Improved Link-Based Algorithms for Ranking Web Pages [*]

Ziyang Wang
Department of Computer Science
New York University
New York, NY 10012
ziyang@cs.nyu.edu

ABSTRACT

Several link-based algorithms, such as PageRank [19], HITS [15] and SALSA [16], have been developed to evaluate the popularity of web pages. These algorithms can be interpreted as computing the steady-state distribution of various Markov processes over web pages. The PageRank and HITS algorithms tend to over-rank tightly interlinked collections of pages, such as well-organized message boards. We show that this effect can be alleviated using a number of modifications to the underlying Markov process. Specifically, rather than weight all outlinks from a given page equally, greater weight is given to links between pages that are, in other respects, further off in the web, and less weight is given to links between pages that are nearby. We have experimented with a number of variants of this idea, using a number of different measures of "distance" in the Web, and a number of different weighting schemes. We show that these revised algorithms often do avoid the over-ranking problem and give better overall rankings.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*retrieval models, search process*; F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Non-numerical Algorithms and Problems—*sorting and searching*

General Terms

Algorithms, Experimentation

Keywords

web ranking, stochastic process, Circular Contribution, web local aggregation, hyperlink evaluation, evaluation-based web ranking, PageRank, HITS, SALSA

1. INTRODUCTION

The ranking of Web pages returned in response to a user query combines a measure of the relevance of the page to the query together with a query-independent measure of the quality of the page. The latter measure is based on the structure of the Web, considered as a directed graph of pages and links: A page with many in-links is presumed to be a high-quality page, particularly if (circularly) the links come from pages that are themselves high-quality.

A number of techniques have been developed to rank Web pages based purely on the structure of hyperlinks. The best known of these are the PageRank algorithm[19], used in Google; the HITS algorithm [15], proposed by Kleinberg; and the SALSA algorithm [16]. (The original proposal for HITS and SALSA is to apply them to a collection of pages found to be relevant to a query, but the same algorithms can be applied to any collection of web pages.) These and similar algorithms can be viewed as computing the steady-state distribution of various Markov processes over Web pages.

Algorithms of this kind tend to over-rank tightly interlinked collections of pages, such as message boards. Intuitively, a Markov process executing a random walk through the Web tends to get stuck inside such a collection, crossing links from one of the pages to another. However, the large number of links between these pages does not constitute any independent endorsement of quality, or at best is a comparatively weak endorsement of quality. We call this the "Circular Contribution effect"; effectively, each of the pages in such a collection boosts the evaluation of all the others, and hence, indirectly, of itself.

To counteract this undesirable effect, we propose to modify the underlying Markov processes by giving different weights to different outlinks from a page. A link from P to Q1 is weighted as more important than a link from P to Q2 if, in some sense, Q1 is "further" than Q2 from P in the Web. We present three different definitions of this notion of "distance in the Web" and a number of different weighting schemes. Adapting each of these modifications to the PageRank and HITS algorithms, we show experimentally that the revised algorithm is much less prone to the circular contribution effect. The modifications do not have a measurably favorable effect on the SALSA algorithm, which, in any case, is much less prone to the circular contribution effect.

1.1 Major link-based ranking systems and their properties

We begin by discussing the three algorithms PageRank [19], HITS [15], and SALSA [16] and some of their properties. The detailed algorithms will be presented in Section 4.1.

[*] NYU computer science dept. technical report TR2003-846, June 2003.

This work is supported by NSF grant #IIS-0097537

The stochastic model used in PageRank is a random-walk through the web. In its original model, the web surfer follows the out link from his current page, and jumps to linked pages with equal probabilities; with small probability (0.15) he jumps randomly in the web. The PageRank is the steady-state distribution of this stochastic process.

HITS and SALSA are based on a model of the web that distinguishes hubs and authorities. Each page is assigned a "hub" value and an "authority" value. The hub value of page H is a function of the authority values of the outlinks from H; the authority value of page A is a function of the hub values of the inlinks into A. Borodin et al. [4] show that this can be interpreted in terms of a stochastic process that alternates traveling forward and backward across links.

