Hilltop: A Search Engine based on Expert Documents

Krishna Bharat
Compaq, Systems Research Center, Palo Alto, CA 94301
(*Current Address:* Google Inc., 2400 Bayshore Parkway,
Mountain View, CA 94043)
`krishna@google.com`

George A. Mihaila
Department of Computer Science
University of Toronto.
`georgem@cs.toronto.edu`

Abstract:

In response to a query a search engine returns a ranked list of documents. If the query is broad (i.e., it matches many documents) then the returned list is usually too long to view fully. Studies show that users usually look at only the top 10 to 20 results. In this paper, we propose a novel ranking scheme for broad queries that places the most *authoritative* pages on the query topic at the top of the ranking. Our algorithm operates on a special index of "expert documents." These are a subset of the pages on the WWW identified as directories of links to non-affiliated sources on specific topics. Results are ranked based on the match between the query and relevant descriptive text for hyperlinks on expert pages pointing to a given result page. We present a prototype search engine that implements our ranking scheme and discuss its performance. With a relatively small (2.5 million page) expert index, our algorithm was able to perform comparably on broad queries with the best of the mainstream search engines.

1 Introduction

When searching the WWW broad queries tend to produce a large result set. This set is hard to rank based on content alone, since the quality and "authoritativeness" of a page (namely, *a measure of how authoritative the page is on the subject*) cannot be assessed solely by analyzing its content. In traditional information retrieval we make the assumption that the articles in the corpus originate from a reputable source and all words found in an article were intended for the reader. These assumptions do not hold on the WWW since content is authored by sources of varying quality and words are often added indiscriminately to boost the page's ranking. For example, some pages are created to purposefully mislead search engines, and are known popularly as "spam" pages. The most virulent of spam techniques involves deliberately returning someone else's popular page to search engine robots instead of the actual page, to steal their traffic. Even when there is no intention to mislead search engines, the WWW tends to be crowded with information on topics popular with users. Consequently, for broad queries keyword matching seems inadequate.

Prior approaches that have used content analysis to rank broad queries on the WWW cannot distinguish between authoritative and non-authoritative pages (e.g., they fail to detect spam pages). Hence the ranking tends to be poor and search services have turned to other sources of information besides content to rank results. We next describe some of these ranking strategies, followed by our new approach to authoritative ranking - which we call *Hilltop*.

1.1 Related Work

Three approaches to improve the authoritativeness of ranked results have been taken in the past:

Ranking Based on Human Classification: Human editors have been used by companies such as Yahoo! and Mining Company to manually associate a set of categories and keywords with a subset of documents on the web. These are then matched against the user's query to return valid matches. The trouble with this approach is that: (a) it is slow and can only be applied to a small number of pages, and (b) often the keywords and classifications assigned by the human judges are inadequate or incomplete. Given the rate at which the WWW is growing and the wide variation in queries this is not a comprehensive solution.

Ranking Based on Usage Information: Some services such as DirectHit collect information on: (a) the queries individual users submit to search services and (b) the pages they look at subsequently and the time spent on each page. This information is used to return pages that *most* users visit after deploying the given query. For this technique to succeed a large amount of data needs to be collected for each query. Thus, the potential set of queries on which this technique applies is small. Also, this technique is open to spamming.

Ranking Based on Connectivity: This approach involves analyzing the hyperlinks between pages on the web on the assumption that: (a) pages on the topic link to each other, and (b) authoritative pages tend to point to other authoritative pages.

PageRank [Page et al 98] is an algorithm to rank pages based on assumption *b*. It computes a query-independent authority score for every page on the Web and uses this score to rank the result set. Since *PageRank* is query-independent it cannot by itself distinguish between pages that are authoritative in general and pages that are authoritative on the query topic. In particular a web-site that is authoritative in general *may* contain a page that matches a certain query but is not an authority on the topic of the query. In particular, such a page may not be considered valuable within the community of users who author pages on the topic of the query.

