The Shape of the Web and Its Implications for Searching the Web

Kemal Efe, Vijay Raghavan, C. Henry Chu, Adrienne L. Broadwater, Levent Bolelli, Seyda Ertekin

Abstract— **With the rapid growth of the number of web pages, designing a search engine that can retrieve high quality information in response to a user query is a challenging task. Automated search engines that rely on keyword matching usually return too many low quality matches and they take a long time to run. It is argued in the literature that link-following search methods can substantially increase the search quality, provided that these methods use an accurate assumption about useful patterns in the hyperlink topology of the web. Recent work in the field has focused on detecting identifiable patterns in the web graph and exploiting this information to improve the performance of search algorithms. We survey relevant work in this area and comment on the implications of these patterns for other areas such as advertisement and marketing.**

Keywords— **Search engines, link analysis, information exploration, related pages, World Wide Web.**

I. Introduction

Use of the link structure has recently emerged as a promising approach for searching the web. Link-based approaches have been inspired by an analogy with citation of related works in scientific literature. A citation provides a link between two articles, and often is the only way for readers to learn about other articles related to the topic of a given article. A link on a web page serves a similar purpose as it leads the way from one page to another, but there are important differences between a scientific citation and a web link:

- Human judgement applied to a web citation is generally more subjective and noisy than in scientific literature. Most link creators may not even have a claim of relevance, objectivity, or information quality.
- While some links on a web page may lead to related (or unrelated) pages, others may be there merely for navigational purposes (e.g. "click here to return to the home page").
- A citation in the scientific literature is a static and unidirectional pointer; once an article is published, there is no way to add new references to it. For this reason, it is exceptionally rare for two articles to cite one another. In contrast, web pages may (and often do) link to other documents created afterwards. The fact that the average distance between two web pages is relatively small (19 clicks [3], [2]) is a direct consequence of this freedom to add links to existing pages.

The first two points above weaken the assertion that links on web pages could serve a useful purpose in an automated method for searching the web, but the last point is a significant help. New links added to existing pages makes it easier to form certain patterns in the web graph that would be harder to find in a citation graph of scientific literature, and even harder to find in a random graph. An intuitive implication of the *19 clicks theory* is that the web graph must contain densely connected regions that are in turn a few clicks away to one-another. These densely connected regions must form certain recognizable patterns as a *signature* of collective intelligence even though different pages may have been created and maintained independently from one another. Indeed, research that we review here has shown that although an individual link is weak evidence of relevance, an aggregate of links forming a special pattern is a robust indicator of relevance. When the link information is augmented with text-based information on the page and/or around the anchor text, even better search results have been obtained. In this paper we review a number of such techniques applied to information retrieval on the web, and identify possible research directions.

II. Basic Graph Patterns

The most basic element of a graph is a directed link. A link on a web page connects one document to another, and represents an implicit endorsement of the target page.

When we consider two links, we obtain a number of possible basic patterns as shown in Figure 1. Two pages pointing to each other reinforce our intuition about their mutual relevance. Co-citation occurs when a page points to two distinct pages. In bibliometric studies [30], it is asserted that relevant papers are often cited together, and here we assume that a similar assertion holds. For example, a page that cites the home page of the New York Times is very likely to cite the home page of the Washington Post also. Social choice (or social filtering) is the situation where two documents link to a third page. From this pattern, we infer that the two pages are related to each other since they both link to the same document. Finally, transitive endorsement occurs when page p_1 links to p_2, which in turn links to p_3. Transitively, p_1 is considered to endorse p_3, though this is a weaker form of endorsement.

These basic structures can blend together to form more complex patterns that further strengthen the relationships among a set of web pages. See Figure 2 for some examples. One of these is the complete bipartite graph. In [23], Kumar et al. used a special form of a directed complete bipartite graph as the signature of an *emerging* web community[1]. In this graph, the nodes are divided into two

K. Efe is with Bilkent University and the University of Louisiana–Lafayette.

V. Raghavan, C.H. Chu, A. L. Broadwater, L. Bolelli and S. Ertekin are with the University of Louisiana–Lafayette

[1] A web community is a set of page creators with similar interests.

