Spectral filtering for resource discovery

Soumen Chakrabarti Byron E. Dom David Gibson * Ravi Kumar
Prabhakar Raghavan Sridhar Rajagopalan Andrew Tomkins

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

Abstract

We develop a technique we call *spectral filtering*, for discovering high-quality topical resources in hyperlinked corpora. Through relevance and quality judgements collected from 37 users, we show that, over 26 topics, spectral filtering usually finds web pages that are rated better than those returned by the hand-compiled Yahoo! resource list, and by the Altavista search engine.

1 Introduction

As the amount of information available on-line grows exponentially, the user's capacity to digest that information cannot keep pace [24, 30]. In this paper we address the problem of distilling high-quality sources of information on broad topics in hyperlinked corpora (such as the WWW). As an example, we seek algorithms that can answer the question: "What are the twenty best pages on the web about History?" There are over five million pages on the web today that contain the word history; we seek only twenty, but they must be exemplary.

The authors of [3] present a system called ARC which addresses the same problem; our results differ in two respects. First, the algorithms presented here are more general and, guided by the lessons learned in [3] and subsequently, more effective. Second, the earlier paper compared ARC to two familiar web engines. Subjects performing the comparison visited all three sites and made judgements based on quality of resources *and* issues of presentation within each engine. In the current study, we decouple the presentation of a page from its content. Thus, we consider only whether an engine contains links to high-quality resources, and leave other value added by the engine (such as brief annotations describing each page) as an orthogonal component of engine quality. We discuss these distinctions in further detail, below.

Our approach to resource discovery is a general technique called *spectral filtering*, which is based on the spectral properties of matrices derived from relationships within the corpus (Section 2). Unlike traditional web search, spectral filtering estimates the quality of a page using both the *content* of the page, and the *context* of the page: the pages it points to, the pages that point to it, and the web neighborhood in which it appears. Moreover, spectral filtering circumvents the computational bottlenecks associated with numerical and algebraic methods such as LSI.

For a set of 26 broad topics benchmarked previously by [1, 3],[1] we compare spectral filtering to Altavista, a popular web search engine, and to Yahoo!, a taxonomy of hand-generated lists of resources on a large number of topics. To perform this comparison we need a metric that does not require a benchmark corpus (such as TREC [25]) for which expert relevance judgements are readily available. We adopt *comparitive precision*, which is akin to traditional precision except that relevance judgements are performed by users on the set of pages returned by spectral filtering, Altavista, and Yahoo!, rather than on the entire web (Section 3.1). It is important to note that the users do not know which system generated which page. Without judgements for a substantial fraction of the entire web, we cannot compute an analog to recall; so following [1], we adopt comparitive precision of a fixed number of pages as our metric.

Under this metric, spectral filtering performs substantially better than Altavista, and typically outperforms the hand-constructed resources of Yahoo! as well (Section 3.3). This latter result seems surprising, but spectral filtering has three advantages over Yahoo!. First, an automatic

*Department of Computer Science, University of California, Berkeley. This work was done while the author was at IBM Almaden Research Center.

[1] Prior studies contained 28 topics. We omitted one query, "architecture," because our evaluators could not be sure whether we meant "buildings" or "computers"; and a second query, "Zen Buddhism," because none of our evaluators for that query completed the evaluation.

system can consider many more candidate pages than a manual approach. Second, spectral filtering uses the human effort implicit in the largely hand-built link-structure of the WWW, so while the search is fully automatic, it incorporates human judgements. And third, the set of 26 topics was generated without reference to any search engine or site, so while some of the topics correspond exactly to Yahoo! nodes, some do not.

2 The spectral filtering method

Kleinberg [13] builds on notions from bibliometry [26] in developing the HITS technique for searching hypertext. Since spectral filtering generalizes HITS, we begin with a review (Section 2.1) of the technique. Following this, we develop spectral filtering (Section 2.2) and specialize it for the web (Section 2.3). We then show how this method may be specialized to yield a variety of application-dependent searching and clustering techniques (Section 2.4). Finally, we discuss some computational issues (Section 2.5) and describe prior work (Section 2.6).