An alternative to *PageRank* is *Topic Distillation* [Kleinberg 97, Chakrabarti et al 98, Bharat et al 98, Chakrabarti et al 99]. Topic distillation first computes a query specific subgraph of the WWW. This is done by including pages on the query topic in the graph and ignoring pages not on the topic. Then the algorithm computes a score for every page in the subgraph based on hyperlink connectivity: every page is given an authority score. This score is computed by summing the weights of all incoming links to the page. For each such reference, its weight is computed by evaluating how good a source of links the referring page is. Unlike *PageRank*, *Topic Distillation* is only applicable to broad queries, since it requires the presence of a community of pages on the topic.

A problem with *Topic Distillation* is that computing the subgraph of the WWW which is on the query topic is hard to do in real-time. In the ideal case every page on the WWW that deals with the query topic would need to be considered. In practice an approximation is used. A preliminary ranking for the query is done with content analysis. The top ranked result pages for the query are selected. This creates a *selected set*. Then, some of the pages within one or two links from the selected set are also added to the selected set if they are on the query topic. This approach can fail because it is dependent on the comprehensiveness of the selected set for success. A highly relevant and authoritative page may be omitted from the ranking by this scheme if it either did not appear in the initial selected set, or some of the pages pointing to it were not added to the selected set. A "focused crawling" procedure to crawl the entire web to find the complete subgraph on the query's topic has been proposed [Chakrabarti et al 99] but this is too slow for online searching. Also, the overhead in computing the full subgraph for the query is not warranted since users only care about the top ranked results.

1.2 Hilltop Algorithm Overview

Our approach is based on the same assumptions as the other connectivity algorithms, namely that the number and quality of the sources referring to a page are a good measure of the page's quality. The key difference consists in the fact that we are only considering "expert" sources - pages that have been created with the specific purpose of directing people towards resources. In response to a query, we first compute a list of the most relevant experts on the query topic. Then, we identify relevant links within the selected set of experts, and follow them to identify target web pages. The targets are then ranked according to the number and relevance of non-affiliated experts that point to them. Thus, the score of a target page reflects the collective opinion of the best independent experts on the query topic. When such a pool of experts is not available, Hilltop provides no results. Thus, Hilltop is tuned for result accuracy and not query coverage.

Our algorithm consists of two broad phases:

(i) Expert Lookup

We define an expert page as a page that is about a certain topic and has links to many non-affiliated pages on that topic. Two pages are non-affiliated conceptually if they are authored by authors from non-affiliated organizations. In a pre-processing step, a subset of the pages crawled by a search engine are identified as experts. In our experiment we classified 2.5 million of the 140 million or so pages in AltaVista's index to be experts. The pages in this subset are indexed in a special inverted index.

Given an input query, a lookup is done on the expert-index to find and rank matching expert pages. This phase computes the best expert pages on the query topic as well as associated match information.

(ii) Target Ranking

We believe a page is an authority on the query topic if and only if some of the best experts on the query topic point to it. Of course in practice some expert pages may be experts on a broader or related topic. If so, only a subset of the hyperlinks on the expert page may be relevant. In such cases the links being considered have to be carefully chosen to ensure that their qualifying text matches the query. By combining relevant out-links from many experts on the query topic we can find the pages that are most highly regarded by the community of pages related to the query topic. This is the basis of the high relevance that our algorithm delivers.

Given the top ranked matching expert-pages and associated match information, we select a subset of the hyperlinks within the expert-pages. Specifically, we select links that we know to have all the query terms associated with them. This implies that the link matches the query. With further connectivity analysis on the selected links we identify a subset of their targets as the top-ranked pages on the query topic. The targets we identify are those that are linked to by *at least two* non-affiliated expert pages on the topic. The targets are ranked by a ranking score which is computed by combining the scores of the experts pointing to the target.

1.3 Roadmap

The rest of the paper is organized as follows: Section 2 describes the selection and indexing of expert documents; Section 3 provides a detailed description of the ranking scheme used in query processing; Section 4 presents a user-based evaluation of our prototype implementation; and Section 5 concludes the paper.

2 Expert Documents

Broad subjects are well represented on the Web and as such are also likely to have numerous human-generated lists of resources. There is value for the individual or organization that creates resource lists on specific topics since this boosts their popularity and influence within the community interested in the topic. The authors of these lists thus have an incentive to make their lists as comprehensive and up to date as possible. We regard these links as recommendations, and the pages that contain them, as experts. The problem is, how can we distinguish an expert from other types of pages? In other words *what makes a page an expert*? We felt than an expert page needs to be objective and diverse: that is, its recommendations should be unbiased and point to numerous *non-affiliated* pages on the subject. Therefore, in order to find the experts, we needed to detect when two sites belong to the same or related organizations.