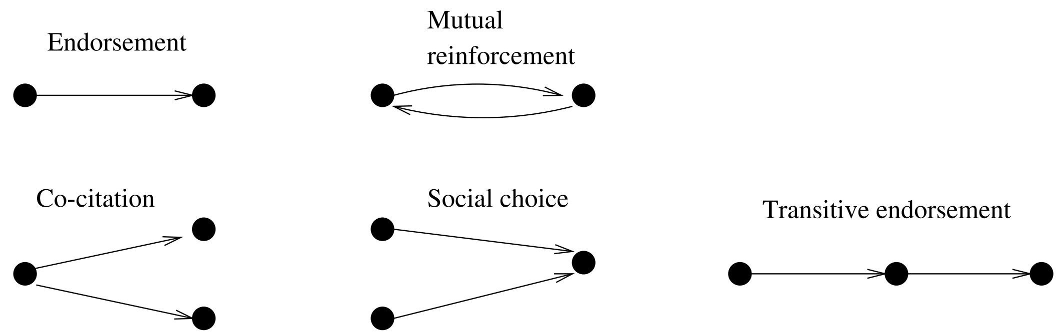


Fig. 1. Basic patterns formed by two directed edges.

subsets F and A such that each node in F links to every node in A. The set of nodes in F are called the fans, and the set of nodes in A are called the authorities. Another useful structure is the clan graph. An NK-clan is a set of K nodes in which there is a path of length N or less (ignoring the edge directions) between every pair of nodes. This structure has been used for detecting and visualising inter-site clan graphs in [28].

A generalization of *social choice* is an in-tree. Conversely, a generalization of *co-citation* yields an out-tree. Of particular interest are the trees with large in/out degrees at the root. The interest in in-trees is due to an assertion that if many different pages link (directly or transitively) to a document, it is likely that the heavily linked page is an authoritative source on some topic of interest shared by other pages in its graph neighborhood. This is analogous to measuring the impact of scientific papers by the number of citations they receive. The interest in out-trees is due to an analogy with survey papers. If a web page links to many authoritative pages on some topic, then we consider it to be a good source for searching relevant information.

III. Structural Analysis

As we noted above, NK-clan graphs and directed complete bipartite graphs have been used as the basic patterns to be searched for in the web graph. In a related work, tree structures have been used as a guideline to design better hyperlinked structures [7]. The reverse process of extracting tree structures to discover and visualize topical hierarchies in hyperlinked text has also been studied [7], [24], [25]. In case of a topic search on the web, we don't need to extract tree structures from the web graph. Often, the user is only interested in finding a small number of authoritative pages on the search topic. These are the pages that would play a prominent role in a tree (such as the root), had we extracted the tree itself. An alternative to extracting trees in a web search is to apply a ranking method to the nodes of the web graph that has an analogous outcome in detecting prominent nodes. In this section, we review such methods proposed in the literature. To provide a unified view of the different models in the literature, we first develop a few basic concepts.

A. Basic Concepts

We first consider a directed graph G and its adjacency matrix X as shown in Figure 3. An entry $x_{p,q} = 1$ if and only if there is an edge from p to q. Otherwise $x_{p,q} = 0$. Now consider two linear transformations defined on unit vectors a and h as follows:

$$a = X^T h \quad (1)$$

$$h = Xa \quad (2)$$

This is equivalent to

$$a = X^T X a \quad (3)$$

$$h = X X^T h \quad (4)$$

It is interesting to examine these matrix products. First of all, both product matrices are diagonally symmetric. This property is of no immediate interest to us, except that it is useful if one is interested in analyzing the convergence properties of related search algorithms. Of immediate interest to us are the following observations:

- An entry (p,q) in the product XX^T is equal to the number of other pages to which both pages p and q point. This value could be used as a measure of how much p and q have in common. Two pages that have a large overlap in their citations are likely to be very similar to each other. For pages with small outdegrees, a relatively large overlap plays an important role in the formation of the directed complete bipartite graphs which happen to be robust indicators of web communities[2].
- An entry (p,q) in the product $X^T X$ represents the number of other pages that link to both p and q. This information can be used as a measure of how many other's consider these two pages as being related. This measure is called the *degree of co-citation* between p and q in [17], and used for detecting related pages in the web graph.
- A diagonal entry (p,p) in XX^T represents the out-degree of the node p in G.

[2] For pages with large out-degrees too much overlap in their links often turned out to be a sign of plagiarism between web pages. Kumar et al. [23] found that several pages of Yahoo! were plagiarised more than 50 times each. While plagiarised pages are strongly similar as predicted from the overlap of their outgoing links, several researchers preferred to delete such duplicates from the web graph before applying their algorithms [17], [23]