2.1 An overview of the HITS technique

HITS performs resource discovery on the web, producing two distinct but related types of pages in response to a topic query: *hubs* and *authorities*. Hubs and authorities exhibit a mutually reinforcing relationship: a good hub points to many good authorities, and a good authority is pointed to by many good hubs.[2] The algorithm begins by constructing a *root set* of pages that are likely to be relevant to the topic. This construction is arbitrary, but could be performed by creating an *initial set* of pages by querying a traditional search engine using the topic as query, and then expanding this set to the full root set by including all pages that point to, or are pointed to by, a page in the initial set (Figure 1). The algorithm then associates with each page p a *hub-weight* $h(p)$ and an *authority-weight* $a(p)$, all initialized to 1. HITS then iteratively updates these weights as follows:

$$a(p) := \sum_{q \to p} h(q), \qquad h(p) := \sum_{p \to q} a(q).$$

To restate using linear algebra, let $A = [a_{ij}]$ denote the adjacency matrix of the directed graph of hyperlinks connecting the pages in the base set: $a_{ij} = 1$ if page i has a link to page j, and 0 otherwise. HITS may then be viewed as repeatedly iterating the following matrix operations on the vectors $\boldsymbol{h}$ and $\boldsymbol{a}$ corresponding to the hub and authority scores of all pages:

$$\boldsymbol{h} \leftarrow A\boldsymbol{a}, \qquad \boldsymbol{a} \leftarrow A^T\boldsymbol{h} \tag{1}$$

[2] Pages can be both good authorities and good hubs.

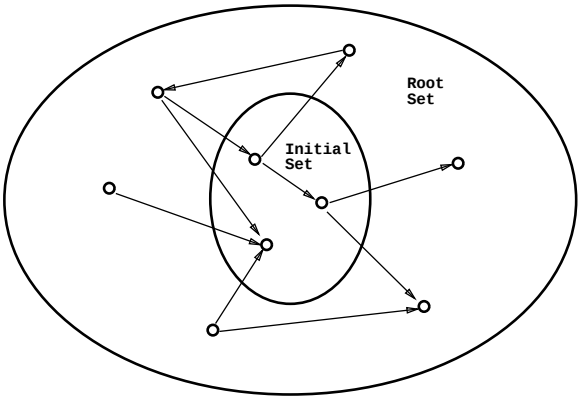

Figure 1: Expanding the initial set into a root set.

From classical matrix theory [12], it follows that with appropriate renormalization, $\boldsymbol{h}$ (resp. $\boldsymbol{a}$) converges to the principal eigenvector of AA^T (resp. A^TA). Kleinberg further points out that by analogy with *spectral graph partitioning* [9], the *non-principal* eigenvectors of AA^T and A^TA can be used to partition the pages into groups of related hubs and authorities, respectively.

2.2 Spectral filtering

Rather than focusing on web pages, in spectral filtering we consider arbitrary *entities*. These may be web pages, documents, terms, or other structures. While HITS exploits the annotative power latent in hyperlinks, we wish to think more generally in terms of "what does entity i say about entity j?" To quantify this, we define a numerical *affinity* from i to j denoted a_{ij}. At a high level, our method consists of three steps:

1. acquisition of the root set S of entities to be analyzed. In many applications of spectral filtering this process consists of obtaining an initial set via a Boolean keyword search and then expanding it to include neighbors (one link distance away);

2. approximate calculation of one or more of the eigenvectors of one or both of two possible similarity matrices (defined below);

3. analysis of the computed eigenvector(s) to rank and/or partition the set of entities.

We now give the algorithm more formally for the case of discovery of authoritative sources, which corresponds to the analysis of principal eigenvectors. Section 2.4 extends to applications that require non-principal eigenvectors. Let $n = |S|$ and let a_{ij} be a non-negative real-valued *affinity* of the ordered pair of entities (i, j), so a_{ij} need not equal a_{ji}. Typically, each a_{ij} is a carefully-chosen

function of the terms and (where applicable) links in the entities; this choice is corpus- and application-dependent. (In HITS, for instance, a_{ij} is a boolean value indicating the presence of an edge from page i to page j.) Let $A = [a_{ij}]$. We perform exactly the iterations of (1) to arrive at hub and authority scores converging to the principal eigenvectors (those associated with the largest eigenvalue) of $A^T A$ and AA^T, respectively (we call these *similarity matrices*).[3] Then, we output the entities with the largest entries in the principal eigenvector of $A^T A$ (resp. AA^T) as the top authorities (resp. hubs).

2.3 Spectral filtering for web pages

In this section we adapt spectral filtering to the WWW, identifying and addressing some problems with the simple affinity function used in HITS. The generation of the root set follows the description of Section 2.1. Let $a_{ij} = 0$ if there is no link from page i to page j, and is positive if a link exists. The value of the affinity is a sum of three components: the first is a default value given to every link, the second component depends on which, if any, of page i and j fall within the initial set, and the third has a contribution from each query term. The contribution of a query term appearing at distance i within a window W of terms from the hyperlink is proportional to $W - i$. Query terms within quotes are treated as atomic units, so the word "car" generates no contribution for the query "vintage car."

We also modify the basic algorithm in several ways. Each modification is motivated by a specific idiosyncrasy of the web. We describe these modifications now.

