structure of the intranet; the extent to which a given intranet satisfies a hypothesis should ultimately decide whether or not (and the extent to which) the corresponding ranking heuristic should participate in the final ranking method. We envisage a scenario where the search tool is customized for deployment in a new environment with only a modicum of effort. Later, when we describe our experiments, we point out how we can measure the influence of each ranking function on the final ranking; by understanding the correlation between the influence of a ranking function and the quality of results, an administrator can decide which ranking functions to combine.

While we advocate the use of good aggregation techniques to synthesize a robust ranking method from somewhat unreliable heuristics, we have nevertheless tried to be quite objective about which ranking heuristics we think will be fairly general, avoiding ones that are very specific to the intranet that we conducted our experiments on. The more important message here is that our architecture achieves a de-coupling of the two logical aspects—selection of ranking heuristics and the synthesis of the ranking method.

To summarize, we believe that our two-phase approach—identifying a variety of ranking functions based on heuristic and experimental analysis of the structure of an intranet, and a rank aggregation architecture for combining them—is a natural and logical choice that leads to a convenient, customizable, and robust search architecture for intranets.

The rest of the paper is structured as follows. In Section 2, we present our theses on the structure of intranets. In Section 3 we describe the architecture of the components in our search prototype along with the factors that we examined for ranking documents. A key paradigm of our approach is rank aggregation, which is discussed further in Section 4. This is followed in Section 5 with the presentation of the results of our experiments, together with an analysis. Finally, in Section 6 we outline some concluding thoughts and avenues for future work.

2. INTRANETS VS. THE INTERNET

While intranets have the same basic overall structure as the Internet, namely a collection of documents connected by hyperlinks, there are several qualitative differences between intranets and the Internet. These differences stem from the fact that the underlying content generation processes for intranets and for the Internet are fundamentally different.

While the Internet tends to grow more democratically, content generation in intranets often tends to be autocratic or bureaucratic; this is more or less a consequence of the fact that a fairly centralized process exists whereby a small number of individuals (employees, IT contractors, etc.) are *assigned* the responsibility of building/maintaining pieces of intranets, and there is much careful review and approval (if not censorship) of content. Documents are often designed to be informative (in a fairly minimal sense), and are usually not intended to be "interesting" (e.g., rich with links to related documents). In fact, there is often no incentive for the content creator to strive to design a particularly good web page that attracts traffic from the users of the intranet. This is quite the opposite of the Internet, where being in the top 10 results for certain queries has a direct effect on the traffic that a page receives, which, in turn, may be correlated with revenue. Similarly, in intranets, there is no incentive for creating and frequently updating hubs on specific topics. A consequence of the above discussion is that several ranking functions based on link analysis (e.g., HITS [10] and PageRank [7]) tend to be less effective for intranets. This discussion leads to our first axiom.

Axiom 1. Intranet documents are often created for simple dissemination of information, rather than to attract and hold the attention of any specific group of users.

A fascinating aspect of the Internet is that there is usually a large number of documents that are relevant to any reasonably natural query. For example, if we make the rather uncommon query "How do I remove tree sap from my car?" in an Internet search engine, we will see that there is a wide variety of pages that will provide advice (vendors of cleaning products, helpful hints specialists, random chroniclers who have experienced the situation before, etc.). On the other hand, in intranets, even fairly common queries like "I forgot my intranet password; how do I reset it?" or "How do I place a conference call from my office telephone?" tend to have a small number of relevant pages (usually just one). Thus, the search problem on the Internet is, in some sense, easier than on intranets—any reasonable subset of pages on the topic would be considered good answers for the tree sap query, but unless an intranet search engine hones in on the exact page for the queries of the kind mentioned, it would be considered poor. To make it even trickier, these pages often are not distinguished in any special way that makes them easily identifiable as relevant for certain queries. This yields our second axiom, which also implies that a good ranking algorithm needs to be based on several clues matching the query terms and web pages.

Axiom 2. A large fraction of queries tend to have a small set of correct answers (often unique), and the unique answer pages do not usually have any special characteristics.

Not all is negative about intranets, though. One of the biggest problems with Internet search is spam; intranets, on the other hand, tend to be significantly less prone to devious manipulation of web pages with an intent to improve their ranks. Consequently, several ranking heuristics that are extremely unreliable on the Internet turn out to be quite safe and useful in intranets. The use of anchortext information is a perfect example of this phenomenon. For example, consider the ranking heuristic that says "rank page P highly for query q if the words in q are part of the anchortext for a hyperlink that points to page P." This is a dangerous heuristic on the Internet, and is prone to be misled by spamming. In intranets, however, we may expect this heuristic to be fairly reliable. Other examples are the use of in-degree, URL depth, etc. This gives our third axiom.

Axiom 3. Intranets are essentially spam-free.

Finally, there is yet another set of features that distinguish intranets from the Internet; these arise from the internal architecture of the intranet, the kind of servers used, document types specific to an organization (e.g., calendars, bulletin boards that allow discussion threads), etc. For example, the IBM intranet contains many Lotus Domino servers that present many different views of an underlying document format by exposing links to open and close individual sections. For any document that contains sections, there are URLs to open and close any subset of sections in an arbitrary order, which results in exponentially many different URLs that correspond to different views of a single document. This is an example of a more general and serious phenomenon: large portions of intranet documents are not designed to be returned as answers to search queries, but rather to be accessed through portals, database queries, and other specialized interfaces. This results in our fourth axiom.

Axiom 4. Large portions of intranets are not search-engine-friendly.

There is, however, a positive side to the arguments underlying this axiom: adding a small amount of intelligence to the search engine could lead to dramatic improvements in answering fairly large classes of queries. For example, consider queries that are directory look-ups for people, projects, etc., queries that are specific to sites/locations/organizational divisions, and queries that can be easily targeted to specific internal databases. One could add heuristics to the search engine that specifically address queries of these types, and divert them to the appropriate directory/database lookup.