2.1 Detecting Host Affiliation

We define two hosts as affiliated if one or both of the following is true:

- They share the same first 3 octets of the IP address.
- The rightmost non-generic token in the hostname is the same.

We consider tokens to be substrings of the hostname delimited by "." (period). A suffix of the hostname is considered generic if it is a sequence of tokens that occur in a large number of distinct hosts. E.g., ".com" and ".co.uk" are domain names that occur in a large number of hosts and are hence generic suffixes. Given two hosts, if the generic suffix in each case is removed and the subsequent right-most token is the same, we consider them to be affiliated.

E.g., in comparing "www.ibm.com" and "ibm.co.mx" we ignore the generic suffixes ".com" and ".co.mx" respectively. The resulting rightmost token is "ibm", which is the same in both cases. Hence they are considered to be affiliated. Optionally, we could require the generic suffix to be the same in both cases.

The affiliation relation is transitive: if A and B are affiliated and B and C are affiliated then we take A and C to be affiliated even if there is no direct evidence of the fact. In practice some non-affiliated hosts may be classified as affiliated, but that is acceptable since this relation is intended to be conservative.

In a preprocessing step we construct a host-affiliation lookup. Using a union-find algorithm we group hosts, that either share the same rightmost non-generic suffix or have an IP address in common, into sets. Every set is given a unique identifier (e.g., the host with the lexicographically lowest hostname). The host-affiliation lookup maps every host to its set identifier or to itself (when there is no set). This is used to compare hosts. If the lookup maps two hosts to the same value then they are affiliated; otherwise they are non-affiliated.

2.2 Selecting the Experts

In this step we process a search engine's database of pages (we used AltaVista's crawl from April 1999) and select a subset of pages which we consider to be good sources of links on specific topics, albeit unknown. This is done as follows:

Considering all pages with out-degree greater than a threshold, k (e.g., k=5) we test to see if these URLs point to k distinct *non-affiliated* hosts. Every such page is considered an expert page.

If a broad classification (such as *Arts*, *Science*, *Sports* etc.) is known for every page in the search engine database then we can additionally require that most of the k non-affiliated URLs discovered in the previous step point to pages that share the same broad classification. This allows us to distinguish between random collections of links and resource directories. Other properties of the page such as regularity in formatting can be used as well.

2.3 Indexing the Experts

To locate expert pages that match user queries we create an inverted index to map keywords to experts on which they occur. In doing so we only index text contained within "key phrases" of the expert. A key phrase is a piece of text that qualifies one or more URLs in the page. Every key phrase has a scope within the document text. URLs located within the scope of a phrase are said to be "qualified" by it. For example, the title, headings (e.g., text within a pair of `<H1> </H1>` tags) and anchor text within the expert page are considered key phrases. The title has a scope that qualifies all URLs in the document. A heading's scope qualifies all URLs until the next heading of the same or greater importance. An anchor's scope only extends over the URL it is associated with.

The inverted index is organized as a list of match positions within experts. Each match position corresponds to an occurrence of a certain keyword within a key phrase of a certain expert page. All match positions for a given expert occur in sequence for a given keyword. At every match position we also store:

i. An identifier to identify the phrase uniquely within the document
ii. A code to denote the kind of phrase it is (title, heading or anchor)
iii. The offset of the word within the phrase.

In addition, for every expert we maintain the list of URLs within it (as indexes into a global list of URLs) and for each URL we maintain the identifiers of the key phrases that qualify it.

To avoid giving long key phrases an advantage, the number of keywords within any key phrase is limited (e.g., to 32).

3 Query Processing

In response to a user query, we first determine a list of N experts that are the most relevant for that query. E.g. N = 200 in our experiment. Then, we rank results by selectively following the relevant links from these experts and assigning an authority score to each such page. In this section we describe how the expert and authority scores are computed.

3.1 Computing the Expert Score

For an expert to be useful in response to a query, the minimum requirement is that there is at least one URL which contains all the query keywords in the key phrases that *qualify* it. A fast approximation is to require all query keywords to occur in the document. Furthermore, we assign to each candidate expert a score reflecting the number and importance of the key phrases that contain the query keywords, as well as the degree to which these phrases match the query.