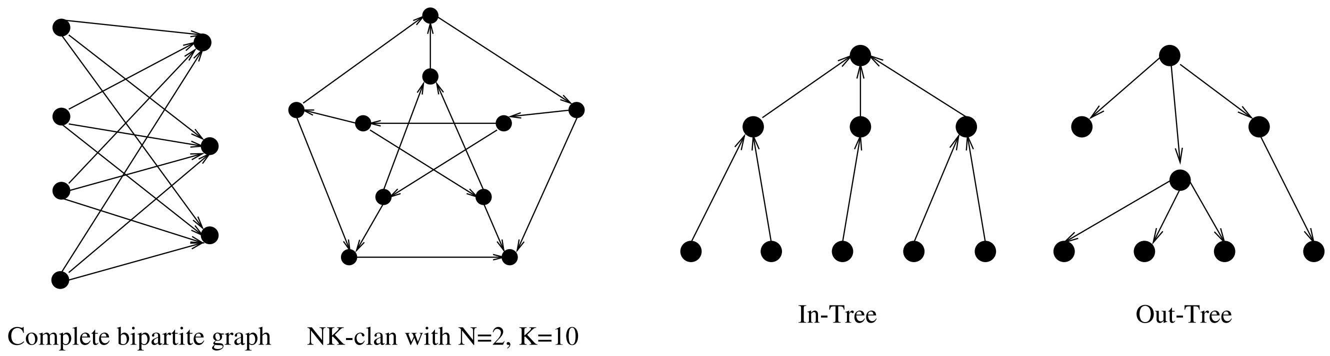
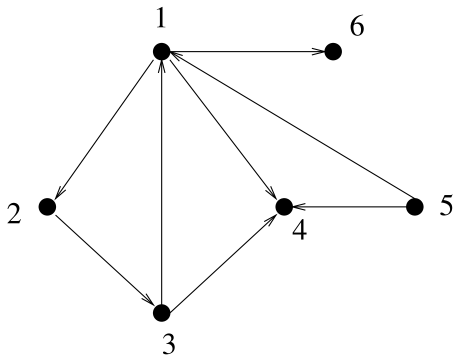


Fig. 2. Complex patterns that are indicative of web communities.



	1	2	3	4	5	6
1		1		1		1
2			1			
3	1			1		
4						
5	1			1		
6						

$$\underset{X}{\begin{bmatrix} 0&1&0&1&0&1\\ 0&0&1&0&0&0\\ 1&0&0&1&0&0\\ 0&0&0&0&0&0\\ 1&0&0&1&0&0\\ 0&0&0&0&0&0 \end{bmatrix}} \quad \underset{X^T}{\begin{bmatrix} 0&0&1&0&1&0\\ 1&0&0&0&0&0\\ 0&1&0&0&0&0\\ 1&0&1&0&1&0\\ 0&0&0&0&0&0\\ 1&0&0&0&0&0 \end{bmatrix}} = \underset{X * X^T}{\begin{bmatrix} 3&0&1&0&1&0\\ 0&1&0&0&0&0\\ 1&0&2&0&2&0\\ 0&0&0&0&0&0\\ 1&0&2&0&2&0\\ 0&0&0&0&0&0 \end{bmatrix}}$$

$$\underset{X^T}{\begin{bmatrix} 0&0&1&0&1&0\\ 1&0&0&0&0&0\\ 0&1&0&0&0&0\\ 1&0&1&0&1&0\\ 0&0&0&0&0&0\\ 1&0&0&0&0&0 \end{bmatrix}} \quad \underset{X}{\begin{bmatrix} 0&1&0&1&0&1\\ 0&0&1&0&0&0\\ 1&0&0&1&0&0\\ 0&0&0&0&0&0\\ 1&0&0&1&0&0\\ 0&0&0&0&0&0 \end{bmatrix}} = \underset{X^T * X}{\begin{bmatrix} 2&0&0&2&0&0\\ 0&1&0&1&0&1\\ 0&0&1&0&0&0\\ 2&1&0&3&0&1\\ 0&0&0&0&0&0\\ 0&1&0&1&0&1 \end{bmatrix}}$$

Fig. 3. Adjacency matrix and its useful properties

• A diagonal entry (p,p) in X^TX represents the in-degree of the node p.

Pages with large in/out degrees often play a central role in the web graph. The two algorithms we present next capitalise on this notion.

B. A Basic Method for Page Ranking

An example technique that is reminiscent of finding the roots of in-trees is the ranking method developed in Google [8]. Page ranking is done by using an algorithm that is called (none other than) PageRank. Google's web crawlers continuously search the web to collect new pages and update the old ones. These pages are stored in a data repository. The link structure of these pages are stored separately from other information and represented as the web graph. This graph is then used for computing page ranks. The rank of a page determines its location in the output list, if it is selected in response to a user query. Let r_p be the rank of a page p and x_p be the number of outgoing links on a page. Recursively, the rank of a page p is computed as:

$$r_p = (1-d) + d \sum_{\forall q; q \rightarrow p} r_q / x_q$$

where d is a damping factor selected between 0 and 1. As can be seen, the rank of a page depends on the number of pages and the individual ranks of pages pointing to it. This equation can be seen as modeling the behavior of a "random surfer" (term coined by Brin and Page) who keeps clicking on the links, but gets bored eventually and starts from another random page. The summation term in the above equation is just the probability that a page is selected from one of the neighboring pages that link to it. As the readers will notice, the equation awards higher ranks to pages with high in-degrees, or pages that are linked to by highly ranked pages.