2.1 Intranets vs. the Internet: structural differences

Until now, we have developed a sequence of hypotheses about how intranets differ from the Internet at large, from the viewpoint of building search engines. We now turn to more concrete evidence of the structural dissimilarities between intranets and the Internet. One of the problems in researching intranet search is that it is difficult to obtain unbiased data of sufficient quantity to draw conclusions from. We are blessed (beleaguered?) with working for a very large international corporation that has a very heterogeneous intranet, and this was used for our investigations. With a few notable exceptions, we expect that our experience from studying the IBM intranet would apply to any large multinational corporation, and indeed may also apply to much smaller companies and government agencies in which authority is derived from a single management chain.

The IBM intranet is extremely diverse, with content on at least 7000 different hosts. Because of dynamic content, the IBM intranet contains an essentially unbounded number of URLs. By crawling we discovered links to approximately 50 million unique URLs. Of these the vast majority are dynamic URLs that provide database access to various underlying databases. IBM's intranet has nearly every kind of commercially available web server represented on the intranet as well as some specialty servers, but one feature that distinguishes IBM from the Internet is that a larger fraction of the web servers are Lotus Domino. These servers influence quite a bit of the structure of IBM's intranet, because the URLs generated by Lotus Domino servers are very distinctive and contribute to some of the problems in distinguishing duplicate URLs. Whenever possible we have made an effort to identify and isolate their influence. As it turns out, this effort led us to an architecture that is easily tailored to the specifics of any particular intranet.

From among the approximately 50 million URLs that were identified, we crawled about 20 million. Many of the links that were not crawled were forbidden by robots, or were simply database queries of no consequence to our experiments. Among the 20 million crawled pages, we used a duplicate elimination process to identify approximately 4.6 million non-duplicate pages, and we retained approximately 3.4 million additional pages for which we had anchortext.

The indegree and outdegree distributions for the intranet are remarkably similar to those reported for the Internet [8, 14]. The connectivity properties, however, are significantly different. In Figure 1, SCC refers to the largest strongly connected component of the underlying graph, IN refers to the set of pages not in the SCC from which it is possible to reach pages in the SCC via hyperlinks, OUT refers to the set of pages not in the SCC that are reachable from the SCC via hyperlinks, and P refers to the set of pages reachable from IN but that are not part of IN or SCC. There is an assortment of other kinds of pages that form the remainder of the intranet. On the Internet, SCC is a rather large component that comprises roughly 30% of the crawlable web, while IN and OUT have roughly 25% of the nodes each [8]. On the IBM intranet, however, we note that SCC is quite small, consisting of roughly 10% of the nodes. The OUT segment is quite large, but this is expected since many of these nodes are database queries served by Lotus Domino with no links outside of a site. The more interesting story concerns the component P, which consists of pages that can be reached from the seed set of pages employed in the crawl (a standard set of important hosts and pages), but which do not lead to the SCC. Some examples are hosts dedicated to specific projects, standalone servers serving special databases, and pages that are intended to be "plain information" rather than be navigable documents.

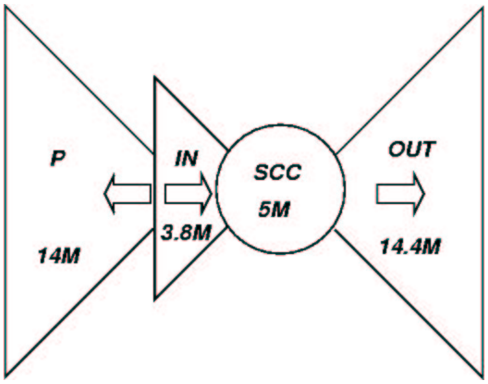

Figure 1: Macro-level connectivity of IBM intranet

One consequence of this structure of the intranet is the distribution of PageRank [7] across the pages in the intranet. The PageRank measure of the quality of documents in a hyperlinked environment corresponds to a probability distribution on the documents, where more important pages are intended to have higher probability mass. We compared the distribution of PageRank values on the IBM intranet with the values from a large crawl of the Internet. On the intranet, a significantly larger fraction of pages have high values of PageRank (probability mass in the distribution alluded to), and a significantly smaller fraction of pages have mid-range values of PageRank. This suggests that PageRank might not be an effective discriminator among intranet pages—it can clearly distinguish between the very important pages and the others, but probably does not offer finer distinctions among the latter.

Another manifestation of this phenomenon is the following. The precise definition of PageRank implies, for example, that a large strongly connected component is likely to have large total probability mass. Thus the SCC on the Internet is where most of the probability mass is concentrated. On the IBM intranet, however, we observe the following distribution: only about 5% of the probability mass is concentrated in the SCC, about 37% of the mass is in OUT, about 30% is in the segment P, and about 3% is in IN.

3. SYSTEM ARCHITECTURE

In this section we describe our research prototype for intranet search. The search system has six identifiable components, namely a crawler, a global ranking component, a duplicate elimination component, an inverted index engine, a query runtime system, and a result markup and presentation system. We will describe each of these, with special attention in the next section to the query runtime system, since it is the primary focus of our investigation.

The crawler has a few minor characteristics that were added to facilitate crawling of an intranet specifically, but is otherwise a fairly standard crawler that can scale up to the Internet. The crawler keeps records for every URL in a DB2 database, and produces a structured record for each crawled URL that stores metadata about the pages. These records are later used in the result markup and presentation system when results are presented to the user. In order to reduce the number of near-duplicate pages fetched from Lotus Domino servers, the crawler was also tailored to perform aggressive URL canonicalization on Lotus Domino URLs.

Duplicate elimination is an important optimization to make for both intranet and Internet ranking, since there are often many copies of documents that appear, and these copies can complicate the ranking process as well as consume resources for crawling, storing, indexing, and querying. The technique that we use is to compute a shingle [9] on the content for each page, group the URL records by the value of the shingle, and then from all the pages that have an identical shingle value, choose a favored representative to use in the indexing process. The selection criterion for the primary representative uses the length of the URL, the content size, the static rank of the page, and a few minor heuristics that are specific to the structure of mirrors inside IBM.

In the global ranking phase, we computed several tables that allow us to order web pages statically (i.e., independent of a query). The techniques that we use include PageRank [7] as well as indegrees. These are used in the final phase for calculation of "best results" in response to queries, and are described in Section 4.

The indexing phase is similar to any other indexing phase. One minor difference is that we index only the primary copies of documents. The major difference between our approach and others is that we constructed multiple separate indices that we consult independently. At the present time we are using three indices:

Content index. This is the traditional means by which text documents are indexed. We tokenized the documents to produce a sequence of terms, some of which have attributes such as "title" or "heading". This index contains terms from 4,672,819 non-duplicate documents.