In the computational models of these stochastic processes, if there exists a path from node i to node j and also a path from node j to node i in the selected web graph, then the rank value of node i will boost the rank value of node j and vice versa; indirectly, each node ends up "endorsing" itself. We call this effect *Circular Contribution.* In general, Circular Contribution tends to be a feature of stochastic models, and often a useful one. But it can also lead to incorrect results. It has often been observed [16, 3, 17, 7] that the mutual reinforcement effect in Kleinberg's HITS algorithm can produce bad ranking distribution for certain web graphs. In general, if there is a Web Local Aggregation in the web graph, discussed in the next section, the Circular Contribution effect can mislead these algorithms into inaccurate rankings.

1.2 Web Local Aggregation effect in stochastic ranking systems

A hyperlink implies some relationship between the linked documents. We divide hyperlinks into two major categories based on their functionalities:

Category 1: Links are used for information reference and information association. For example, students may add university home page to their personal home pages because the university home page is an important information source for university community. Such links works as information reference based on human knowledge. Topic-related pages may link to each other as information association. The author of the source page creates the links because he considers the destination page of value; hence, from the point of view of the search engine, the link is evidence that the user carrying out the search may also consider the destination page of value.

Category 2: Links are used for directories. This makes web browsing meaningful. These links probably provide strong information reference or association. For example, Yahoo! main page contains many links to its subcategories. These links provides strong information reference to help people to navigate the web in order to locate the information they need. But directory links frequently provide weak information reference and association. A discussion board may add links to many messages. The topics of messages may be very diverse and there may be even no association between father page and child pages.

In addition, some links may be used for advertisements, or even no meaningful functionality. These links give much weaker association of the web pages. But since it is almost impossible to identify them based on the web link structure, we won't discuss these effects.

When the hyperlinks functioning as directories gives weak information reference and association, they may cause negative effect in global web ranking. It is very common that the father page hosting many directory links may gain many back links from its children. Thus the father page and child pages form cycles and the Circular Contribution effect happens. In ranking systems that use stochastic models, all pages in such collections will have their ranking raised as a result. The father page will gain high rank with the contribution of many child pages, and consequently all the child pages will benefit from the high rank of father page. We call this effect *Web Local Aggregation.* A Web Local Aggregation is characterized by the property that the number of intra-links within the collection is very large and is significantly larger than those of the in-bound links entering the collection and out-bound links leaving the collection. When information reference and association of the hyperlink is weak in such situation, the ranking biased by Web Local Aggregation does not correctly evaluate the global importance of web pages.

In the dataset that we have used in our experimentation, there is a concrete example of Web Local Aggregation; its negative effect on the accuracy of these ranking algorithms is very clear. From a crawl on NYU web in Oct. 2002, there exists a tightly linked discussion message archive in the NYU Medical School. Four pages host directory links to 958 messages sorted by thread, date, subject and author respectively. Each message has links to the four directory pages, two links to the previous message and two links to the next message. The intra-links of this collection make up 99.6% of all the links which have at least one node in this collection. Running PageRank on the entire NYU web (about 100,000 different pages) gives the rank of these four directory pages from 9 to 12, which is entirely out of scale with the actual importance of these pages to the average user of the NYU web site. Further details are given in section 4.

The issue of local connectivity in the Web was addressed by Lempel and Moran [16], who introduced the concept of Tightly Knit Community (TKC) and discussed an artificial example. TKC effect is not intensively explored in the real web and could be identified only when the ranking is computed using HITS algorithm. The presented concept of Web Local Aggregation here is more general, and is topic-independent and ranking-independent.