As a different way to view this computation, consider the adjacency matrix X of the web graph. In this graph, rows represent the outgoing links such that the entry $a(p,q) = 0$ if there is no link from page p to page q. Otherwise, $a(p,q) = 1/x_p$, where x_p is the number of outgoing links on page p. The summation term in the above computation is just the matrix-vector multiplication $X^T r$ where X^T is the transpose of X, and r is the rank vector. In this computation r can be initialized to the unit vector, and the computation can be repeated until certain nodes distinguish themselves by a relatively higher rank than the others. This should normally happen after a few tens of iterations since the computation converges to the principal eigenvector of the matrix X^T [9].

C. Extracting Hubs and Authorities

Kleinberg developed an experimental search technique [21], called the HITS (Hyperlink-Induced Topic Search), that is particularly effective for finding the pages with a similarly central role in the web graph. This algorithm finds both *authorities* and *hubs*. Authorities are those pages prominent in their neighborhood of the web graph due to many other pages pointing toward them. Hubs are prominent in their neighborhood for pointing toward many good authorities. Authorities and hubs in the web graph have a mutually reinforcing relationships; good authoritative pages on a search topic are likely to be found near good hubs that in turn link to many good sources of information.

The HITS algorithm has two major steps: sampling and weight-propagation. The sampling step uses a keyword-based search to select around 200 pages by using one of the commercially available search engines. This set of pages is called the *root set*. This root-set is then expanded into a *base set* by adding any page on the web that has a link to/from a page in the root set. The base set typically contains a few thousand pages. The pages in the base set may or may not constitute a connected graph, but at least it has a large connected component [22].

The purpose of the weight-propagation step is to compute a weight for each page in the base set that can be used to rank their relevance to the query. Two forms of relevance are considered: *authority* and *hub*. This is a recursive process, where each page p is assigned an authority weight a_p and a hub weight h_p, which are equal for all pages initially. Recursively, the algorithm updates these values as follows:

$$a_p = \sum_{\forall q; q \rightarrow p} h_q$$

$$h_p = \sum_{\forall p; p \rightarrow q} a_p$$

where $q \rightarrow p$ means that q has a link to p. Hence we see that the authority weight of a page will be higher if it is pointed to by many pages, or pointed to by pages that have higher hub weights. Conversely, the hub weight of a page will be higher if it points to many pages, or points to pages with higher authority weights.

This computation is very similar to the matrix computations in equations 1 and 2, and carries all the properties we outlined in Section III.A. The important difference, however, is a normalization applied to the weight vectors between iterations. Before each iteration, the weights are normalized so that their squares sum to 1. The matrix entries are binary values, rather than fractional values used in the PageRank algorithm. When recursive updates are applied, the weight vectors a and h converge to the principal eigenvectors of X^TX and XX^T, respectively[21]. In practice, the iterative computation is repeated for only a small number of steps. The output of the algorithm is a short list of pages with the largest hub weights and a separate list of pages with the largest authority weights. The implementation typically outputs 10 from each group as the final list.

IV. Implementations of Structural Analysis Techniques

The page ranking techniques reviewed so far have been used in a number of research projects, but almost all implementations had to modify the basic ideas discussed above.

Some of these modifications tried to counter certain peculiarities of the algorithms that became apparent once implemented. Others try to counter difficulties that arise due to the large amount of noise in the web structure.

A. Links-Only Techniques and Related Difficulties

The original purpose of the HITS algorithm was to rank the pages found by a text-based search engine. It was meant for broad search topics with some amount of presence on the web. Bharat and Henzinger [6] reported an implementation of the HITS algorithm for the purpose of *topic distillation*. Given a broad topic, topic distillation is the process of extracting a small number of high-quality pages most representative of the topic. While the HITS algorithm worked well for some cases, it performed poorly in general. The authors implemented a visualization tool [5] that helped discover three problems with the links-only approach:

- A mutually reinforcing relationship occurs between hosts when several pages on one host point to a single page on another host. This situation inflates the authority weight of the single document, which in turn drives up the hub weights of other documents pointing to it. This typically happens when designers of individual pages copy the page template from a master copy (e.g. one that is designed by the site programmer), and the new pages inherit the link from the master copy.
- The reverse problem occurs if a single document on a host points to several documents on another host. The large number of outgoing links gives the source document an unduly large hub weight, which in turn magnifies the authority weight of every document it points to.
- The problem of *topic drift* may occur if even one of the documents in the *root set* is non-relevant to the search topic. This problem may not be very pronounced if the non-relevant document is sparsely connected. But if that document has many incoming links from outside the root set, then all of those pages linking to it will be included in the extended *base set*. Consequently, it may be output a high authority page on the search topic even though it may have no relevance to the search topic.