Title index. We extracted titles from all HTML files and built a separate index for these, in part to test the hypothesis that for most queries there is a document whose title contains all of the query terms. In cases where titles did not exist, we substituted the first heading. We also added additional tokens for keywords and descriptions that appear in `META` tags. This is one area where we differ from Internet indexing, where the opportunities for incorporating spam into `META` tags has diminished their appeal for indexing. On an intranet the spam problem is largely absent (Axiom 3), although we observed a related problem where authors would often copy a template from another page and edit the contents without changing the tags in order to preserve the common look and feel of an intranet. This index contains terms from approximately 4,343,510 non-duplicate documents.

Anchortext index. Anchortext is well-known to be valuable in the context of web search [11, 5], and has also been recently used to classify documents [19]. For us, anchortext is defined as the text that appears within the bounds of a hypertext link; thus, we chose not to include pre- and post-text surrounding the link. For each document we take all of the anchortext for links to that document (and its copies), and concatenate it all together. We then index this as a virtual version of the original target document. This index contains terms for 7,952,481 documents.

The dictionaries for these indices are not shared. The title index dictionary contained 738,348 terms, the anchortext index dictionary contained 1,579,757 terms, and the content index dictionary contained 13,850,221 terms. This reinforces the intuition that the variety of terms used in anchortext is not substantially more diverse than the language used in titles, but is considerably less diverse than what is used in content. This is one reason why anchortext is very effective in identifying "home pages", even for commonly used terms.

4. RANK AGGREGATION

The primary innovation in our work arises in the method by which we retrieve and rank results. Our approach is to use *rank aggregation* upon a collection of indices and ranking methods. The use of rank aggregation is motivated in part because it allows us to build a flexible system in which we can experiment with the results of different factors. In addition, rank aggregation holds the advantage of combining the influence of many different heuristic factors in a robust way to produce high-quality results for queries.

4.1 Rank aggregation algorithms

A rank aggregation algorithm [16] takes several ranked lists, each of which ranks part of a collection of candidates (web pages), and produces an ordering of all the candidates in the union of the lists; the ordering produced by the algorithm is aimed at minimizing the total number of *inversions*, that is, the total number of "upsets" caused by the final ordering with respect to the initial orderings. In this paper, we employ rank aggregation as a tool to combine the initial rankings of intranet pages (produced by various ranking functions) into an aggregate ranking that is hopefully much better than any one of the constituent ranking functions.

More technically, for the j-th ranked list, let U_j be the set of pages ranked by the list and let $\tau_j(i)$ be the position of page $i \in U_j$ in the list. Let U denote the union of the U_j. Our goal is produce a particular permutation σ of the pages in U. The *Kendall tau distance* $K(\sigma, \pi)$ between two permutations σ and π of U is the number of pairs of items of U which are ordered differently in σ and π, that is, the number of distinct unordered pairs $\{k, l\}$ of members of U such that $\sigma(k) < \sigma(l)$ and $\pi(k) > \pi(l)$ or vice versa. We want to find a permutation σ that is close to the partially ranked lists τ_j. To accommodate the fact that the input lists rank only some of the elements of U, we consider a normalized and modified Kendall tau distance $K_j(\sigma, \tau_j)$. Let $S_j(\sigma, \tau_j)$ be the set of all unordered pairs from U_j that are ordered differently in σ and τ_j; that is, $S_j(\sigma, \tau_j)$ is the set of all unordered pairs $\{k, l\}$ of members of U_j such that either $\sigma(k) < \sigma(l)$ and $\tau_j(k) > \tau_j(l)$ or vice-versa. Then we set $K_j(\sigma, \tau_j) = |S_j(\sigma, \tau_j)|/\binom{|U_j|}{2}$. Our goal is to find a permutation σ on U that minimizes $\sum_j K_j(\sigma, \tau_j)$.

Unfortunately, this quantity is NP-hard to compute for 4 or more lists [16], so we settle for heuristics. The specific aggregation method we use is based on Markov chains, and is called MC_4 in [16]. The pages in U correspond to elements in the chain. If we are currently at page $i \in U$, we pick a page $j \in U$ uniformly at random. If a majority of the input lists that ranked both i and j rank j better than i, then move to j, and otherwise stay at i. We compute the stationary probabilities of this chain, and the permutation of U computed is the list of pages in order of stationary probability, from highest to lowest.

The aggregation method MC_4 is simple to implement, quite efficient, and is fairly robust in the quality of the results it produces. As mentioned in the introduction, rank aggregation allows us to easily add and remove ranking heuristics. In particular, it admits easy and efficient composition of ranking functions when some of them are *static*, or *query-independent*, and some are *dynamic*, or *query-dependent*. We describe some of these in the Section 4.3.

4.2 Relation to other work

Rank aggregation techniques have previously been applied in the context of metasearch, where they have particularly nice properties in their resistance to spam [16]. The Borda method of rank aggregation was applied to the problem of metasearch in [3]. The present work focuses on the problem of combining multiple ranking factors in a single system. Another variation of rank aggregation was discussed in [26], and the general problem of combining different ranking factors in web search remains an active area of research. In [13] and [20] the authors present the problem of combining different ranking functions in the context of a Bayesian probabilistic model for information retrieval. This approach was used with a naive Bayesian independence assumption in [25] and [30] to combine ranking functions on document length, indegree and URL depth as prior probabilities on the documents in a collection. They further suggested a technique to combine different content models for anchor text and content. Due to limitations in their software system, their model was apparently not fully implemented. The Bayesian approach was also applied in [23], where they evaluated an approach to modeling the language of titles in documents.

Rank aggregation methods have been studied extensively in the context of fair elections. The well-known theorem of Arrow [2] shows that there can be no election algorithm that simultaneously satisifes five seemingly desirable properties for elections. Whether these or other properties are desirable for aggregation of search results is an interesting question that is beyond the scope of this paper.

