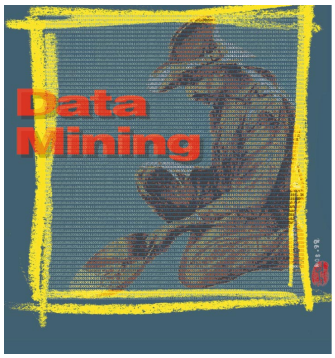
Mining the Web's Link Structure



Sifting through the growing mountain of Web data demands an increasingly discerning search engine, one that can reliably assess the quality of sites, not just their relevance.

Soumen Chakrabarti
Indian Institute of Technology, Bombay

Byron E. Dom

S. Ravi Kumar

Prabhakar Raghavan

Sridhar Rajagopalan

Andrew Tomkins
IBM Almaden Research Center

David Gibson
University of California, Berkeley

Jon Kleinberg
Cornell University

The Web is a hypertext body of approximately 300 million pages that continues to grow at roughly a million pages per day. Page variation is more prodigious than the data's raw scale: Taken as a whole, the set of Web pages lacks a unifying structure and shows far more authoring style and content variation than that seen in traditional text-document collections. This level of complexity makes an "off-the-shelf" database-management and information-retrieval solution impossible.

To date, index-based search engines for the Web have been the primary tool by which users search for information. The largest such search engines exploit technology's ability to store and index much of the Web. Such engines can therefore build giant indices that let you quickly retrieve the set of all Web pages containing a given word or string.

Experienced users can make effective use of such engines for tasks that can be solved by searching for tightly constrained keywords and phrases. These search engines are, however, unsuited for a wide range of equally important tasks. In particular, a topic of any breadth will typically contain several thousand or million relevant Web pages. Yet a user will be willing, typically, to look at only a few of these pages.

How then, from this sea of pages, should a search engine select the *correct* ones—those of most value to the user?

AUTHORITATIVE WEB PAGES

First, to distill a large Web search topic to a size that makes sense to a human user, we need a means of identifying the topic's most definitive or authoritative Web pages. The notion of authority adds a crucial second dimension to the concept of relevance: We wish to locate not only a set of relevant pages, but also those relevant pages of the highest quality.

Second, the Web consists not only of pages, but hyperlinks that connect one page to another. This hyperlink structure contains an enormous amount of latent human annotation that can help automatically infer notions of authority. Specifically, the creation of a hyperlink by the author of a Web page represents an implicit *endorsement* of the page being pointed to; by mining the collective judgment contained in the set of such endorsements, we can gain a richer understanding of the relevance and quality of the Web's contents.

To address both these parameters, we began development of the Clever system[1-3] three years ago. Clever is a search engine that analyzes hyperlinks to uncover two types of pages:

- authorities, which provide the best source of information on a given topic; and
- hubs, which provide collections of links to authorities.

In this article, we outline the thinking that went into Clever's design, report briefly on a study that compared Clever's performance to that of Yahoo and AltaVista, and examine how our system is being extended and updated.

FINDING AUTHORITIES

You could use the Web's link structure in any of several ways to infer notions of authority—some much more effective than others. Because the link structure implies an underlying social structure in the way that pages and links are created, an understanding of this social organization can provide us with the most leverage. Our goal in designing algorithms for mining link information is to develop techniques that take advantage of what we observe about the Web's intrinsic social organization.

Search obstacles

As we consider the types of pages we hope to discover, and to do so automatically, we quickly confront some difficult problems. First, it is insufficient to apply purely text-based methods to collect many potentially

0018-9162/99/$10.00 © 1999 IEEE

relevant pages, and then comb this set for the most authoritative ones. For example, were we to look for the Web's main search engines, we would err badly if we searched only for "search engines." Although the set of pages containing this term is enormous, it does not contain most of the natural authorities we would expect to find, such as Yahoo, Excite, InfoSeek, and AltaVista. Similarly, we cannot expect Honda's or Toyota's home pages to contain the words "Japanese automobile manufacturers," nor that Microsoft's or Lotus' home pages will contain the words "software companies." Authorities are seldom particularly self-descriptive. Large corporations design their Web pages carefully to convey a certain feel and project the correct image—a goal that might differ significantly from that of actually describing the company. Thus, people outside a company frequently create more recognizable and sometimes better judgments about it than does the company itself.