Thus, we compute the score of an expert as as a 3-tuple of the form (S_0, S_1, S_2). Let k be the number of terms in the input query, q. The component S_i of the score is computed by considering only key phrases that contain precisely k - i of the query terms. E.g., S_0 is the score computed from phrases containing all the query terms.

$$S_{\mathrm{i}} = \mathrm{SUM}_{\{\text{key phrases } p \text{ with } k - i \text{ query terms}\}}\ LevelScore(p) * FullnessFactor(p, q)$$

LevelScore(*p*) is a score assigned to the phrase by virtue of the type of phrase it is. For example, in our implementation we use a *LevelScore* of 16 for title phrases, 6 for headings and 1 for anchor text. This is based on the assumption that the title text is more useful than the heading text, which is more useful than an anchor text match in determining what the expert page is about.

FullnessFactor(*p, q*) is a measure of the number of terms in *p* covered by the terms in *q*. Let *plen* be the length of *p*. Let *m* be the number of terms in *p* which are not in *q* (i.e., surplus terms in the phrase). Then, *FullnessFactor*(*p, q*) is computed as follows:

- If m <= 2, *FullnessFactor*(*p, q*) = 1
- If m > 2, *FullnessFactor*(*p, q*) = 1 - (*m* - 2) / *plen*

Our goal is to prefer experts that match all of the query keywords over experts that match all but one of the keywords, and so on. Hence we rank experts first by S_0. We break ties by S_1 and further ties by S_2. The score of each expert is converted to a scalar by the weighted summation of the three components:

$$Expert\_Score = 2^{32} * S_0 + 2^{16} * S_1 + S_2.$$

3.2 Computing the Target Score

We consider the top *N* experts by the ranking from the previous step (e.g., the top 200) and examine the pages they point to. These are called *targets*. It is from this set of targets that we select top ranked documents. For a target to be considered it must be pointed to by at least 2 experts on hosts that are mutually non-affiliated and are not affiliated to the target. For all targets that qualify we compute a target score reflecting both the number and relevance of the experts pointing to it and the relevance of the phrases qualifying the links.

The target score *T* is computed in three steps:

1. For every expert *E* that points to target *T* we draw a directed edge (*E,T*). Consider the following

"qualification" relationship between key phrases and edges:

- The title phrase *qualifies* all edges coming out of the expert
- A heading *qualifies* all edges whose corresponding hyperlinks occur in the document *after* the given heading and *before* the next heading of equal or greater importance.
- A hyperlink's anchor text *qualifies* the edge corresponding to the hyperlink.

For each query keyword w, let $occ(w, T)$ be the number of distinct key phrases in E that contain w and *qualify* the edge (E,T). We define an "edge score" for the edge (E,T) represented by $Edge\_Score(E,T)$, which is computed thus:

- If $occ(w, T)$ is 0 for any query keyword then the $Edge\_Score(E,T) = 0$.
- Otherwise, $Edge\_Score(E,T) = Expert\_Score(E) * \mathrm{Sum}_{\{\text{query keywords } w\}}\, occ(w, T)$

2. We next check for affiliations between expert pages that point to the same target. If two affiliated experts have edges to the same target T, we then discard one of the two edges. Specifically, we discard the edge which has the lower *Edge_Score* of the two.
3. To compute the *Target_Score* of a target we sum the *Edge_Score*s of all edges incident on it.

The list of targets is ranked by *Target_Score*. Optionally, this list can be filtered by testing if the query keywords are present in the targets. Optionally, we can match the query keywords against each target to compute a *Match_Score* using content analysis, and combine the *Target_Score* with the *Match_Score* before ranking the targets.

Figure 1. Hilltop Ranking for the Query: "jobs"

4 Evaluation

In order to evaluate our prototype search engine, we conducted two user studies aiming to estimate the recall and precision. Both experiments also involved three other search engines, namely *AltaVista*, *DirectHit* and *Google*, for comparison and were done in August 1999. Note that the current rankings by these engines may differ.