4.3 Construction of ranked result lists

The building blocks for our query runtime system consists of a set of three indices and scoring functions. In response to a query, each of the three indices returns zero or more results. We use a query engine that implements a variation on the INQUERY [1] tf·idf scoring function to extract an ordered list of results from each of the three indices. An important feature of this is that the tf·idf scores are calculated only on the terms within the index, so that anchortext terms are kept separate from terms in the document itself.

In addition to the indices mentioned already, we used a number of heuristics that can be used to compare the quality of web pages. The function of an index is to find documents that are *relevant*, whereas the function of our ranking heuristics is to decide which of the many relevant pages to present in the top k list to a user. Some of these are well-known, and some are ad hoc but motivated by intuition and experience with the IBM intranet.

PageRank. PageRank [7] is generally regarded as providing a very good overall static rank for web pages independent of their content. The computation of PageRank depends only on the link structure between web pages, and is therefore primarily useful for HTML content. There are numerous variations on basic PageRank, and we experimented with several of these in the course of this investigation.

Indegree. Because this measure is vulnerable to spam, it is generally not used on the Internet. We expected it to be strongly correlated to PageRank.

Discovery date. If a crawl is started from a single seed, then the order in which pages will be crawled tends to be similar to a breadth first search through the link graph [27] (the crawl seldom follows pure breadth first order due to crawler requirements to obey politeness and robots restrictions). If we record the time that a page is discovered by a hyperlink, then this sequence of times provides an approximation to the hyperlink graph distance of the page from the root seed of the intranet. With a bit more work we could replace this with the actual "click distance" from the seeds, but we chose to use this simple approximation.

Words in URL. When a URL is shown to be relevant to a query, we give a slight preference to a page that contains a query term as a substring in the URL. Alternatively, we could choose to include tokens for the words in the URL, but the tendency to concatenate words to form URLs (e.g., apachemanual) interferes with this approach. Unlike static ranking algorithms, this one is query-dependent.

URL length. While this cannot be interpreted as a good absolute ranking on web pages, it is motivated by the idea that if two pages contain comparable content, the one with the shorter URL tends to be the more authoritative.

URL depth. This is closely correlated to the length of a URL, and measures the number of slash / characters that appear in the path component in the URL. Between two pages relevant to a query on the same host, we would tend to favor a page near the top of the directory hierarchy, since it would often be more general and have links to pages that are lower in the hierarchy.

Discriminator. This is a "pure hack" to discriminate in favor of certain classes of URLs over others. The favored URLs in our case consisted of those that end in a slash `/` or `index.html`, and those that contain a tilde ∼. Those that were discriminated against are certain classes of dynamic URLs containing a question mark, as well as some Domino URLs. This heuristic is neutral on all other URLs, and is easily tailored to knowledge of a specific intranet.

The exact choice of heuristics is not etched in stone, and our system is designed to incorporate any partial ordering on results. Others that we considered include favoritism for some hosts (e.g., those maintained by the CIO), different content types, age of documents, click distance, HostRank (Pagerank on the host graph), etc.

In our experiments these factors were combined in the following way. First, all three indices are consulted to get three ranked lists of documents that are scored purely on the basis of tf·idf (remembering that the title and anchortext index consists of *virtual* documents). The union of these lists is then taken, and this list is reordered according to each of the seven scoring factors to produce seven new lists. These lists are all combined in rank aggregation. In practice if k results are desired in the final list, then we chose up to $2k$ documents to go into the individual lists from the indices, and discard all but the top k after rank aggregation.

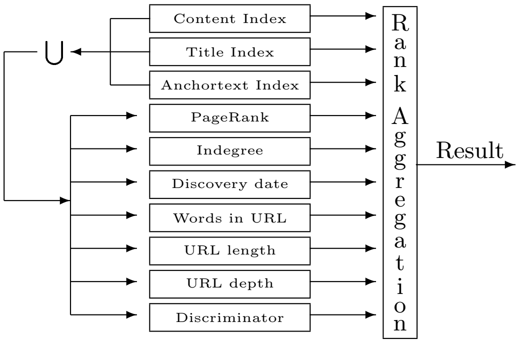

Figure 2: Rank aggregation: The union of results from three indices is ordered by many heuristics and fed with the original index results to rank aggregation to produce a final ordering.

Our primary goal in building this system was to experiment with the effect of various factors on intranet search results, but it is interesting to note that the performance cost of our system is reasonable. The use of multiple indices entails some overhead, but this is balanced by the small sizes of the indices. The rank aggregation algorithm described in Section 4.1 has a running time that is quadratic in the length of the lists, and for small lists is not a serious factor.

5. EXPERIMENTAL RESULTS

In this section we describe our experiments with our intranet search system. The experiments consist of the following steps: choosing queries on which to test the methods, identifying criteria based on which to evaluate the quality of various combinations of ranking heuristics, identifying measures of the usefulness of individual heuristics, and finally, analyzing the outcome of the experiments.

5.1 Ranking methods and queries

By using the 10 ranking heuristics (three directly based on the indices and the seven auxiliary heuristics), one could create 1024 combinations of subsets of heuristics that could be aggregated. However, the constraint that at least one of the three indices needs to be consulted eliminates one-eighth of these combinations, leaving us with 896 combinations.

Our data set consists of the following two sets of queries, which we call Q_1 and Q_2, respectively.

The first set Q_1 consists of the top 200 queries issued to IBM's intranet search engine during the period March 14, 2002 to July 8, 2002. Several of these turned out to be broad topic queries, such as "hr," "vacation," "travel," "autonomic computing," etc.—queries that might be expected to be in the bookmarks of several users. These queries also tend to be short, usually single-word queries, and are usually directed towards "hubs" or commonly visited sites on the intranet. These 200 queries represent roughly 39% of all the queries; this suggests that on intranets there is much to be gained by optimizing the search engine, perhaps via feedback learning, to accurately answer the top few queries.

The second set Q_2 consists of 150 queries of median frequency, that is these are the queries near the 50th percentile of the query frequency histogram (from the query logs in the period mentioned above). These are typically not the "bookmark" type queries; rather, they tend to arise when looking for something very specific. In general, these tend to be longer than the popular queries; a nontrivial fraction of them are fairly common queries disambiguated by terms that add to the specificity of the query (e.g., "american express travel canada"). Other types of queries in this category are misspelled common queries (e.g., "ebusiness"), or queries made by users looking for specific parts in a catalog or an specific invoice code (e.g., "cp 10.12", which refers to a corporate procedure). Thus Q_2 represents the "typical" user queries; hence the satisfaction experienced by a user of the search engine will crucially depend on how the search engine handles queries in this category.