Working with hyperlink information causes difficulties as well. Although many links represent the type of endorsement we seek—for example, a software engineer whose home page links to Microsoft and Lotus—others are created for reasons that have nothing to do with conferring authority. Some links exist purely for navigational purposes: "Click here to return to the main menu." Others serve as paid advertisements: "The vacation of your dreams is only a click away." We hope, however, that in an *aggregate* sense, over a large enough number of links, our view of links as conferring authority will hold.

Modeling authority conferral

How can we best model the way in which authority is conferred on the Web? Clearly, when commercial or competitive interests are at stake, most organizations will perceive no benefit from linking directly to one another. For example, AltaVista, Excite, and InfoSeek may all be authorities for the topic "search engines," but will be unlikely to endorse one another directly.

If the major search engines do not explicitly describe themselves as such, how can we determine that they are indeed the most authoritative pages for this topic? We could say that they are authorities because many relatively anonymous pages, clearly relevant to "search engines," link to AltaVista, Excite, and Infoseek. Such pages are a recurring Web component: hubs that link to a collection of prominent sites on a common topic. Hub pages appear in a variety of forms, ranging from professionally assembled resource lists on commercial sites to lists of recommended links on individual home pages. These pages need not be prominent themselves, or even have any links pointing to them. Their distinguishing feature is that they are potent conferrers of authority on a focused topic. In this way, they actually form a symbiotic relationship with authorities: A good authority is a page pointed to by many good hubs, while a good hub is a page that points to many good authorities.[3]

This mutually reinforcing relationship between hubs and authorities serves as the central theme in our exploration of link-based methods for search, the automated compilation of high-quality Web resources, and the discovery of thematically cohesive Web communities.

We view the Web as a directed graph, consisting of a set of nodes with directed edges between certain node pairs.

HITS: COMPUTING HUBS AND AUTHORITIES

The HITS (Hyperlink-Induced Topic Search) algorithm[3] computes lists of hubs and authorities for Web search topics. Beginning with a search topic, specified by one or more query terms, the HITS algorithm applies two main steps:

- a sampling component, which constructs a focused collection of several thousand Web pages likely to be rich in relevant authorities; and
- a weight-propagation component, which determines numerical estimates of hub and authority weights by an iterative procedure.

HITS returns as hubs and authorities for the search topic those pages with the highest weights.

We view the Web as a directed graph, consisting of a set of nodes with directed edges between certain node pairs. Given any subset S of nodes, the nodes induce a subgraph containing all edges that connect two nodes in S. The HITS algorithm starts by constructing the subgraph in which we will search for hubs and authorities. Our goal is to have a subgraph rich in relevant, authoritative pages.

To construct such a subgraph, we first use the query terms to collect a root set of pages—say, 200—from an index-based search engine. We do not expect that this set necessarily contains authoritative pages. However, since many of these pages are presumably relevant to the search topic, we expect at least some of them to have links to most of the prominent authorities. We therefore expand the root set into a base set by including all the pages that the root-set pages link to, and all pages that link to a page in the root set, up to a designated size cutoff.

This approach follows our intuition that the prominence of authoritative pages derives typically from the endorsements of many relevant pages that are not, in themselves, prominent. We restrict our attention to this base set for the remainder of the algorithm. We find that this set typically contains from 1,000 to 5,000 pages, and that hidden among these are many pages that, subjectively, can be viewed as authoritative for the search topic.

Our techniques for uncovering authorities and hubs can uncover Web communities, defined by a specific interest, that even a human-assisted search engine may overlook.

We work with the subgraph induced by the base set, with one modification. We find that links between two pages with the same Web domain frequently serve a purely navigational function, and thus do not confer authority. By "Web domain," we mean simply the first level in the URL string associated with a page. We therefore delete all links between pages with the same domain from the subgraph induced by the base set, and then apply the remainder of the algorithm to this modified subgraph.