- *Same-site Pages:* To avoid *self-promotion*, or web sites that confer authority upon themselves, we discard links to pages on the same site. We define two pages to be on the *same* website using the following heuristic: class A and B IP addresses must match two most significant octets; class C addresses must match three most significant octets, and class D addresses must match all four octets.

- *Covering Heuristic:* The value of a hub page is by definition in its links rather than its contents. If all the destinations accessible from a particular hub are also accessible from better hubs, we do not need to output this hub. More generally, we seek to return a set of hub pages that together contain as many unique, high-quality links as possible. We therefore apply a well-known greedy set-cover [15] heuristic as follows. Once the iteration step has converged, we repeat the following process until we have generated the correct number of hubs: return the best hub, zero the authority values of all pages pointed to by the hub, and recompute hub values.

- *Packing Heuristic:* Despite the same-site link removal heuristic, it is possible for instance for an organization's homepage, and several children of that page, to accumulate authority. However, in the final output we wish to provide the user with as much authoritative substance as possible in a small number of pages. To achieve this, after each step of the iteration we "re-pack" the authority of any site, as follows: if multiple documents within a logical site (as defined above) have non-zero authority, the authorities of all but the page with the largest authority are set to zero.

- *Hub Functions:* Many resource compilations (e.g., bookmark files) contain pages pertinent to a number of disjoint topics. This causes such compilations to become good hubs, which in turn causes irrelevant links from the same page to become good authorities. To address this problem we note that pointers to pages on the same topic tend to be clustered together in resource compilations. We allow each link in a web page to have its own hub value, so the hub value of a page is now a function of the particular link rather than a constant. When computing authority values, the authority of the destination is incremented by the hub value of the link. When recomputing hub values, the authority value of the destination is used to increment the hub value of the source link, and according to a spreading function, the hub values of neighboring links. Thus, useful regions of a large hub page can be identified. The final hub value of a page is the integral of the hub values of its links.

Convergence of the spectral filtering computation presented in Section 2 depends on phrasing the iterated steps as a matrix multiplication. We must determine whether the modifications described above still fit the framework. Notice that the same-site and covering heuristics are simply pre- and post-processing steps, and the hub function heuristic is a linear transformation that may be expressed as a matrix multiplication. With these heuristics, we still have guaranteed convergence. But packing heuristics are non-linear, so we have a nonlinear dynamical system whose convergence is not guaranteed. In practice, however, the results converge rapidly in all cases we have considered.

[3] The matrix A as presented contains affinities between entities of the same type. A straightforward generalization gives affinities between entities of different types, e.g., a term-document matrix. In this case, the rows of A could correspond to terms, and the columns to documents. Although A may not be square, $A^T A$ and AA^T are square and symmetric.

2.4 Spectral filtering for other domains

Spectral filtering is a technique we have developed for the web, but it applies more generally to other corpora and to other tasks. We briefly discuss these applications, and also compare the approach to LSI.

Other corpora We mention in passing that we have successfully applied to a number of corpora besides the WWW. For *non-hyperlinked corpora*, one way we have applied spectral filtering is by defining the affinity function as follows: for documents i, j let $|i \cap j|$ denote the number of terms they have in common. Let $a_{ij} = |i \cap j|/|i|$, where $|i|$ denotes the number of terms in i.

We have also applied spectral filtering to *time-serial corpora* such as the US Patent database and the Supreme Court rulings. Here the citations only go backwards in time; the fact that the iterations in spectral filtering go back and forth across (possibly weighted) links is crucial in extracting structure. If, for instance, one were to only iterate along citations (but never in the reverse direction), all the authority would end up in the oldest cases/patents.

Other applications Spectral partitioning can be used for clustering and partitioning either a corpus or a selected subset as follows. Having set up the matrix A as before, we can also compute the non-principal eigenvectors of $A^T A$. Since $A^T A$ is real and symmetric, its eigenvectors are real. We can view the components of each non-principal eigenvector as assigning to each document a position on the real line. We examine the values in the eigenvector (in sorted increasing order). At the largest gap between successive values, we declare a partition into those documents corresponding to values above the gap, and those documents with values below. This is illustrated in Figure 2. We may view the entries of $A^T A$ as (symmetric) "authority similarities" between documents, and likewise those of AA^T as "hub similarities". Intuitively, the eigenvector operations serve to pull together groups of documents that are all close to one another under the authority (or hub) similarity function.

Spectral filtering also applies to the problem of collaborative filtering [11], though we do not provide experimental data. Consider a setting with two kinds of entities: documents, and people who access them (the precise notion of "access" may be application-dependent). For person i and document j, let $a_{ij} = 1$ if i accesses j and 0 otherwise (a_{ij} could be some non-negative function such as the frequency of access). Now, partitioning using the non-principal eigenvectors would group the people into subsets with similar document-access patterns, and also group the documents into subsets. More generally, the "documents" could be products or other preferences expressed by the people.