We regard both sets of queries as being important, but for different reasons. The distribution of queries to the IBM intranet resembles that found in Internet search engines in the sense that they both have a heavy-tail distribution for query frequency. A very few queries are very common, but most of the workload is on queries that individually occur very rarely. To provide an accurate measure of user satisfaction, we included both classes of queries in our study.

To be able to evaluate the performance of the ranking heuristics and aggregations of them, it is important that we have a clear notion of what "ground truth" is. Therefore, once the queries were identified, the next step was to collect the correct answers for these queries. A subtle issue here is that we cannot use the search engine that we are testing to find out the correct answers, as that would be biasing the results in our favor. Therefore, we resorted to the use of the existing search engine on the IBM intranet, plus a bit of good old browsing; in a handful of cases where we could not find any good page for a query, we did employ our search engine to locate some seed answers, which we then refined by further browsing. If the query was ambiguous (e.g., "american express", which refers to both the corporate credit card and the travel agency) or if there were multiple correct answers to a query (e.g., "sound driver"), we permitted all of them to be included as correct answers.

In this context, we wish to point out that collecting these unique answers was a highly nontrivial task, and exposed various vagaries of the intranet. Given the diversity of IBM and the expanse of its intranet, the correct answers to some queries are geography-specific (e.g., "hiking") and site-specific (e.g., "printers"). We realized, perhaps not surprisingly, that finding answers to queries in Q_2 was much more difficult than finding answers to queries in Q_1. Perhaps more surprisingly, many queries in Q_2 had *precise* answer pages. Often, the correct answers were found one or two hops away from the pages returned by the existing search engine.

After collecting the correct answers for the sets Q_1 and Q_2 of queries, we eliminated all queries whose correct answers were not in the set of crawled pages. This left us with 131 queries in Q_1 and 82 queries in Q_2. One might ask how a search engine could afford to miss a large fraction of correct answers from its crawl. There are several reasons for this, including (but not limited to) the presence of `robots.txt` files that limited access, a number of `https` pages that were password-protected, several sites whose administrators expressly requested not to be crawled, etc.

5.2 Evaluation criteria

Traditional information retrieval employs the notions of *precision* and *recall* to measure the performance of a search system. Recall evaluates a search system based on how highly it ranks the documents that corresponds to ground truth. Precision evaluates a search system based on how relevant the documents highly ranked by the search system are to the query. Given our situation where most queries have essentially a small number (often one) of correct answers, the two are essentially equivalent. (It is possible that our search engine finds pages that we did not know about and that are relevant to the query. However, manually evaluating the precision on a large number of combinations of the ranking methods is impossible.) Therefore, we essentially evaluate various measures of recall of our ranking schemes, with respect to the set of correct answers we collected. More precisely, we employ *recall at position p*, for $p \geq 1$, which is the fraction of queries for which a correct answer is returned by the search system in a position $\leq p$.

A measure of goodness (e.g., recall at positions ≤ 3) helps us understand which of the 896 combinations works best with respect to the measure. In understanding the role of each heuristic in the context of aggregation, we need a way to measure how much each heuristic contributes to the quality. Let α be an attribute (one of the ranking heuristics), let S_α^+ denote the set of combinations of attributes where every combination includes α, and let S_α^- denote the set of combinations of attributes where none of the combinations include α. We then define the *influence* of attribute α with respect to a goodness measure μ as follows: let C^+ denote the combination in S_α^+ that has the highest μ value, and let C^- denote the combination in S_α^- that has the highest μ value. Then the influence of α with respect to μ, denoted by $I_\mu(\alpha)$, is defined as $(\mu(C^+) - \mu(C^-))/\mu(C^-)$; that is, $I_\mu(\alpha)$ is the difference according to the μ performance measure between the best combinations with and without the attribute α, divided by $\mu(C^-)$. In describing the results, we typically express this as a percentage.

The notion of influence, as defined above, needs some clarification. One might wonder why we do not instead consider the following quantity: fix μ and α, let C^- denote the combination in S_α^- that has the highest μ value, then define influence as $(\mu(C^- \cup \{\alpha\}) - \mu(C^-))/\mu(C^-)$. The reason we consider $\mu(C^+)$ rather than $\mu(C^- \cup \{\alpha\})$ is that rank aggregation is a fairly subtle process, and adding α to some $C' \in S_\alpha^-$ with $C' \neq C^-$ might reinforce some of the pairwise comparisons made by the heuristics in C', which could then lead to a much better performance than that of $C^- \cup \{\alpha\}$.

Our final evaluation methodology we employ is aimed at measuring the similarity between the top k lists produced by the various ranking heuristics, and how similar the aggregation is to each one of them. Here we use the concept of comparing top k lists (see [17]). Averaged over all queries, we compare the "distance" of the aggregated output to each of the individual rankings given by the attributes. We also evaluate the distance between the individual rankings themselves. The particular top k distance measure we use is $K_{\min}$, which is defined as follows. The $K_{\min}$ distance between two top k lists τ_1 and τ_2 is defined to be the minimum, over all permutations σ_1 extending τ_1 and σ_2 extending τ_2, of the Kendall tau distance between σ_1 and σ_2, normalized to lie between 0 and 1. For further discussion of $K_{\min}$, see [17], where it is also shown that $K_{\min}$ is a "near metric" in a precise sense. Here it suffices to note that $K_{\min}(\tau_1, \tau_2)$, where τ_1 and τ_2 are two top k lists, achieves its minimal value of 0 if and only if $\tau_1 = \tau_2$; in general, values close to 0 indicate lack of disagreement between τ_1 and τ_2, and larger values indicate that either τ_1 and τ_2 are fairly disjoint from each other, or that they disagree on the relative ranking of a significant number of pairs.