We extract good hubs and authorities from the base set by giving a concrete numerical interpretation to our intuitive notions of authorities and hubs. We associate a nonnegative authority weight x_p and a nonnegative hub weight y_p with each page $p \in V$. We are interested in the relative values of these weights only, not their actual magnitudes. In our manipulation of the weights, we apply a normalization so that their total sum remains bounded. The actual choice of normalization does not affect the results—we maintain the invariant that the squares of all weights sum to 1. A page p with a large weight x_p will be viewed as a "better" authority, while a page with a large weight y_p will be viewed as a "better" hub. Since we do not impose any a priori estimates, we set all x and y values to a uniform constant initially; we will see later, however, that the final results are essentially unaffected by this initialization.

We now update the authority and hub weights as follows. If a page is pointed to by many good hubs, we would like to increase its authority weight. Thus we update the value of x_p, for a page p, to be the sum of y_q over all pages q that link to p:

$$x_p = \sum_{q \text{ such that } q \to p} y_q, \qquad \textbf{(1)}$$

where the notation $q \to p$ indicates that q links to p. In a strictly dual fashion, if a page points to many good authorities, we increase its hub weight via

$$y_p = \sum_{q \text{ such that } p \to q} x_q. \qquad \textbf{(2)}$$

There is a more compact way to write these updates, and it sheds more light on what occurs mathematically. Let us number the pages $\{1, 2, \ldots, n\}$ and define their *adjacency matrix* A to be the $n \times n$ matrix whose $(i,j)^{\text{th}}$ entry is equal to 1 if page i links to page j, and is 0 otherwise. Let us also write the set of all x values as a vector $x = (x_1, x_2, \ldots, x_n)$, and similarly define $y = (y_1, y_2, \ldots, y_n)$. Then our update rule for x can be written as $x \leftarrow A^T y$ and our update rule for y can be written as $y \leftarrow Ax$. Unwinding these one step further, we have

$$x \leftarrow A^T y \leftarrow A^T A x = (A^T A)x \qquad \textbf{(3)}$$

and

$$y \leftarrow Ax \leftarrow AA^T y = (AA^T)y. \qquad \textbf{(4)}$$

Thus, the vector x after multiple iterations is precisely the result of applying the power iteration technique to A^TA: We multiply our initial iterate by larger and larger powers of A^TA. Linear algebra tells us that this sequence of iterates, when normalized, converges to the principal eigenvector of A^TA. Similarly, we discover that the sequence of values for the normalized vector y converges to the principal eigenvector of AA^T. Gene Golub and Charles Van Loan[4] describe this relationship between eigenvectors and power iteration in detail.

Power iteration will converge to the principal eigenvector for any *nondegenerate* choice of initial vector—in our case, for example, for any vector whose entries are all positive. This says that the hub and authority weights we compute are truly an intrinsic feature of the linked pages collected, not an artifact of our choice of initial weights or the tuning of arbitrary parameters. Intuitively, the pages with large weights represent a very *dense* pattern of linkage, from pages of large hub weight to pages of large authority weight.

Finally, HITS outputs a short list consisting of the pages with the largest hub weights and the pages with the largest authority weights for the given search topic. Once the root set has been assembled, HITS is a purely link-based computation with no further regard to the query terms. Nevertheless, HITS provides surprisingly good search results for a wide range of queries. For example, when tested on the sample query "search engines," HITS returned the top authorities—Yahoo, Excite, Magellan, Lycos, and AltaVista—even though none of these pages contained the phrase "search engines" at the time of the experiment. Results such as this confirm our intuition that in many cases the use of hyperlinks can help circumvent some of the difficulties inherent in purely text-based search methods.

Our techniques for uncovering authorities and hubs provide a further benefit. As the "Trawling the Web for Emerging Cybercommunities" sidebar shows, our algorithms can uncover Web communities, defined by a specific interest, that even a human-assisted search engine like Yahoo may overlook.