The net effect of these anomalies is that some pages are awarded higher ranks than warranted by their relevance to the search topic. The first two problems are effectively mitigated by modifying the weights in the adjacency matrix so that fractional weights may be used instead of binary. To address the first item above, Bharat and Henzinger modified the edge weights in X^T so that whenever k documents at one site point to a single document on another site, each of these links get an authority weight of $1/k$. The second problem is similarly solved: if a single document on one site links to l documents on another site, the corresponding links in X get a hub weight of $1/l$. The last item above is addressed by using textual information which we will discuss in Section IV.B.

Similar modifications were also used in Chakrabarti et al. [14]. In addition to the above anomalies, Chakrabarti et al. observed:

- When the topics of discussion vary on different parts of the same page, the outgoing links also point to different topics depending on their location on the page. If the page has a large out-degree, it will be awarded a large hub weight. It will in turn award high authority weights to each page it links to on the subject of the user query, whereas only one or two of those linked pages may be related to the user query.
- *Topic generalization* occurs if the search topic is not sufficiently broad. On narrowly focused topics, HITS frequently returns good sources for a more general topic. An example given was the Nebraska tourist information page being returned in response to a query for skiing in Nebraska. Gibson et al. observed that [19] topic generalization in the behavior of the HITS algorithm does not always result in a drift from more specific pages toward more general pages; the reverse can happen too. For example, when searching for authoritative pages on "linguistics," the returned list of pages was dominated by pages in the field of "computational linguistics." While this is a sub-topic of the initial query, HITS has converged to it because of the considerably greater density of linkage in its neighborhood of the web graph.

To solve the first problem, Chakrabarti et al. [14] used a page splitting heuristic. The basic intuition here is that in a large hub with several outgoing links, the links close together are more likely to focus on a common topic than links that are far apart. The second problem is addressed by a text-based method as discussed in Section IV.B.

So far we have seen examples where links-only algorithms had reasonably good performance, but they eventually run into problems that have no apparent solutions without considering textual information. The work in [19] and [23] showed that links-only approaches can be very effective when searching for web communities. A web community is a set of content creators sharing a common interest. Newsgroups and commercial web directories are examples of web communities. At a minimum, the pages in a community must fall into the same taxonomy in a hierarchical categorization of topics. Automated methods for discovering web communities can be used when, for example, populating a commercial web directory. According to Kumar et al., there were about 20,000 large communities with well established existences on the web, and which are explicitly defined in directories such as Yahoo! and Infoseek. However, as argued in [10], considering the rapid growth of the web, manual methods used in these commercial efforts are too slow to have any hope of catching up. Automated methods for finding web communities can help expedite the work of human experts in discovering new candidates for inclusion in the existing taxonomies or for starting new taxonomies. As argued in [23], the ability to detect web communities also represents an opportunity for identifying and distinguishing communities for target advertising at a very precise level.

The work of Gibson et al. [19] focused on communities that are discovered by the HITS algorithm. After the first iteration, the top authorities in the base set are simply the pages with the largest number of incoming links. However, these pages may not have any thematic relationship among themselves. As the iterations are continued, different communities within the same base set crystalize in the form of tightly-knit patterns, each containing their own hubs and authorities. The reinforcing nature of hubs and authorities found in these communities bear relevance to *index* and *reference* nodes that play similar roles in hypermedia [7]. The reinforcing nature of hubs and authorities also underscores the reliance of the HITS algorithm on the *collective intelligence* of independent page designers. An interesting observation made was that the iterative computation can be forced to converge to different eigenvectors other than the principal eigenvectors. In this way, one could extract different communities from the same base set.

Kumar et al. focused on discovering *emerging* communities. There is an estimated number of more than 100,000 emerging communities on the web. While few of these emerging comunities eventually grow large enough to be included in major directories, most communities focus on a level of detail that is too finely grained to attract the interest of large portals. Example web communities discovered by their proposed algorithm underscores this point: the community of Turkish student organizations in the US, the community centered around oil spills off the coast of Japan, or the community of people interested in the Japanese pop singer, *Hekiru Shiina.* Such emerging communities often contain specific, up-to-date, and reliable information not found elsewhere on the web. The authors assert that even though emerging communities may not have a large presence on the web, they should be detectable by their *community signature.*

Thus, what is the signature of an emerging comunity? In the scientific literature, it is considered to be good practice to cite related work, but this tradition doesn't carry to web links often enough. For example, DELL and Gateway both have web sites that sell computers, but there is no link from one to the other. Besides conflict of interest, often sites closely related to each other do not link to each other, because they may not be aware of one another's existence, or they may cater to conflicting points of view on a topic. On the other hand, if a page has multiple outgoing links, those linked pages are likely to be related to each other. For example, a site that links to DELL is very likely to link to Gateway also.