5.3 Results and analysis

On the data set Q_1, out of the 131 queries, the best combination achieved a recall of 75 (approx. 57%) in the top 20 positions; in fact there were over 50 combinations that produced at least 70 queries in the top 20 positions. On the data set Q_2, out of the 82 queries, the best combination achieved a recall of 38 (approx. 46%) in the top 20 positions; again over 50 combinations produced at least 35 queries in the top 20 positions. If we consider these to be the good combinations, roughly 15 combinations qualified as good under both data sets. These combinations included between four and eight heuristics, implying that aggregation is *necessary* for producing robust results.

A few words of explanation are in order about the fairly low percentage of queries correctly solved in the top 20 positions. This is primarily due to our *very* stringent evaluation mechanism; in fact, visual inspection of the results for the 213 queries indicated that the top results are usually accurate (the reader may note that our notion of recall is with respect to results largely based on the existing search engine). In addition, the following scenarios are common.

(1) Between the time the intranet was crawled and the time the correct answers to the queries were identified (a couple of months), many URLs were dynamically relocated. Therefore, even if the content were identical, the URLs would be different and a correct answer would be considered incorrect. (We did not perform an approximate match of the content, e.g., using shingles [9], during the evaluation.)

(2) There are several queries for which the top few answers returned by our search engine are fairly "close" (in browsing distance) to the correct answer (e.g., a top-level page is returned from which it would be easy to locate the correct answer). These were not considered correct.

(3) Some URL canonicalizations were missed, leading the evaluation to believe that the correct result was missed while it was reported under an alias URL.

Tables 1 and 2 present the results for the influences of various heuristics. For a ranking method α and a measure of goodness μ, recall that $I_\mu(\alpha)$ denotes the influence (positive or negative) of α with respect to the goodness measure μ. Our μ's are the recall value at various top k positions—1, 3, 5, 10, and 20; we will abbreviate "recall at 1" as "$R1$," etc.

Legend. The following abbreviations are used for the 10 ranking heuristics:

Ti = index of titles, keywords, etc., An = anchortext, Co = content, Le = URL length, De = URL depth, Wo = query words in URL, Di = discriminator, PR = PageRank, In = indegree, Da = discovery date.

α	$I_{R1}(\alpha)$	$I_{R3}(\alpha)$	$I_{R5}(\alpha)$	$I_{R10}(\alpha)$	$I_{R20}(\alpha)$
Ti	29.2	13.6	5.6	6.2	5.6
An	24.0	47.1	58.3	74.4	87.5
Co	3.3	−6.0	−7.0	−4.4	−2.7
Le	3.3	4.2	1.8	0	0
De	−9.7	−4.0	−3.5	−2.9	−4.0
Wo	3.3	0	−1.8	0	1.4
Di	0	−2.0	−1.8	0	0
PR	0	13.6	11.8	7.9	2.7
In	0	−2.0	−1.8	1.5	0
Da	0	4.2	5.6	4.6	0

Table 1: Influences of various ranking heuristics on the recall at various positions on the query set Q_1

α	$I_{R1}(\alpha)$	$I_{R3}(\alpha)$	$I_{R5}(\alpha)$	$I_{R10}(\alpha)$	$I_{R20}(\alpha)$
Ti	6.7	8.7	3.4	3.0	0
An	23.1	31.6	30.4	21.4	15.2
Co	−6.2	−4.0	3.4	0	5.6
Le	6.7	−4.0	0	0	−5.3
De	−18.8	−8.0	−10	−8.8	−7.9
Wo	6.7	−4.0	0	0	0
Di	−6.2	−4.0	0	0	0
PR	6.7	4.2	11.1	6.2	2.7
In	−6.2	−4.0	0	0	0
Da	14.3	4.2	3.4	0	2.7

Table 2: Influences of various ranking heuristics on the recall at various positions on the query set Q_2

Some salient observations. (1) Perhaps the most noteworthy aspect of Tables 1 and 2 is the amazing efficacy of anchortext. In Table 1, the influence of anchortext can be seen to be progressively better as we relax the recall parameter. For example, for recall at position 20, anchortext has an influence of over 87%, which means that using anchortext leads to essentially *doubling* the recall performance!

(2) Table 1 also shows that the title index (which, the reader may recall, consists of words extracted from titles, `meta`-tagged information, keywords, etc.) is an excellent contributor to achieving very good recall, especially at the top 1 and top 3. Specifically, notice that adding information from the title index improves the accuracy at top 1 by nearly 30%, the single largest improvement for the top 1. Interestingly enough, at top 20, the role of this index is somewhat diminished, and, compared to anchortext, is quite weak.

One way to interpret the information in the first two rows of Table 1 is that anchortext fetches the important pages into the top 20, and the title index pulls up the most accurate pages to the near top. This is also evidence that different heuristics have different roles, and a good aggregation mechanism serves as a glue to bind them seamlessly.

(3) Considering the role of the anchortext in Table 2, which corresponds to the query set Q_2 (the "typical," as opposed to the "popular" queries), we notice that the monotonic increase in contribution (with respect to the position) of anchortext is no longer true. This is not very surprising, since queries in this set are less likely to be extremely important topics with their own web pages (which is the primary cause of a query word being in some anchortext). Nevertheless, anchortext still leads to a 15% improvement in recall at position 20, and is the biggest contributor.

The effect of the title index is also less pronounced in Table 2, with no enhancement to the recall at position 20.

Observations (1)–(3) lead to several inferences.

Inference 1. Information in anchortext, document titles, keyword descriptors and other meta-information in documents, is extremely valuable for intranet search.

Inference 2. Our idea of building separate indices based on this information, as opposed to treating this as auxiliary information and using it to tweak the content index, is particularly effective. These indices are quite compact (roughly 5% and 10% of the size of the content index), fairly easy to build, and inexpensive to access.

Inference 3. Information from these compact indices is query-dependent; thus, these ranking methods are dynamic. The rank aggregation framework allows for easy integration of such information with static rankings such as PageRank.

Continuing with our observations on Tables 1 and 2:

(4) The main index of information on the intranet, namely the content index, is quite ineffective for the popular queries in Q_1. However, it becomes increasingly more effective when we consider the query set Q_2, especially when we consider recall at position 20. This fact is in line with our expectations, since a large number of queries in Q_2 are pointed queries on specialized topics, ones that are more likely to be discussed inside documents rather than in their headers.