COMBINING CONTENT WITH LINK INFORMATION

Although relying extensively on links when searching for authoritative pages offers several advantages, ignoring textual content after assembling the root set can lead to difficulties. These difficulties arise from certain features of the Web that deviate from the pure hub-authority view:

- On narrowly focused topics, HITS frequently returns good resources for a more general topic. For instance, the Web does not contain many resources for skiing in Nebraska; a query on this topic will typically generalize to Nebraska tourist information.
- Since all the links out of a hub page propagate the same weight, HITS sometimes drifts when hubs discuss multiple topics. For instance, a chemist's home page may contain good links not only to chemistry resources, but also to resources for her hobbies and regional information for her hometown. In such cases, HITS will confer some of the "chemistry" authority onto authorities for her hobbies and town, deeming these authoritative pages for chemistry.
- Frequently, many pages from a single Web site will take over a topic simply because several of the pages occur in the base set. Moreover, pages from the same site often use the same HTML design template, so that in addition to the information they give on the query topic, they may all point to a single popular site that has little to do with the query topic. This inadvertent topic hijacking can give a site too large a share of the authority weight for the topic, regardless of the site's relevance.

System heuristics

The Clever system addresses these issues by replacing the sums of Equations 1 and 2 with weighted sums, assigning to each link a nonnegative weight. The weight assigned depends in several ways on the query terms and the endpoints of the link. Together with some additional heuristics, weighting helps mitigate HITS' limitations.

The text that surrounds hyperlink definitions (hrefs) in Web pages is often referred to as *anchor text*. In our setting, we choose to use anchor text to weight the links along which authority is propagated. A typ-

Trawling the Web for Emerging Cybercommunities

The Web harbors many communities—groups of content creators who share a common interest that manifests itself as a set of Web pages. Though many communities are defined explicitly—newsgroups, resource collections in portals, and so on—many more are implicit. Using a subgraph-enumeration technique called trawling, we discovered fine-grained communities numbering in the hundreds of thousands—many more than the number of portals and newsgroup topics. The following communities are a sampling of those we have extracted from the Web:

- people interested in Hekiru Shiina, a Japanese pop singer;
- people who maintain information about fire brigades in Australia; and
- people belonging to Turkish student organizations in the US.

Identifying these communities helps us understand the intellectual and sociological evolution of the Web. It also helps provide detailed information to groups of people with certain focused interests. Owing to these communities' astronomical number, embryonic nature, and evolutionary flux, they are hard to track and find through sheer manual effort. Thus, when uncovering communities, we treat the Web as a huge directed graph, use graph structures derived from the basic hub-authority-linkage pattern as a community's "signature," and systematically scan the Web graph to locate such structures.

We begin with the assumption that thematically cohesive Web communities contain at their core a dense pattern of linkage from hubs to authorities. The pattern ties the pages together in the link structure, even though hubs do not necessarily link to hubs, and authorities do not necessarily link to authorities. We hypothesize that this pattern is a characteristic of both well-established and emergent communities. To frame this approach in more graph-theoretic language, we use the notion of a directed bipartite graph—one whose nodes can be partitioned into two sets A and B such that every link in the graph is directed from a node in A to a node in B. Since the communities we seek contain directed bipartite graphs with a large density of edges, we expect many of them to contain smaller bipartite subgraphs that are in fact complete: Each node in A has a link to each node in B.

Using a variety of pruning algorithms,[1] we can enumerate all such complete bipartite subgraphs on the Web using only a standard desktop PC and about three days of runtime. In our experiments to date, we have used an 18-month-old crawl of the Web provided by Alexa (www.alexa.com), a company that archives Web snapshots. The process yielded about 130,000 complete bipartite graphs in which three Web pages all pointed to the same set of three other Web pages.

Were these linkage patterns coincidental? Manual inspection of a random sample of about 400 communities suggests otherwise: Fewer than five percent of the communities we discovered lacked an apparent unifying topic. These bipartite cliques could then be fed to our HITS algorithms. These algorithms "expanded" the cliques to many more Web pages from the same community.