This reasoning has led Kumar et al. to conclude that a community of web pages on a common topic must contain a densely connected directed bipartite subgraph. A graph is bipartite if its nodes can be partitioned into two subsets F and A, such that every edge whose source is in F has its destination in A. If such a graph is densely connected (which is what we expect in a web community), then a well known fact in graph theory states that, with very high probablity it has a core (a subgraph) that is a *complete* bipartite graph. The authors report that their experiments on the web generated over 100,000 $C_{3,3}$ graphs (directed complete bipartite graph with $|F| = |A| = 3$), and visual inspection of a randomly selected sample of about 400 of these showed less than 5% to be coincidental. This is a substantial level of accuracy achieved by a links-only approach.

Another algorithm that works well with links-only information is the Co-citation algorithm in [17]. Here the algorithm starts with a sample URL (instead of a keyword) and finds pages that are related to it. This is similar to the "What's Related" facility in Netscape. The method used in [17] is based on finding the pages that link to the sample URL and then determining "who else" they link to besides the sample URL. The algorithm outputs 10 of the pages that are most frequently co-cited with the sample URL.

The output of this simple-minded approach had much better precision than that of Netscape in experiments conducted. It also generally outperformed another links-only approach derived from the HITS algorithm that the authors implemented for comparison with the Co-citation algorithm. In this implementation, the base set required by the HITS algorithm is obtained from the sample URL by including its parents (the pages that link to it), its children (the pages that it links to), children of its parents, and parents of its children. The corresponding adjacency matrix is modified as in the method of Bharat and Henzinger we reviewed above [4]. At the end of the iterative computations, the algorithm outputs 10 of the highest ranked authority pages.

We think that a possible reason for the worse performance of the HITS algorithm (although still better than that of Netscape) may be attributed to the method of choosing the base set. Recall that a fundamental notion behind the HITS algorithm is the reinforcing nature of hubs and authorities. In HITS algorithm, hubs play an important role as conferrers of authority which help crystalize the role of authorities through iterative convergence. In the absence of conferrers of authority, it would be harder to find pages that have the authority. In a graph, one would expect that hubs would generally point toward authorities, but there is no reason for all the good hubs to be adjacent to the sample URL. Different hubs are more likely to be found among the "siblings" of parents and even grandparents of the sample URL. Different authorities are more likely to be found among the siblings of the sample URL. Excluding the grandparents of the initial URL may possibly leave a number of potentially good hubs (that are not necessarily adjacent to the sample URL) out of the base set. This may, in turn, affect the creation of good authorities.

B. *Adding Text-Based Heuristics*

The link-following methods reviewed above need a starting page or a set of pages from which they can explore the web. In a "What is Related" search, the starting page is a sample URL provided by the user. In a topic search, keyword-based techniques from the field of Information Retrieval are used to construct the initial set of pages. In Google, these pages are ordered according to the pre-computed ranks. In HITS, weights are computed on-the-fly from the neighborhood graph formed by the set of pages selected by text-based search methods (such as those derived from information retrieval [20]).

Given a search topic, finding relevant information on the web is a difficult problem. The existing search engines try to index and classify the pages on the web based on their content and associated metadata. Automating the classification of web pages with the help of link onformation has been studied in [11], [12], [13], [16], [25], [27], [29]. Recent work on the application of database techniques for modeling and querying the web, for information extraction and integration, and for web site construction has been surveyed in [18]. Gudivada et al. [20] give a detailed review of automated indexing methods and their use in document retrieval in searching the web.

Here we are mainly interested in different techniques that are effective in solving the problems encountered when using the link-following algorithms. First, we define a similarity measure between two documents, which is a key concept in information retrieval. Different measures of similarity have been defined (see for example [26], page 318), and they are all based on computing the inner product of term-frequency vectors x, y derived from two documents. Similarity measures essentially differ in the way they normalize the inner-product computation. A popular method is the Cosine normalization given by

$$S = \frac{\sum_{i=1}^{t} x_i \times y_i}{(\sum_{i=1}^{t} x_i^2 \times \sum_{i=1}^{t} y_i^2)^{1/2}}$$

where t is the length of the vectors x and y.

When discussing application of the HITS algorithm, we mentioned that two cases required text-based heuristics. These were the problems of topic drift and topic generalization. In both cases, the HITS algorithm drifts toward more heavily linked regions in the graph, and some auto-control mechanism is needed to prevent this drip. A simple idea used in the CLEVER project [13], [15] is based on the observation that text around the anchor of a link generally gives a good idea about the page being pointed to (e.g. "click here to post a message on our message board"). By comparing the search terms against the text around the link, a relevance weight is computed for each link. The weight $w(p, q)$ is just the number of matches found on page p around the link q. This yields a modified adjacency matrix where the entries are computed as $x(p, q) = 1 + w(p, q)$. This method can solve the topic generalization problem if the links pointing to the broader topic page have small weights. Small link weights should work as filters that block transfer of authority weights from highly relevant pages toward broad topic pages. The same net effect should ensue for non-relevant pages that may happen to be in the root set. This would indirectly solve the topic drift problem also. The authors report that the results of the CLEVER algorithm improved substantially over the results of the HITS algorithm.