4.1 Locating Specific Popular Targets

For the first experiment we asked seven volunteers to suggest the home pages of ten organizations of their choice (companies, universities, stores, etc.). Some of the queries are reproduced in the table below:

Alpha Phi Omega	Best Buy	Digital	Disneyland
Dollar Bank	Grouplens	INRIA	Keebler
Mountain View Public Library	Macy's	Minneapolis City Pages	Moscow Aviation Institute
MENSA	OCDE	ONU	Pittsburg Steelers
Pizza Hut	Rice University	SONY	Safeway
Stanford Shopping Center	Trek Bicycle	USTA	Vanguard Investments

The same query was sent to all four search engines. We assume that there is exactly one home page in each case. Every time the home page was found within the first ten results, its rank was recorded. Figure 2 summarizes the average recall for the ranks 1 to 10 for each of the four engines: our engine *Hilltop* (HT), *Google* (GG), *AltaVista* (AV), and *DirectHit* (DH). Average recall at rank k for this experiment is the probability of finding the desired home page within the first k results.

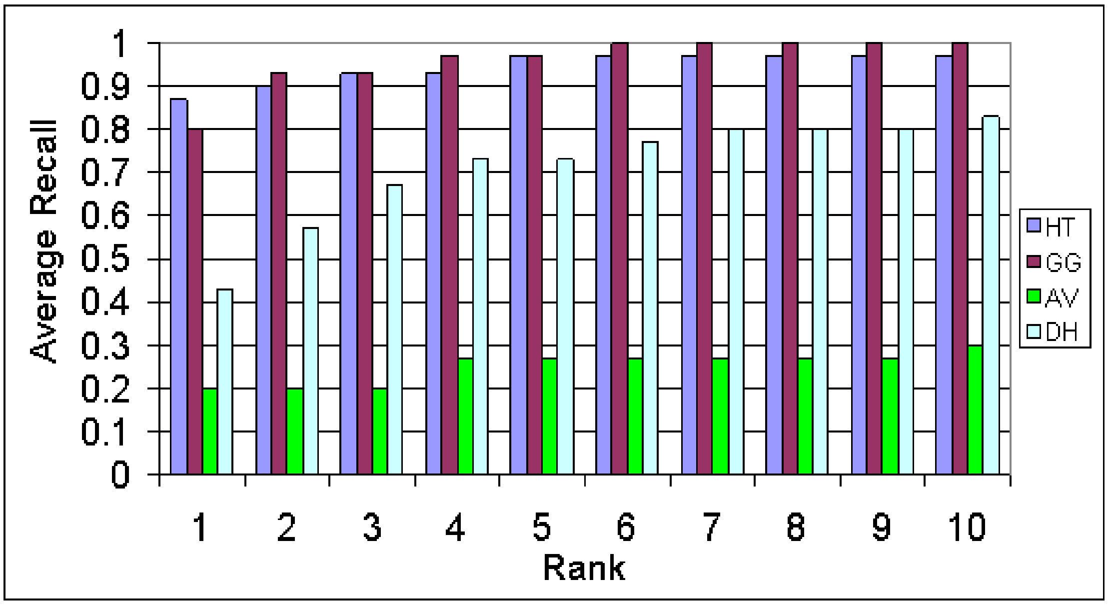

Figure 2. Average Recall vs. Rank

Our engine performed well on these queries. Thus, for about 87% of the queries, *Hilltop* returned the desired page as the first result, comparable with *Google* at 80% of the queries, while *DirectHit* and *AltaVista* succeeded at rank 1 only in 43% and 20% of the cases, respectively. As we look at more results, the average recall increases to 100% for *Google*, 97% for *Hilltop*, 83% for *DirectHit*, and 30% for *AltaVista*.

4.2 Gathering Relevant Pages

In order to estimate Hilltop's ability to generate a good first page of results for broad queries, we asked our volunteers to think of broad topics (i.e., topics for which it is likely that many good pages exist) and formulate queries. We collected 25 such queries, listed below:

Aerosmith	Amsterdam	backgrounds	chess	dictionary
fashion	freeware	FTP search	Godzilla	Grand Theft Auto
greeting cards	Jennifer Love Hewitt	Las Vegas	Louvre	Madonna
MEDLINE	MIDI	newspapers	Paris	people search
real audio	software	Starr report	tennis	UFO

We then used a script to submit each query to all four search engines and collect the top 10 results from each engine, recording for each result the URL, the rank, and the engine that found it. We needed to determine which of the results were relevant in an unbiased manner. For each query we generated the list of unique URLs in the union of the results from all engines. This list was then presented to a judge in a random order, without any information about the ranks of page or their originating engine. The judge rated each page for relevance to the given query on a binary scale (1 = "good page on the topic", 0 = "not relevant or not found"). Then, another script combined these ratings with the information about provenance and rank and computed the average precision at rank *k* (for *k* = 1, 5, and 10). The results are summarized in Figure 3.