Moreover, Yahoo does not list about 25 percent of these communities, even today. Of those that do appear, many are not listed until the sixth level of the Yahoo topic tree. These observations lead us to believe that trawling a current copy of the Web will result in the discovery of many more communities that will become explicitly recognized in the future.

Reference

1. S.R. Kumar et al., "Trawling Emerging-Cyber-Communities Automatically," *Proc. 8th World Wide Web Conf.*, Elsevier Science, Amsterdam, 1999, pp. 403-415.

Our study results suggest that Clever can be used to compile large topic taxonomies automatically.

ical example shows why we do so: When we seek authoritative pages on chemistry, we might reasonably expect to find the term "chemistry" in the vicinity of the tails—or anchors—of the links pointing to authoritative chemistry pages. To this end, we boost the weights of links in whose anchor—a fixed-width window—query terms occur.

We base a second heuristic on breaking large hub pages into smaller units. On a page containing many links, it is likely that not all links focus on a single topic. In such situations it becomes advantageous to treat contiguous link subsets as minihubs, or pagelets; we can then develop a hub score for each pagelet, down to the level of single links. We hypothesize that contiguous sets of links on a hub page focus more tightly on a single topic than does the entire page. For instance, a page may be a good hub for the general topic of "cars," but different portions of it may cater to the topics of "vintage cars" and "solar-powered cars."

We apply one further set of modifications to HITS. Recall that HITS deletes all links between two pages within the same Web domain. Because we work with weighted links, we can address this issue through our choice of weights. First, we give links within a common domain low weight, following the rationale that authority should generally be conferred globally rather than from a local source on the same domain. Second, when many pages from a single domain participate as hubs, we scale down their weights to prevent a single site from becoming dominant.

All these heuristics can be implemented with minimal effort and without significantly altering the mathematics of Equations 1 through 4. The sums become weighted sums, and matrix **A** now has nonnegative real-valued entries rather than just 0s and 1s. As before, the hub and authority scores converge to the components of the principal eigenvectors of AA^T and A^TA, respectively. In our experience, the relative values of the large components in these vectors typically resolve themselves after about five power iterations, obviating the need for more sophisticated eigenvector computation methods.

COMPARING CLEVER WITH OTHER SEARCH ENGINES

How do the resources computed by Clever compare with those found by other methods? We have conducted several user studies that compare Clever's compilations with those generated by AltaVista (www.altavista.com), a term-index engine, and by Yahoo (www. yahoo.com), a manually compiled topic taxonomy in which a team of human ontologists create resource lists.

In one such study,[2] which compares Clever with Yahoo and AltaVista, we began with a list of 26 broad search topics. For each topic, we took the top 10 pages from AltaVista, the top five hubs and five authorities returned by Clever, and a random set of 10 pages from Yahoo's most relevant node or nodes. We then interleaved these three sets into a single topic list, masking which method produced which page. Next, we assembled 37 users, who were required to be familiar with using Web browsers but who were not experts in computer science or in the 26 search topics. We then asked the users to visit pages from the topic lists and rank them as "bad," "fair," "good," or "fantastic," in terms of the pages' utility in providing information about the topic. This yielded 1,369 responses in all, which were then used to assess the relative quality of Clever, Yahoo, and AltaVista on each topic. AltaVista failed to receive the highest evaluation for any of the 26 topics. For the other search engines, we obtained the following results:

- For 31 percent of the topics, Yahoo and Clever received evaluations equivalent to each other within a threshold of statistical significance;
- for 50 percent, Clever received a higher evaluation; and
- for the remaining 19 percent, Yahoo received the higher evaluation.

In masking the source from which each page was drawn, this experiment denied Yahoo one clear advantage of a manually compiled topic list: the editorial annotations and one-line summaries that give powerful cues for deciding which link to follow. We did this deliberately because we sought to isolate and study the power of different paradigms for resource finding, rather than for the combined task of compilation and presentation. In an earlier study[1] we did not mask these annotations, and Yahoo's combination of links and presentation beat an early version of Clever.