Another approach presented in [4] focused on controlling the influence of pages rather than the individual links in them. For each page, a weight is computed based on its similarity with the search topic as measured by the cosine-normalized similarity measure above. Since users only type a few key words, it is difficult to compute a meaningful similarity measure between the key words and lengthy documents. On the other hand, the broad topic is better represented by the set of pages in the root set. Thus, the authors constructed a query document by combining together the first 1000 words from each document in the root set. Then they computed the similarity of this reference page with all the pages in the base set. This computation yielded the relevance weights of different documents. These values are used to dampen the hub weights and authority weights of pages before each iteration is started; authority weight a_p of page p is computed as $a_p = a_p \times r_p$, where r_p is the relevance weight of page p. Hub weights are computed similarly.

Intuitively, this modification solves the topic drift problems associated with having non-relevant pages in the base set. Pages with low relevance weights should converge to near-zero hub and authority weights quickly. However, it should also be effective in solving the topic generalization problem, if the broader topic page has a low relevance weight. It is a simpler algorithm to implement than the CLEVER algorithm. Since it does not directly address the problem at the link level, it is a coarser method of tuning the weights than the method used in the CLEVER algorithm. On the other hand, is not clear if the level of precision provided in the CLEVER algorithm is really needed.

V. Conclusions and Future Directions

In this paper, we have reviewed graph theoretical concepts and algorithms that have been proposed in the literature for searching the web.

Besides affecting better search methods, the results of these approaches can be useful for advertisement and marketing decisions on the web. Currently, web advertisement has mainly focused on finding pages with highest numbers of visitors. This motivated the studies of visitation frequencies, such as [1]. In [1], Adamic suggested that analyzing community structures on the web may be beneficial for better targeting advertisements or political campaigns. If a community is large and heavily connected, placing one ad at a central location may suffice. If community is represented by many small groups, the advertiser would need to place ads to many locations.

Kumar et al. suggested that extracting web communities would allow target advertising at a very precise level [23]. We propose that algorithms like HITS can provide additional insight about good advertisemen locations. A hub page may be visited frequently, but the average user time spent on a hub page is likely to be much less than average user time spent on an authority page. This reasoning suggests that authority pages may be better locations for advertisement than hub pages, even though some hub pages may have higher link density.

We are at the start of a new revolution in education, commerce, and communication made possible by the advancement of the web. Effective search algorithms are at the core of the enabling technology in this new media. Future research needs to focus on a deeper level of understanding the link structure of the web and exploiting this information for more effective uses. The research area is so young that even the known techniques have not yet been studied fully. For example, while relatively more work has been done to understand the behavior of the HITS algorithm and its variants, other ideas based on searching for bipartite graphs and NK-clans have not been studied fully. How can we exploit these structures for topic search? How can we use them for finding pages related to a given URL? How can we use them for page ranking? These and many related questions need to be investigated.

Another area of research could focus on combining the link-based techniques with the user feedback. How can we let the user to guide the link-based search? What parameters do we use to fine-tune the search performance? What protocol should be used for communication between a user and the search algorithm? These and other areas appear to be very fruitful for future research.

Acknowledgments

This research was supported by a grant from the Turkish Scientific and Engineering Research Council, grant No: EEEAG-199E013; a grant from the Louisiana Board of Regents, grant no: LEQSF(1998-01)-RD-A-36; and a grant from the U.S. Department of Energy, grant no. DE-FG02-97ER1220.

References

[1] Lada A. Adamic and Bernardo A. Huberman. "The Nature of Markets in the World Wide Web", Proceedings of Computing in Economics and Finance, 1999, Meetings of the Society for Computational Economics, June 24-26.

[2] Lada A. Adamic. "The Small World Web", in proceedings of ECDL99 in Paris, France.

[3] Reka Albert, Hawoong Jeong, Albert-Laszlo Barabasi. "Diameter of the World-Wide Web" Nature, vol 401, no. 9, 1999, page 130.

[4] Krishna Bharat and Andrei Z. Broder. "A technique for measuring the relative size and overlap of public web search engines" in World-Wide Web'98 (WWW7), Brisbane, Australia, 1998.

[5] Krishna Bharat, Andrei Z. Broder, Monika R. Henzinger, Puneet Kumar and Suresh Venkatasubramanian. "The Connectivity Server: Fast access to linkage information on the Web" in Proceedings of World-Wide Web'98 (WWW7), Brisbane, Australia, 1998.

[6] Krishna Bharat and Monika Henzinger. "Improved Algorithms for Topic Distillation in a Hyperlinked Environment" 21st ACM SIGIR conference on Research and Development in Information Retrieval, pp. 469-477, 1998.