Finally, we mention a closely-related application that uses the resulting eigenvectors of similarity matrices for a different purpose. Latent Semantic Indexing[10] (LSI) is a dimensionality reduction technique based on SVD[12] that captures the "latent semantic structure" of a corpus. LSI starts with a term-document matrix A, performs an SVD of A, and uses the subspace spanned by the first few (say 100) singular vectors for information retrieval. Both documents and queries are projected into the "document" subspace, where their similarity is measured by, say, the inner product between the two projected vectors. The similarity between LSI and spectral filtering is clear; they differ in the way the eigenvectors are used.

2.5 Computational issues

The performance of numerical eigenvector computations is often a bottleneck, especially in dealing with large corpora. Spectral filtering avoids this bottleneck for three reasons:

1. Numerical convergence is not our goal when we wish to rank/group documents by their scores in eigenvectors. Rather, it is the relative ranks of the eigenvector entries that matter. Typically, 5 iterations suffice to stabilize the ranks of hubs and authorities, far fewer than required for numerical convergence.

2. The computation is restricted to a relevant subset of the corpus.

3. A is typically very sparse.

2.6 Related prior work

Hyperlinks have been explicitly used in information retrieval for *ranking* (Section 2.6.1) and for *structure discovery* (Section 2.6.2). For most information retrieval tasks, ranking provides an ordering of documents based on relevance to a query. In contrast, the ranking induced by our system tries to capture perceived quality with respect to a query. Structure discovery, on the other hand, refers to tasks like clustering and the identification of aggregates in hypertext.

Each of these two categories can be further divided corresponding to the following four types of attributes used to perform the operations: *text*, *bibliographic citations*, *hyperlinks*, and *hypertext*.

2.6.1 Ranking

Text The bulk of the work and literature in information retrieval has been about the use of only the document's text. See for example NIST's TREC[25] proceedings, or the excellent texts [18] and [20]. Accordingly, evaluation

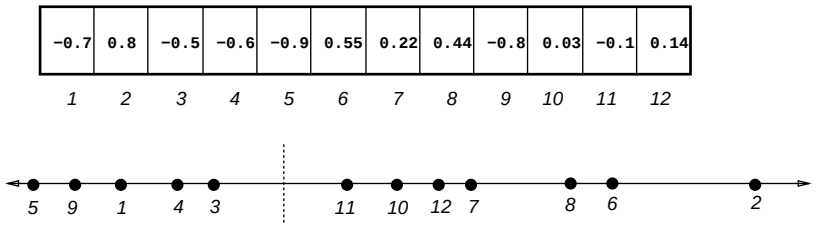

Figure 2: The partitioning process: an eigenvector (top) and the documents ordered on the line (bottom). A split is made between 3 and 11.

of retrieval systems has centered around text-only benchmarks that emphasize the notion of *relevance*, which is in general distinct from quality, making it necessary for us to resort to user studies on hyperlinked corpora for which no benchmark for quality is yet standard.

Bibliographic citations This field, known as "bibliometrics", has focused primarily on the scientific literature. Work in this field is focussed on exploiting structure characterized by the following mutually dual similarity measures between documents: "bibliographic coupling" (the number of common citations they contain [26]) and "cocitation" (the frequency with which both appear as citations in the same document [22].) See the reviews [28] and [14]. Such similarity measures can be coupled with textual similarity, as in the HyPursuit system [27].

Hyperlinks There are numerous methods in this category, including the HITS algorithm described above, PageRank [2], WebQuery [6], and the "Topic Distillation" work of Bharat and Henzinger [1]. PageRank ranks web pages by simulating a random walk on the web, which can be described by a Markov chain whose steady-state probabilities are then taken as the ranks given to the corresponding documents. The entire web (or a large fraction of it) is ranked this way and then the response (i.e., "hit list") to a Boolean query is sorted by this rank before it is presented to the user. Note that our ranking is not only a function of the web graph, but also of the specific query for which the ranking is computed. "Connectivity" (total link count, both in- and out-links) is used in a visualization scheme to rank pages in WebQuery. Finally, Bharat and Henzinger identify and address some problems associated with HITS and with [3], which introduces the notion of using the query terms to build the matrix A.

Hypertext Many methods for hyperlink- and text-based relevance ranking are known. Croft and Turtle [7] propose a scheme for incorporating hypertext links as well as bibliographic citations into an information retrieval system in which the relevance of a document to a query is computed by a Bayesian network. Savoy [21] describes a family of relevance-ranking schemes for doing query-based information retrieval in hypertext. The scheme uses both a term-based inverted index and links that can be either bibliographic citations, hyperlinks, or both. Certain aspects of this approach resemble ours. For example the acquisition and expansion of the root set are virtually identical. For the actual ranking, however, *spreading activation* is used. The starting activation values are computed based on linguistic similarity to the query. Marchiori [16] computes a relevance ranking function for hypertext pages that incorporates a measure of the relevance of the pages to which they point. This measure utilizes any ranking function based solely on the textual content of the pages themselves.