CONSTRUCTING TAXONOMIES SEMIAUTOMATICALLY

Yahoo's large taxonomy of topics consists of a subject tree, each node of which corresponds to a particular topic and which is populated by relevant pages. Our study results suggest that Clever can be used to compile such large topic taxonomies automatically.

Suppose we are given a tree of topics designed by domain experts. The tree can be specified by its topology and the labels on its nodes. We wish to populate each node of the tree with a collection of the best hubs and authorities. The following paradigm emerges: If we can effectively describe each node of the tree as a query to Clever, the Clever engine could then populate the node as often as we please. For instance, the

Assigning Web Pages to Categories

In addition to finding hubs, authorities, and communities, hyperlinks can be used to categorize Web pages. Categorization is a process by which a system learns from examples to assign documents to a set of predefined topic categories such as those found in a taxonomy. Hyperlinks contain high-quality semantic clues to a page's topic; these clues are lost when the links are processed by a purely term-based categorizer. Exploiting this link information is challenging, however, because it is highly noisy. Indeed, we have found that naive use of terms in a document's link neighborhood can *degrade* accuracy.

HyperClass[1] embodies one approach to this problem, making use of robust statistical models such as Markov random fields (MRFs) together with a relaxation labeling technique. HyperClass obtains improved categorization accuracy by exploiting link information in the neighborhood around a document. The MRF framework applies because pages on the same or related topics tend to be linked more frequently than those on unrelated topics. Even if none of the linked pages' categories are known initially, you can obtain significant taxonomy improvement using relaxation labeling, wherein you iteratively adjust the category labels of the linked pages and of the page to be categorized until you find the most probable configuration of class labels. In experiments performed[1] using preclassified samples from Yahoo and the US Patent Database (www.ibm.com/patents), HyperClass with hyperlinks cut the patent error rate by half and the Yahoo documents error rate by two thirds.

HyperClass is also used in a focused Web crawler[2] designed to search for pages on a particular topic or set of topics only. By categorizing pages as it crawls, the focused crawler does more than filter out irrelevant pages—it also uses the associated relevance judgment, as well as a rank determined by a version of the Clever algorithm, to set the crawling priority of the outlinks on the pages it finds.

References

1. S. Chakrabarti, B. Dom, and P. Indyk, "Enhanced Hypertext Classification Using Hyper-links," *ACM SIGMOD Int'l Conf. Management of Data*, ACM Press, New York, 1998, pp. 307-318.
2. S. Chakrabarti, B. Dom, and M. van den Berg, "Focused Crawling: A New Approach for Topic-Specific Resource Discovery," *Proc. 8th World Wide Web Conf.*, Elsevier Science, Amsterdam, 1999, pp. 545-562.

resources at each node could be refreshed on a nightly basis following the one-time human effort of describing the topics. How, then, should we describe a topic node to Clever?

Most simply, we may take the name or label of the node as a query term. More generally, we may wish to use the descriptions of other nodes on the path to the root. For instance, if the topic headings along a root-to-leaf path are Business/Real Estate/Regional/United States/Oregon, the query "Oregon" is not accurate; we might prefer instead the query "Oregon real estate."

Additionally, we may provide some exemplary authority or hub pages for the topic. For instance, the sites www.att.com and www.sprint.com may be exemplary authority pages for the topic "North American telecommunications companies." In practice, we envision a taxonomy administrator first trying a simple text query to Clever. Often this query will yield a good collection of resources, but other times Clever may return a mix of high-quality and irrelevant pages. In such cases, the taxonomy administrator may highlight some of the high-quality pages in the Clever results as exemplary hubs, exemplary authorities, or both. This is akin to the well-studied technique of relevance feedback in information retrieval.