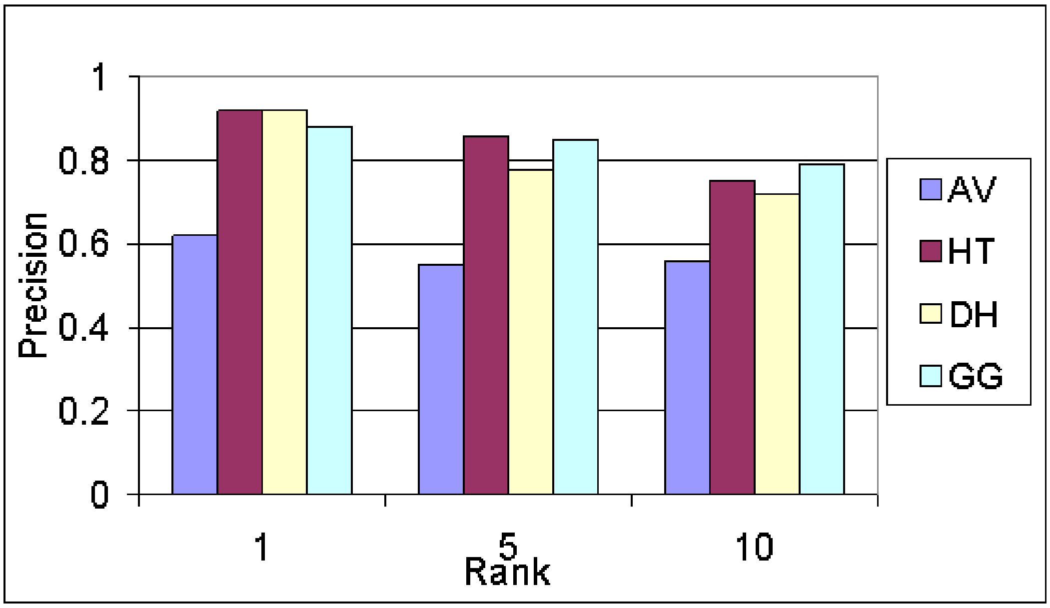

Figure 3. Average Precision at Rank *k*

These results indicate that for broad subjects our engine returns a large percentage of highly relevant pages among the ten best ranked pages, comparable with *Google* and *DirectHit*, and better than *AltaVista*. At rank 1 both *Hilltop* and *DirectHit* have an average precision of 0.92. Average precision at 10 for *Hilltop* was 0.77, roughly equal to the best search engine, namely *Google*, with a precision of 0.79 at rank 10.

5 Conclusions

We described a new ranking algorithm for broad queries called *Hilltop* and the implementation of a search engine based on it. Given a broad query *Hilltop* generates a list of target pages which are likely to be very authoritative pages on the topic of the query. This is by virtue of the fact that they are highly valued by pages on the WWW which address the topic of the query. In computing the usefulness of a target page from the hyperlinks pointing to it, we only consider links originating from pages that seem to be experts. Experts in our definition are directories of links pointing to many non-affiliated sites. This is an indication that these pages were created for the purpose of directing users to resources, and hence we regard their opinion as valuable. Additionally, in computing the level of relevance, we require a match between the query and the text on the expert page which qualifies the hyperlink being considered. This ensures that hyperlinks being considered are on the query topic. For further accuracy, we require that at least 2 non-affiliated experts point to the returned page with relevant qualifying text describing their linkage. The result of the steps described above is to generate a listing of pages that are highly relevant to the user's query and of high quality.