2.6.2 Structure discovery

Text The text-only version of structure discovery is simply clustering. See [29] for a review.

Bibliographic citations Some work has been done on the use of only bibliographic citations to discover structure in a corpus. For example Small[22] uses citations to discover clusters, which he refers to as "co-citation networks." A similar study of citation graphs is undertaken by Larson [23].

Hyperlinks As discussed, the method of HITS includes a purely link-based method of structure discovery: finding clusters using higher-order eigenvectors. Though deployed only for hypertext, it could also be used for pure citation-based clustering.

Hypertext Pirolli, Pitkow, and Rao [17] address the problem of identifying aggregates of pages that correspond to a conceptually unified entity. They use (among other things) link structure, usage paths (taken from server logs), text, and meta information about pages. Rivlin *et. al.* [4, 5, 19], address a similar issue in the context of providing the user with better navigational aids. They introduce the notion of index (high out-link count) and reference (high in-link count) nodes, similar to the HITS notion

of hubs and authorities. Weiss *et. al.* [27] describe *HyPursuit*, a similar structure-based hypertext navigational system.

3 Experiments on the WWW

We have implemented the algorithm of Section 2.3 in a system called Clever, which performs spectral filtering for the web. This section reports on an experiment comparing Clever to Altavista and Yahoo!. Section 3.1 highlights some difficulties that arise in applying traditional information recall metrics to the WWW. Section 3.2 describes our experiments, and Section 3.3 gives results. Section 3.4 includes some discussion.

3.1 Performance evaluation for the WWW

Traditionally, retrieval systems are evaluated using the measures of precision and recall on a large pre-evaluated corpus [20]. As there is no standard benchmark for the web that has been rated and classified, we cannot take this approach directly. But the situation is worse than this:

- The web currently contains around 300 million documents so rating the entire corpus is a daunting task. But we cannot label only a small subset because broad-topic queries routinely return a million page hits scattered around the web.
- Even if it were possible to create such a corpus, a million new pages arrive daily, so the corpus could be used to evaluate actual search engines for only a brief window (probably shorter than the time to gather the relevance judgements) before too many new pages arrived.
- The composition of a "typical" web document in terms of inlinks, outlinks, average size, graphical content, layout, function, etc., is dynamic. Even if we are not interested in comparisons that include actual search engines, and instead wish to label a snapshot of the web then compare algorithms on this snapshot, results for the web of a few years ago may not generalize to the web of today.
- Existing labeled corpora such as TREC[25] are primarily standalone text while we examine algorithms that rely fundamentally on the hyperlinked nature of the web. Even other hyperlinked corpora tend to contain documents of much higher quality than the web. So modifying existing labeled corpora for evaluating web-targeted algorithms is also difficult.
- The closest approximations to "relevance judgements" on today's web are sites such as Yahoo![31], which through human involvement collect high-quality pages on a number of topics. Unfortunately, due to the growth figures given above these sites cannot hope to index all pages on the web. If a search engine returns a page on a particular topic that is also indexed by Yahoo! under that topic, we have a strong indication that the page is high quality. If, however (as is more likely), Yahoo! does not index the page, we have no information about the quality of the page.

3.2 Our experiment

Because of these difficulties, we choose to evaluate our approach using relevance judgements (gathered through a user study) of a small subset of the web. We therefore compare ourselves against the best-known automatic search engine, Altavista[8], and the best-known human-compiled resource site, Yahoo![31]. We compute the precision of all three sources on a fixed number of pages according to our user-provided relevance judgements and compare these results. We refer to this technique as *comparitive precision*.

More precisely, for each of the 26 queries listed in Table 1 we extracted ten pages from each of our three sources. Altavista and Clever were both given the query as it appears in the table (i.e., with quotes, plus-signs, and capitalization intact). The same search was entered manually into Yahoo!'s search engine, and of the resulting leaf nodes, the one best matching the query was picked by hand. If the best match contained too few links, the process was repeated to generate additional links. Using this procedure we took the top ten pages from Altavista, the top five hubs and five authorities returned by spectral filtering, and a random ten pages from the most relevant node or nodes of Yahoo![4]. We then interleaved these three sets and sorted the resulting approximately 30 pages alphabetically (there are almost never duplicate pages from the three sources). We asked each user to rank each of these pages "bad," "fair," "good," or "fantastic" based on how useful the page would be in learning about the query. We took good and fantastic pages to be relevant, and then computed precision in the traditional manner. Since our users evaluated only the pages returned from our three sources, but did not know which source returned which page, we refer to this type of data as *blind post-facto* relevance judgements.