To take advantage of exemplary pages, we add an exemplary hub to the base set, along with all pages that it points to, and then increase the weights of the links emanating from the exemplary hub in the iterative computation. We treat exemplary authorities similarly, except that instead of adding to the base set any page pointing to an exemplary authority—a heuristic found to pull in too many irrelevant pages—we add any page pointing to at least *two* exemplary authorities. We use a similar heuristic to delete from the base set user-designated "stop-sites" and their link neighborhoods. This is typically necessary because of the overwhelming Web presence of certain topics. For instance, if our topic is Building and Construction Supplies/Doors and Windows, the "Windows" keyword makes it difficult to ignore Microsoft. Stop-siting www.microsoft.com eliminates this concern.

Thus, we may envision a topic node being described to Clever as a combination of query terms, exemplified authority and hub pages, and, optionally, stop-sites. We have developed a Java-based graphical user interface—called "TaxMan," for Taxonomy Manager—to administer such taxonomy descriptions. Using this tool, we have constructed taxonomies with more than a thousand topics. We have benchmarked both the time spent in creating these taxonomies and the resultant quality of using simple text-only queries versus a combination of text queries and exemplary Web pages. In our study, we found that the average time spent per node grows from about seven seconds to roughly three minutes when you move to a combination of text and exemplary page queries. Outside users quantified the increase in quality by reporting that—when comparing the pages generated using exemplaries to pages generated by textual queries—they considered eight percent more of the exemplary pages to be good link sources.

The "Assigning Web Pages to Categories" sidebar describes how hyperlinks can be used to establish clearer taxonomy categories as well.

CITATION ANALYSIS

The mining of Web link structures has intellectual antecedents in the study of social networks and citation analysis.[5] The field of citation analysis has developed several link-based measures of scholarly papers' importance, including the impact factor and influence weights.[5] These measures in effect identify authorita-

Our analysis of hyperlink topology focuses on the extraction of densely connected regions in the link structure.

tive sources without introducing the notion of hubs. The view of hubs and authorities as dual sets of important documents is inspired by the apparent nature of content creation on the Web, and indicates some of the deep contrasts between Web and scholarly literature content.

The methodology of influence weights from citation analysis relates to a link-based search method developed by Sergey Brin and Lawrence Page.[6] They used this method as the basis for their Google Web search engine. Google first computes a score, called the PageRank, for every page indexed. The score for each page is the corresponding component of the principal eigenvector of a matrix **B**, which can be viewed as the adjacency matrix **A** with a very small constant added to each entry. Given a query, Google returns pages containing the query terms, ranked in order of these pages' PageRanks.

The actual implementation of Google incorporates several additional heuristics, similar in intent and spirit to those used for deriving Clever from HITS. Google focuses on authoritative pages, however, while Clever seeks both authorities *and* good hub pages. Some hub pages may have few or no links into them, giving them low PageRank scores and making it unlikely that Google would report them. Several participants in our user studies suggested that good hubs are especially useful when trying to learn about a new topic, but less so when seeking a very specific piece of information. Google and Clever also differ in their behavior toward topics with a commercial theme. A company's Web-page description of itself may use terms and language different from these that a user might search for. Thus, a direct search for "mainframes" in Google would not return IBM's home page, which does not contain the term "mainframes." Yet IBM would still be pulled in by Clever because of the many hub pages that describe IBM as a mainframe manufacturer.

In independent work, Krishna Bharat and Monika R. Henzinger[7] have given several other extensions to the basic HITS algorithm, substantiating their improvements via a user study. For instance, their paper was the first to describe the modification in which the weights of multiple links from within a site are scaled down.

We believe the mining of Web link topology has the potential for beneficial overlap with several areas, including the field of information retrieval.[8] Mining well-structured relational data offers another possibility. Extracting from an unstructured medium such as the Web a structure of the kind that succumbs to traditional database techniques[9] presents a considerable challenge. We hope that the techniques described here represent a step toward meeting this challenge. ❖