Inference 4. Different heuristics have different performances for different types of queries, especially when we compare their performances on "popular" versus "typical" queries. An aggregation mechanism is a convenient way to unify these heuristics, especially in the absence of "classifiers" that tell which type a given query is. Such a classifier, if available, is a bonus, since we could choose the right heuristics for aggregation depending on the query type.

(5) The ranking heuristics based on URL length (Le), presence of query words in the URL (Wo), discovery date (Da), and PageRank (PR) form an excellent support cast in the rank aggregation framework. The first three of these are seen to be especially useful in Q_2, the harder set of queries. An interesting example is discovery date (Da), which has quite a significant effect on the top 1 recall for Q_2.

(6) While PageRank is uniformly good, contrary to its high-impact role on the Internet, it does not add much value in bringing good pages into the top 20 positions; its value

seems more in nudging the ranks of the good pages further.

(7) The heuristics based on URL depth (De), discriminators (Di) and indegree of node (In) appear ineffective for both types of queries. This is not to say that these heuristics will be always bad; conceivably, they might work well on other intranets (conversely, Le, Wo, Da, etc., might be poorer on other intranets). The URL length and depth heuristics (Le and De) are worse on Q_2 than on Q_1, indicating that for more pointed queries, looking for shorter, shallower URLs is probably a bad idea.

Summing up observations (5)–(7), we have:

Inference 5. A plethora of auxiliary heuristics are quite useful to consider. Some of them are helpful in improving the quality of the ranking, and some of them might not be. A plug-and-play architecture, such as ours, allows an administrator to choose the right ones for a given intranet.

Next we summarize the $K_{\min}$ distance defined in Section 5.2 between the ten heuristics employed, and also their distances to the aggregation of all ten heuristics. Table 3 reports these distances as percentages, averaged over all the queries in $Q_1 \cup Q_2$; we report these for the union, rather than for the query sets individually, since the tables from the two query sets were very similar.

α	AG	Ti	An	Co	Le	De	Wo	Di	PR	In	Da
AG	0	26	34	26	32	42	47	42	36	34	13
Ti	26	0	27	21	34	31	26	29	27	31	08
An	34	27	0	32	45	46	40	43	41	33	13
Co	26	21	32	0	21	31	36	35	33	37	12
Le	32	34	45	21	0	33	54	43	51	49	17
De	42	31	46	31	33	0	53	52	51	52	20
Wo	47	26	40	36	54	53	0	48	48	49	16
Di	42	29	43	35	43	52	48	0	47	46	15
PR	36	27	41	33	51	51	48	47	0	31	12
In	34	31	33	37	49	52	49	46	31	0	13
Da	13	08	13	12	17	20	16	15	12	13	0

Table 3: Distances between the heuristics and to their aggregation, query set $Q_1 \cup Q_2$, normalized to be between 0 and 100

We will make a short list of observations concerning Table 3. Recall that in our architecture, we first consult the three indices (Title, Anchortext, and Content), compute the union of the top 100 results from each index, and rank them according to the other seven heuristics.

(1) The seven auxiliary ranking heuristics are much more closely aligned with the ranking based on the title index (distances in the high 20's) than with the ranking based on the content index (distances in the mid 30's), and much more than with the ranking based on the anchortext index (distances in the low 40's). Thus they are more likely to boost results from the title and content indices than they do the results from the anchortext index.

(2) The auxiliary ranking heuristics, with the exception of discovery date (Da), are quite dissimilar to each other. Two exceptions (that are not very surprising) are the pairs URL length and URL depth (Le and De), and PageRank and indegree (PR and In).

(3) The discovery date heuristic (Da) disagrees very little with the others, primarily because this information was only used for a subset of about 3 million pages that were discovered the earliest in the crawl. As noted earlier from Tables 1 and 2, this heuristic is nevertheless quite powerful.

(4) The rankings based on the anchortext, title and content indices are not as dissimilar as one might have expected based on Tables 1 and 2. This suggests that even though they make unique contributions with respect to the recall parameter (which depends on bringing a small set of pages into the top k), they nevertheless have significant similarity between each other. In particular, we speculate that the pages ranked highly by more than one of these heuristics are quite reasonable responses to the query. (While it is a daunting task to verify this speculation rigorously, our personal experience with our search system is that the results in the top 20 are usually good.)

5.4 Lessons learned

We conclude this section with some general observations arising from this investigation. There is a conflict between the desire to have a good searchable intranet and the inherent diversification of the way that information is presented using web technology. In many ways this mirrors the tensions that exist on the the Internet. People want their Internet pages to be seen, and Internet implementors want their information to be discoverable. At the same time, myriad other factors such as social forces, technology limitations, and a lack of understanding of search by web developers can lead to decisions that conflict with having good search.

As we have previously observed, intranet search is different from Internet search for several reasons: the queries asked on the intranet are different, the notion of a "good answer" is different, and the social processes that create the intranet are different from those that create the Internet.

We found that most intranet queries are jargon-heavy and use various acronyms and abbreviations. This may be reflective of the culture of the company that we work for, but we suspect it is not unique to IBM. Because of the fact that we studied the intranet of a large geographically distributed corporation, we found that the correct answer to a query is often specific to a site, geographic location, or an organizational division, but the user often does not make this intent explicit in the query. While this may not hold for every intranet, we expect that context-sensitive search is a common problem for other intranets and the Internet.

6. CONCLUSIONS AND FUTURE WORK

Our main conclusion is that intranets and the Internet are rather different. They share a great deal, but there are also different forces guiding their development, and different measures for their success. The social impact of the World Wide Web is indisputable, and this is one reason why it is such an interesting subject to study. Just as the Web is changing the world, intranets are changing the face of business, government, and other organizations. These developments are largely hidden from public view, and somewhat difficult to study in an open way because each researcher only sees his slice of this "hidden web".

In this investigation we have focused on the problem of search. We have described our system for intranet search, which makes use of rank aggregation. This is a flexible, modular approach that allows us to easily combine various ranking heuristics. This approach should adapt well to other intranets, where a different set of heuristics might be applicable. Perhaps the strongest insight of our study is that the combination of users looking for unique resources (our Axiom 2) and the lack of spam (Axiom 3) makes separate anchortext and title text indices very effective, especially in answering popular queries. Although the overall structure of the IBM intranet is quite different from that of the Internet, global analysis techniques such as PageRank are still helpful, though not sufficient in themselves. Local techniques like indegree and URL depth, even in the absence of spam, do not seem to take the place of a more global view. Since PageRank and indegree are known to be correlated in some web models [15], this fact reinforces the idea that intranets are sufficiently different from the Internet to merit further investigation in their own right.