The subjective evaluation of relevance was performed by a set of 37 subjects, yielding 1369 datapoints. The subject was free to browse the list of pages at leisure, visiting each page as many times as desired, before deciding on a final quality score.

[4] Yahoo! lists pages alphabetically and performs no ranking, hence the requirement that we take ten pages at random.

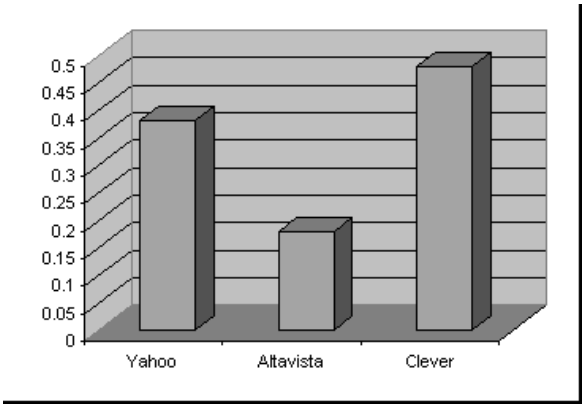

Figure 3: Average precision over all queries of the ten documents returned by each system

3.3 Results

Average precision. Figure 3 shows the average comparitive precision of each search engine over the set of 26 queries; Table 1 then shows these precision values for each topic.[5] Clever outperformed both Yahoo! and Altavista under this metric. While the favorable comparison to Altavista was expected, the advantage over Yahoo! was surprising. An analysis of the standard deviation across queries shows that in fact the results for Clever were slightly more tightly clustered than for Yahoo! (standard deviation of $.19$ versus $.21$), and that results for Altavista were consistently lower than for the other systems (standard deviation of $.10$). Similarly, Figure 4 considers the fraction of queries on which each search engine performed best. In 81% of all topics, Clever was either the best in terms of precision or tied for first place with Yahoo!

Page Rankings. The previous figures consider the average comparitive precision across all pages returned by a particular engine. We now consider the rank assigned to a page by each engine. Figure 5 plots the average precision of the top i pages for each engine, for $i = 1 \ldots 10$. For this purpose, the ranking function that we use for Clever interleaves the hubs and authorities starting with the best hub.

One possible concern is that a large Yahoo! node may contain many good pages and some excellent ones. Choosing only ten pages at random from such a node may penalize Yahoo! for gathering more information on the topic. However, almost all our Yahoo! nodes contained fewer than 30 pages, and the correlation of precision to Yahoo! node size is minimal, only -0.09. This indicates

[5]Note that, while the authors of [1] report precision values for a similar system, private communication from the authors suggests our precision figures are not directly comparable with theirs.

Query	Yahoo!	AltaVista	Clever
+Thailand +tourism	0.3	0.0	0.2
+recycling +cans	0.1	0.1	0.4
“Gulf war”	0.5	0.1	0.3
“affirmative action”	0.6	0.2	0.6
“amusement park”	0.0	0.1	0.4
“classical guitar”	0.3	0.2	0.5
“computer vision”	0.7	0.4	0.8
“field hockey”	0.1	0.1	0.2
“graphic design”	0.2	0.1	0.2
“lyme disease”	0.6	0.1	0.6
“mutual funds”	0.7	0.4	0.5
“parallel architecture”	0.2	0.2	0.3
“rock climbing”	0.6	0.1	0.8
“stamp collecting”	0.2	0.3	0.5
“table tennis”	0.6	0.1	0.6
“vintage car”	0.1	0.2	0.2
HIV	0.4	0.3	0.8
alcoholism	0.2	0.2	0.4
bicycling	0.4	0.0	0.2
blues	0.5	0.1	0.6
cheese	0.6	0.2	0.6
cruises	0.5	0.4	0.5
gardening	0.5	0.1	0.4
shakespeare	0.6	0.1	0.6
sushi	0.4	0.2	0.7
telecommuting	0.4	0.2	0.8

Table 1: Comparitive precision by topic.

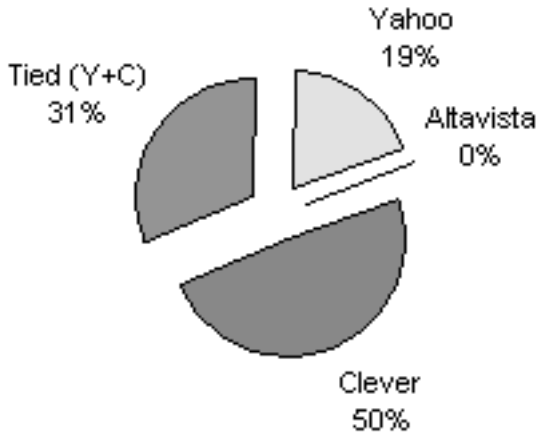

Figure 4: Percentage of topics for which each system had the highest number of high-quality relevant documents.