References

1. S. Chakrabarti et al., "Automatic Resource Compilation by Analyzing Hyperlink Structure and Associated Text," *Proc. 7th World Wide Web Conf.*, Elsevier Science, Amsterdam, 1998, pp. 65-74.
2. S. Chakrabarti et al., "Experiments in Topic Distillation," *SIGIR Workshop Hypertext Information Retrieval*, ACM Press, New York, 1998, http://www.almaden.ibm.com/cs/k53/clever.html.
3. J.M. Kleinberg, "Authoritative Sources in a Hyperlinked Environment," *Proc. 9th ACM-SIAM Symp. Discrete Algorithms*, ACM Press, New York and SIAM Press, Philadelphia, 1998, pp. 668-677.
4. G. Golub and C.F. Van Loan, *Matrix Computations*, Johns Hopkins University Press, Baltimore, 1989.
5. L. Egghe and R. Rousseau, *Introduction to Informetrics*, Elsevier Science, Amsterdam, 1990.
6. S. Brin and L. Page, "The Anatomy of a Large Scale Hypertextual Web Search Engine," *Proc. 7th World Wide Web Conf.*, Elsevier Science, Amsterdam, 1998, pp. 107-117.
7. K. Bharat and M.R. Henzinger, "Improved Algorithms for Topic Distillation in a Hyperlinked Environment," *Proc. SIGIR 98*, ACM Press, New York, 1998, pp. 104-111.
8. G. Salton and M. McGill, *Introduction to Modern Information Retrieval*, McGraw-Hill, New York, 1983.
9. D. Florescu, A. Levy, and A. Mendelzon, "Database Techniques for the World Wide Web: A Survey," *SIGMOD Record*, Vol. 27, No. 3, 1998, pp. 59-74.

Soumen Chakrabarti is an assistant professor in the Department of Computer Science and Engineering at the Indian Institute of Technology, Bombay. His research interests include hypertext information retrieval, Web analysis, and data mining. Chakrabarti received a BTech in computer science from the Indian Institute of Technology, Kharagpur, and an MS and a PhD in computer science from the University of California, Berkeley.

Byron Dom is a manager of information management principles at IBM Almaden Research Center. His research interests are in information retrieval, machine learning, computer vision, and information theory. Dom received a PhD in applied physics from The Catholic University of America.

S. Ravi Kumar is a research staff member at the IBM Almaden Research Center. His research interests include randomization, complexity theory, and infor-

mation processing. Kumar received a PhD in computer science from Cornell University.

Prabhakar Raghavan is a research staff member at the IBM Almaden Research Center and a consulting professor at Stanford University's Computer Science Department. His research interests include algorithms, randomization, and information retrieval and optimization. Raghavan received a PhD in computer science from the University of California, Berkeley.

Sridhar Rajagopalan is a research staff member at the IBM Almaden Research Center. His research interests include algorithms and optimization, randomization, information and coding theory, and information retrieval. Rajagopalan received a BTech from the Indian Institute of Technology, Delhi, and a PhD from the University of California, Berkeley.

Andrew Tomkins is a research staff member for the IBM Almaden Research Center's Principles and Methodologies group. His research interests include algorithms in general and online algorithms in particular, disk scheduling and prefetching, pen computing and OCR, and the Web. Tomkins received a BSc in mathematics and computer science from MIT and a PhD in computer science from Carnegie Mellon University.

David Gibson is a part-time researcher at IBM's Almaden Research Center and a PhD student at the University of California, Berkeley, where he pursues fundamental aspects of computer science and experimental computation.

Jon Kleinberg is an assistant professor in the Department of Computer Science at Cornell University. His research interests include algorithms, discrete optimization, and combinatorial methods for clustering and indexing. He is supported in part by an Alfred P. Sloan Research Fellowship, an NSF Faculty Early Career Development Award, and an Office of Naval Research Young Investigator Award. Kleinberg received a PhD in computer science from MIT.

Contact Kleinberg at Department of Computer Science, Cornell University, Ithaca, NY 14853; kleinber@CS.Cornell.edu. Contact Chakrabarti at soumen@cse.iitb.ernet.in. Contact Gibson at dag@cs.berkeley.edu. Contact Dom, Kumar, Raghavan, Rajagopalan, and Tomkins at {dom, ravi, pragh, sridhar, tomkins}@almaden.ibm.com.