Hilltop most resembles the connectivity techniques, *PageRank* and *Topic Distillation*. Unlike *PageRank* our technique is a dynamic one and considers connectivity in a graph specifically about the query topic. Hence, it can evaluate relevance of content from the point of view of the community of authors interested in the query topic. Unlike *Topic Distillation* we enumerate and consider all good experts on the subject and correspondingly all good target pages on the subject. In order to find the most relevant experts we use a custom keyword-based approach, focusing only on the text that best captures the domain of expertise (the document title, section headings and hyperlink anchor-text). Then, in following links, we boost the score of those targets whose qualifying text best matches the query. Thus, by combining content and connectivity analysis, we are both more comprehensive and more precise. An important property is that unlike *Topic Distillation* approaches, we can *prove* that if a page does not appear in our output it lacks the connectivity support to justify its inclusion. Thus we are less prone to omit good pages on the topic, which is a problem with *Topic Distillation* systems. Also, since we use an index optimized to finding experts, our implementation uses less data than *Topic Distillation* and is therefore faster.

The indexing of anchor-text was first suggested in *WWW Worm* [McBryan 94]. In some *Topic Distillation* systems such as *Clever* [Chakrabarti et al 1998] and in the *Google* search engine [Page et al 98] anchor-text is considered in evaluating a link's relevance. We generalize this to other forms of text that are seen to "qualify" a hyperlink at its source, and include headings and title-text as well. Also, unlike *Topic Distillation* systems, we evaluate experts on their content match to the user's query, rather than on their linkage to good target pages. This prevents the scores of "niche experts" (i.e., experts that point to new or relative poorly connected pages) from being driven to zero, as is often the case in *Topic Distillation* algorithms.

In a blind evaluation we found that *Hilltop* delivers a high level of relevance given broad queries, and performs comparably to the best of the commercial search engines tested.

6 References

[Kleinberg 97] J. Kleinberg. Authoritative sources in a hyperlinked environment. To appear in the *Journal of the ACM*, 1999. Also appears as IBM Research Report RJ 10076, May 1997. http://www.cs.cornell.edu/home/kleinber/auth.ps

[Chakrabarti et al 98] S. Chakrabarti, B. Dom, D. Gibson, J. Kleinberg, P. Raghavan, and S. Rajagopalan. Automatic Resource Compilation by Analyzing Hyperlink Structure and Associated Text. Proceedings of the *7th World-Wide Web conference*, 1998.

http://decweb.ethz.ch/WWW7/1898/com1898.htm

[Chakrabarti et al 99] S. Chakrabarti, M. van den Berg and B. Dom. Focused crawling: A new approach to topic-specific Web resource discovery. In the 8th World Wide Web Conference, Toronto, May 1999. http://www.cs.berkeley.edu/~soumen/doc/www99focus/html/

[Bharat et al 98] K. Bharat and M. Henzinger. Improved algorithms for topic distillation in a hyperlinked environment. In *SIGIR Conference on Research and Development in Information Retrieval*, volume 21. ACM, 1998. ftp://ftp.digital.com/pub/DEC/SRC/publications/monika/sigir98.pdf.

[Page et al 98] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In *WWW Conference*, volume 7, 1998. http://www7.scu.edu.au/programme/fullpapers/1921/com1921.htm

[McBryan 94] Oliver A. McBryan. GENVL and WWWW: Tools for Taming the Web. *First International Conference on the World Wide Web.* CERN, Geneva (Switzerland), May 25-26-27 1994. http://www.cs.colorado.edu/home/mcbryan/mypapers/www94.ps

---

Krishna Bharat is a member of the research staff at Google Inc. in Mountain View, California. Formerly he was at Compaq Computer Corporation's Systems Research Center, which is where the research described here was done. His research interests include Web content discovery and retrieval, user interface issues in Web search and task automation, and relevance assessments on the Web. He received his Ph.D. in Computer Science from Georgia Institute of Technology in 1996, where he worked on tool and infrastructure support for building distributed user interface applications.

George Andrei Mihaila is a Ph.D. student in the Department of Computer Science at the University of Toronto. During the summer of 1999 he was an intern at Compaq Computer Corporation's Systems Research Center, which is where this research was done. His research interests include query languages, information discovery tools, Web-based information systems and database integration. He received his M.Sc. in Computer Science from the University of Toronto in 1996 with the thesis *WebSQL - an SQL-like Query Language for the World Wide Web*.