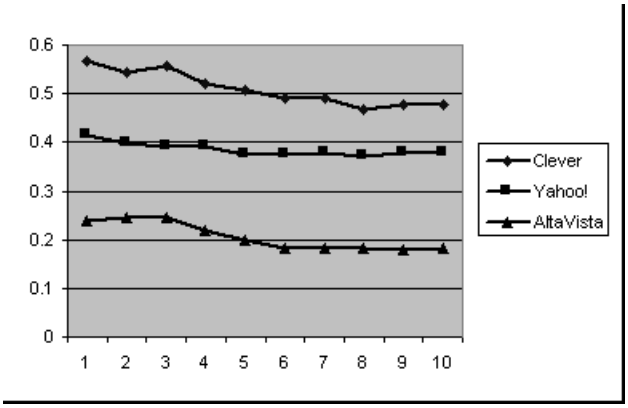

Figure 5: Precision as a function of the rank of pages.

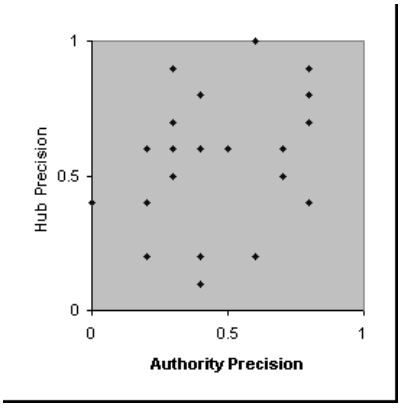

Figure 6: Scatter of hub and authority scores.

that the concern is not serious.

Hubs and Authorities. The precision scores of hubs and authorities show only a mild correlation (.36). Figure 6 shows the scatter plot by topic; in some cases hubs dominate, and in others authorities dominate, suggesting that users find value in both types of pages. Overall, Clever is better at identifying hubs than authorities — in 72% of the queries, the comparitive precision of the hubs was at least as high as the authorities.

Other overall comparative measures. We also give two alternate measures of overall quality. The first measure, "fantastic fraction," is the fraction of pages returned that are rated as "fantastic" (rather than either "good" or "fantastic" in our original measure). The second, "linear measure," weights a "bad" page at 0, a "fair" page at .33, a "good" page at .66 and a "fantastic" page at 1. The results for these measures are given in Table 2. Clever performs better than all other systems under all measures, although Yahoo! finds roughly as many "fantastic" pages.

Measure	Yahoo!	Altavista	Clever
Precision	.38	.18	.48
Fantastic Fraction	.13	.04	.15
Linear Measure	.42	.27	.50

Table 2: Alternate Overall Quality Ratings, by Search Engine.

3.4 Discussion

It remains unclear how to judge the response set of a search engine as a whole, rather than page-by-page. Both the covering and the packing heuristics may reject pages that are individually highly rated in favor of pages that contribute to the overall quality of the response set. Hence we believe that the quality of our result set as a collection of pages will be better than the average precision metric indicates.

4 Conclusions

We have shown that spectral filtering is an effective technique for analyzing text and hypertext for searching, partitioning, and estimating a notion of document quality. This technique both is general and flexible: by appropriately modifying the notions of entities and links, one can extract interesting structure from a wide variety of hyperlinked and non-hyperlinked corpora. We have a prototype implementation that is simple and fast; the only performance bottleneck is the inherent delay of scanning and indexing a large corpus.

Acknowledgment

We are grateful to Jon Kleinberg for numerous discussions on the algorithms and experiments in the Clever system. We thank our test subjects for performing the evaluations described in Section 3.2, and Yael Ravin for her assistance in acquiring some experimental data.

References

[1] K. Bharat and M.R. Henzinger. Improved Algorithms for Topic Distillation in a Hyperlinked Environment. To appear, *Proceedings of ACM SIGIR*, 1998.

[2] S. Brin and L. Page, The Anatomy of a Large-Scale Hypertextual Web Search Engine. To appear in the *Proceedings of the 7th World-wide web conference (WWW7)*, 1998.

[3] S. Chakrabarti, B. Dom, D. Gibson, J. Kleinberg, P. Raghavan and S. Rajagopalan. "Automatic Resource Compilation by Analyzing Hyperlink Structure and Associated Text", To appear in the *Proceedings of the 7th World-wide web conference (WWW7)*, 1998.