Acknowledgements. We thank Jason Zien, Reiner Kraft, Sridhar Rajagopalan, Joerg Meyer, and Yu Liao for assistance and useful conversations during the course of this project. We thank Uma Mahadevan for suggesting the title of this paper.

7. REFERENCES

[1] James Allan, Margaret E. Connel, W. Bruce Croft, Fang-Fang Feng, David Fisher, and Zioayan Li. INQUERY and TREC-9. In *Proc. 9th TREC*, pages 551–562, 2000.

[2] Kenneth J. Arrow. *Social Choice and Individual Values.* Yale University Press, New Haven, 2nd edition, 1963.

[3] Javed A. Aslam and Mark Montague. Models for metasearch. In *Proc. 24th SIGIR*, pages 276–284, 2001.

[4] Lauren A. Bednarcyk and Kevin D. Bond. A local web for information delivery. In *Proc. 2nd WWW*, 1994.

[5] K. Bharat and M. Henzinger. Improved algorithms for topic distillation in a hyperlinked environment. In *Proc. 21st SIGIR*, pages 104–111, 1998.

[6] A. Rosina Bignall, Dalinda Kae Bond, Judy Cossel Rice, and Phllip J. Windley. Uses of Mosaic in a university setting. In *Proc. 2nd WWW*, 1994.

[7] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. pages 107–117, 1998.

[8] Andrei Broder, Ravi Kumar, Farzin Maghoul, Prabhakar Raghavan, Sridhar Rajagopalan, Raymie Stata, Andrew Tomkins, and Janet L. Wiener. Graph structure in the web. In *Proc. 9th WWW*, pages 309–320, 2000.

[9] Andrei Z. Broder, Steven Glassman, Mark S. Manasse, and Geoffrey Zweig. Syntactic clustering of the web. *WWW6/Computer Networks*, 29(8-13):1157–1166, 1997.

[10] Soumen Chakrabarti, Byron Dom, David Gibson, Jon M. Kleinberg, Prabhakar Raghavan, and Sridhar Rajagopalan. Automatic resource compilation by analyzing hyperlink structure and associated text. In *Proc. 7th WWW*, pages 65–74, 1997.

[11] Soumen Chakrabarti, Byron E. Dom, David Gibson, Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, and Andrew Tomkins. Experiments in topic distillation. In *SIGIR Workshop on Hypertext Information Retrieval*, pages 13–21, 1998.

[12] Mike Crandall and Mark C. Swenson. Integrating electronic information through a corporate web. In *Proc. 5th WWW*, pages 1175–1186, 1996.

[13] W. Bruce Croft. Combining approaches to information retrieval. In W. Bruce Croft, editor, *Advances in Information Retrieval.* Kluwer Academic Publishers, 2000.

[14] Stephen Dill, Ravi Kumar, Kevin S. McCurley, Sridhar Rajagopalan, D. Sivakumar, and Andrew Tomkins. Self-similarity in the web. In *Proc. 27th VLDB*, pages 69–78, 2001.

[15] Chris Ding, Xiaofeng He, Parry Husbands, and Horst D. Simon. PageRank, HITS, and a unified framework for link analysis. In *Proc. 25th SIGIR*, pages 353–354, 2002.

[16] Cynthia Dwork, Ravi Kumar, Moni Naor, and D. Sivakumar. Rank aggregation methods for the web. In *Proc. 10th WWW*, pages 613–622, 2001.

[17] Ronald Fagin, Ravi Kumar, and D. Sivakumar. Comparing top k lists. In *Proc. 14th SODA*, pages 28–36, 2003.

[18] Shannon L. Fowler, Anne-Marie J. Novack, and Michael J. Stillings. The evolution of a manufacturing web site. In *Proc. 9th WWW*, volume 33, pages 365–376, 2000.

[19] Eric J. Glover, Kostas Tsioutsiouliklis, Steve Lawrence, David M. Pennock, and Gary W. Flake. Using web structure for classifying and describing web pages. In *Proc. 11th WWW*, pages 562–569, 2002.

[20] Djoerd Hiemstra. *Using Language Models for Information Retrieval.* PhD thesis, University of Twente, Twente, The Netherlands, 2001.

[21] M. Huynh, L. Popkin, and M. Stecker. Constructing a corporate memory infrastructure from internet discovery technologies. In *Proc. 2nd WWW*, 1994.

[22] Vlad Ionesco. Using an intranet for real-time production management: Experiences and effects. *WWW7/Computer Networks*, 30(1-7):479–488, 1998.

[23] Rong Jin, Alex G. Hauptmann, and ChengXiang Zhai. Title language model for information retrieval. In *Proc. 25th SIGIR*, pages 42–48, 2002.

[24] Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *JACM*, 46(5):604–632, 1999.

[25] W. Kraaij, T. Westerveld, and D. Hiemstra. The importance of prior probabilities for entry page search. In *Proc. 25th SIGIR*, pages 27–34, 2002.

[26] Mark Montague and Javed A. Aslam. Condorcet fusion for improved retrieval. In *Proc. 11th CIKM*, pages 538–548, 2002.

[27] Marc Najork and Janet L. Wiener. Breadth-first search crawling yields high-quality pages. In *Proc. 10th WWW*, pages 114–118, 2001.

[28] Steve Pavett, Nihal Samarawera, Neil M. Hamilton, and Gorry Fairhurst. Video Medi-CAL: Supporting MPEG-2 media based computer assisted learning on an intranet. *WWW7/Computer Networks*, 30(1-7):672–675, 1998.

[29] Kaitlin Duck Sherwood. Technical and sociological aspects of developing campus-wide webs: UIUC college of engineering. In *Proc. 2nd WWW*, 1994.

[30] Thijs Westerveld, Wessel Kraaij, and Djoerd Hiemstra. Retrieving web pages using content, links, URLs and anchors. In *Proc. 10th TREC*, pages 663–672, 2001.