[7] Rodrigo A. Botafogo, Ehud Rivlin, and Ben Shneiderman. "Structural Analysis of Hypertexts: Identifying Hierarchies and Useful Metrics" ACM Trans. Inf. Sys., 10(1992), pp. 142-180

[8] Sergey Brin, and Larry page. "The Anatomy of a Large Scale Hypertextual Web Search Engine" In Proc. of WWW7, Brisbane, Australia, April 1998.

[9] Sergey Brin. "Extracting Patterns and Relations from the World Wide Web", Proceedings of WebDB Workshop at EDBT'98, Valencia, Spain, March 1998.

[10] Soumen Chakrabarti. "Recent results in automatic Web resource discovery", ACM computing survey, 1999.

[11] Soumen Chakrabarti, Byron E. Dom, and Piotr Indyk. "Enhanced Hypertext Categorization using Hyperlinks" in Proceedings of ACM SIGMOD'98, 1998.

[12] Soumen Chakrabarti, Byron E. Dom, David Gibson, Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, Andrew Tomkins. "Spectral Filtering for resource discovery" Proceedings of the SIGIR 98 Workshop on Hypertext Information Retrieval for the Web. Editors: Eric Brown and Alan Smeaton.

[13] Soumen Chakrabarti, Byron E. Dom, Prabhakar Raghavan, Sridhar Rajagopalan, David Gibson and Jon M. Kleinberg. "Automatic Resource Compilation by Analyzing Hyperlink Structure and Associated Text" in Proceedings of World-Wide Web'98 (WWW7), Brisbane, Australia, 65-74, April 1998.

[14] Soumen Chakrabarti, Byron E. Dom, David Gibson, Jon Kleinberg, Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, Andrew Tomkins. "Mining the Link Structure of the World Wide Web" IEEE Computer, Vol.32 No.8,August 1999.

[15] Soumen Chakrabarti, Byron E. Dom, David Gibson, Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, and Andrew Tomkins. "Experiments in Topic Distillation" ACM SIGIR'98 Post-Conference Workshop on Hypertext Information Retrieval for the Web, 1998.

[16] Chandra Chekuri, Prabhakar Raghavan. "Web Search Using Automatic Classification", In Proceedings of the Sixth International World Wide Web Conference, Santa Clara, CA, USA, April 7 - 11, 1997.

[17] Jeffrey Dean, and Monika R. Henzinger. "Finding related Pages in the World Wide Web" In Proc. WWW-8, 1999.

[18] Daniela Florescu, Alon Levy, Alberto Mendelzon. "Database Techniques for the World-Wide Web: A Survey", Sigmond Record, Vol. 27, No. 3, 1998, Pages 59-74

[19] David Gibson, Jon Kleinberg, Prabhakar Raghavan. "Inferring Web Communities from Link Topology" Proc. 9th ACM Conference on Hypertext and HyperMedia, 1998

[20] Ventat N. Gudivada, Vijay V. Raghavan, William I. Grosky, Rajesh Kasanagottu. "Information retrieval on the world wide web," EEE Internet Computing, Vol. 1, No. 5, 1997, pp. 58-68

[21] Jon M. Kleinberg. "Authoritative sources in a hyperlinked environment" in Proceedings of ACM-SIAM Syposium on Discrete Algorithms, 668-677, January 1998.

[22] Jon M. Kleinberg, Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan and Andrew S. Tomkins. "The Web as a graph: measurements, models and methods" Proceedings of the 5th International Computing and combinatorics Conference,1999

[23] Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, Andrew Tomkins. "Trawling the web for emerging cyber-communities", Proc. 8th International World Wide Web Conference, WWW8, 1999

[24] Sougata Mukherjea, James D. Foley, Scott Hudson. "Visualizing Complex Hypermedia Networks through Multiple Hierarchical Views", in Proc. ACM CHI 1995 pp.331-337, Denver, Colarado, USA, ACM Press

[25] Peter Pirolli, James Pitkow, Ramana Rao. "Silk from a Sow's Ear:Extracting Usable Structures from the Web" Procedings of ACM SIGCHI Conference on Human Factors in Computing, 1996.

[26] Gerard Salton. "Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer" Addison Wesley Publishing Co., Reading, MA, 1989.

[27] Ellen Spertus. "Parasite: Mining Structural Information on the Web", Proc. 6th International World Wide Web Conference, 1997.

[28] Loren Terveen and Will Hill. "Finding and Visualizing Inter-site Clan Graphs" Proceedings of CHI 98 448-455, Los Angeles, CA

[29] Ron Weiss, Bienvenido Velez, Mark A. Sheldon, Chanathip Namprempre, Peter Szilagyi, Andrzej Duda, David K. Gifford. "HyPursuit: A Hierarchical Network Search Engine that Exploits Content-Link Hypertex Clustering" In ACM Hypertext, Washington DC, Mar. 1996.

[30] H. D. White, K. W. McCain. "Bibliometrics" in Annual Review of Information Science and Technology, Elsevier, pp. 119-186, 1989.