[4] Rodrigo A. Botafogo and Ben Shneiderman, "Identifying aggregates in hypertext structures", *Proceedings of ACM Hypertext '91*, pp. 63-74, 1991

[5] R. Botafogo, E. Rivlin, B. Shneiderman, "Structural analysis of hypertext: Identifying hierarchies and useful metrics," *ACM Trans. Inf. Sys.*, 10(1992), pp. 142–180.

[6] J. Carrière, R. Kazman, "WebQuery: Searching and visualizing the Web through connectivity," *Proc. 6th International World Wide Web Conference*, 1997.

[7] W. Bruce Croft and Howard Turtle, "A retrieval model for incorporating hypertext links", *Proceedings of ACM Hypertext '89*, pp. 213-224, 1989

[8] Digital Equipment Corporation, *AltaVista search engine*, `altavista.digital.com/`.

[9] W.E. Donath, A.J. Hoffman, "Algorithms for partitioning of graphs and computer logic based on eigenvectors of connections matrices," *IBM Technical Disclosure Bulletin*, 15(1972).

[10] S. Deerwester, S. Dumais, T. Landauer, G. Furnas, R. Harshman, "Indexing by latent semantic analysis," *J. American Soc. Info. Sci.*, 41(1990).

[11] D. Goldberg, D. Nichols, B.M. Oki, and D. Terry. Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35:12, pp. 51–60, 1992.

[12] G. Golub, C.F. Van Loan, *Matrix Computations*, Johns Hopkins University Press, 1989.

[13] J. Kleinberg, "Authoritative sources in a hyperlinked environment," *Proc. ACM-SIAM Symposium on Discrete Algorithms*, 1998. Also appears as IBM Research Report RJ 10076(91892) May 1997, and as `www.cs.cornell.edu/home/kleinber/auth.ps`.

[14] Mengxiong Liu, "Progress in documentation the complexities of citation practice: a review of citation studies", *J. Documentation*, **49**(4), pp.370-408, 1993

[15] L. Lovász. On the ratio of the optimal integral and fractional covers. Discrete Mathematics 13, 383–390, 1975

[16] Massimo Marchiori, "The Quest for Correct Information on the Web: Hyper Search Engines", *The 6th International World Wide Web Conference (WWW6)*, 1997. Also available at `atlanta.cs.nchu.edu.tw/www/PAPER222.html`.

[17] P. Pirolli, J. Pitkow, R. Rao, "Silk from a sow's ear: Extracting usable structures from the Web," *Proc. ACM SIGCHI Conference on Human Factors in Computing*, 1996.

[18] C.J. van Rijsbergen, *Information Retrieval*, Butterworths, 1979. Also at `dcs.glasgow.ac.uk/Keith/Preface.html`.

[19] E. Rivlin, R. Botafogo, B. Shneiderman, "Navigating in hyperspace: designing a structure-based toolbox," *Communications of the ACM*, 37(2), 1994, pp. 87–96.

[20] G. Salton. *Automatic Text Processing.* Addison-Wesley, Reading, MA, 1989.

[21] Jaques Savoy, "Ranking schemes in hybrid boolean systems: a new approach", *J. Am. Soc. Information Sci.*, **48**(3), pp.235-253, 1997

[22] H. Small, "Co-citation in the scientific literature: A new measure of the relationship between two documents," *J. American Soc. Info. Sci.*, 24(1973), pp. 265–269.

[23] Ray R. Larson, "Bibliometrics of the World Wide Web: An Exploratory Analysis of the Intellectual Structure of Cyberspace." *Proceedings of the 1996 Annual ASIS Meeting*, Baltimore.

[24] D. Shenk. Data Smog. New York: Harper and Collins, 1997.

[25] TREC - Text REtrieval Conference, co-sponsored by the National Institute of Standards & Technology (NIST) and the Information Technology Office of the Defense Advanced Research Projects Agency (DARPA) as part of the TIPSTER Text Program. `trec.nist.gov/`

[26] Bella Hass Weinberg, "Bibliographic Coupling: A Review", *Information Storage and Retrieval*, Vol.10, pp. 189-196, 1974

[27] R. Weiss, B. Velez, M. Sheldon, C. Nemprempre, P. Szilagyi, D.K. Gifford, "HyPursuit: A Hierarchical Network Search Engine that Exploits Content-Link Hypertext Clustering," *Proceedings of the Seventh ACM Conference on Hypertext*, 1996.

[28] H.D. White, K.W. McCain, "Bibliometrics," in *Ann. Rev. Info. Sci. and Technology*, Elsevier, 1989, pp. 119-186.

[29] Peter Willet, "Recent trends in hierarchical document clustering: a critical review", *Information Processing and Management*, Vol.24, No.5, pp. 577-597, 1988

[30] R.S. Wurman. Information Anxiety. New York: Doubleday, 1989.

[31] Yahoo! Corp. *Yahoo!*, `www.yahoo.com`.