1.3 Our work

To overcome the negative effect of local aggregation and improve the effectiveness of ranking systems, we propose the ideas of *Hyperlink Evaluation* and *Evaluation-based Web Ranking.*

In the stochastic models discussed in section 1.1, the transition coefficients associated with the link from P to Q is, in PageRank a function of out-degree(P); in HITS a constant value; and in SALSA either a function of out-degree(P) or a function of in-degree(Q), depending on the direction of the transition. In all three, all the outlinks from P or all the inlinks to Q are evaluated equally. The modifications that we propose, by contrast, give different weights to different links, so as to avoid or alleviate the effect of Web Local Aggregation. Since our objective is to improve topic-independent and global ranking systems, our hyperlink evaluation can only be derived from the web graph structure, not query or topic based. Based on link evaluation and the frameworks of existing stochastic web ranking algorithms, new ranking algorithms are proposed which can alleviate the negative effect of Web Local Aggregation effectively.

The remainder of this paper is organized as follows: Section 2 presents three different methods for hyperlink evaluation: *collection-interlink amplification*, *collection-rank-based evaluation*, and *minimal back-distance based evaluation*. It also describes an algorithm to compute minimal back-distance efficiently. Section 3 discusses how the PageRank, HITS, and SALSA algorithms can be modified to use the methods of hyperlink evaluation. Section 4 presents several experiments on New York University web site and analyzes the effectiveness of the new algorithms.

2. METRICS OF HYPERLINK EVALUATION

We propose two general methods to reduce the effect on rankings of Web Local Aggregation. The first method is to weight cross-links between different domains more strongly than links between pages in a single domain. Presumably local aggregations in the Web will generally lie in a single domain. Moreover, for any two pages P and Q, it is more likely that P and Q represent substantially different sources of information (authors, organizations, cyber-communities etc.), if P and Q are from different domains than if they are from the same domain; hence, a link from P to Q is, on average, a stronger endorsement of Q if P and Q are from different domains.

The second general method is to weight the link from P to Q more strongly as a function of the length of the shortest path in the Web from Q to P. This directly attacks the Circular Contribution effect, as the contribution of short cycles in the Web to the evaluation of web pages is reduced.

Based on these two methods, we present three metrics of hyperlink evaluation here: collection-interlink amplification, collection-rank-based evaluation and minimal back-distance based evaluation. And for each of these metrics, we discuss the web surfer behavior in the random walk of the stochastic process guided by the link evaluation.

2.1 Metric 1: collection-interlink Amplification

A number of different approaches for clustering the web into non-overlapping web collections (e.g. [21]), but these are mostly based on similarity of content, and thus not link-based. Our analysis simply clusters the web using host information. That is, two pages are placed in the same collection if and only if they belong to the same host. This metric evaluates cross links between different collections greater than links within the collections by a fixed ratio. Mathematically, denote v_{ij} as the value of link $i \rightarrow j$, $Coll(i)$ as the collection index of page i. Let m_i be the number of out links from page i within the collection and n_i be the number of out links from page i that leave the collection. Let $C_i = \delta m_i + n_i$. The value of link (i, j) is given as

$$v_{ij} = \begin{cases} \delta / C_i & if\ Coll(i) = Coll(j) \\ 1/C_i & otherwise \end{cases} \tag{1}$$

where δ is a constant within $[0, 1]$. The coefficient C_i normalizes the values v_{ij} such that the sum of v_{ij} is 1 respect to index j. When $\delta = 1$, the evaluation is exactly the uniform link evaluation used by the ranking systems discussed in section 1.1. The value $1/\delta$ gives the ratio how cross links between collections are evaluated greater than local links. By giving more evaluation on cross links, the surfer can obtain higher probability to jump out of web local aggregation that frequently happens within a collection.

2.2 Metric 2: collection-rank-based evaluation

The second metric is also devised using web collection information but an improvement of Metric 1. We consider not only the cross link between collections, but also the reliability of the destination host of a link. We use a two-stage evaluation approach to evaluate the importance of pages in the web graph: the collections are evaluated first then hyperlink evaluation can be improved using collection evaluation. For collection evaluation, if a collection is well linked by others, it would be very valuable. This argument encourages us to adopt the existing stochastic model in link-based ranking algorithms. As a demonstration, we show how to evaluate collections using a PageRank-like algorithm. Suppose there are q different collections in the web graph and let N_{ij}^{Coll} be the number of outer links from collection i to collection j. Then the transition matrix T^{Coll} is

$$T^{Coll} = \epsilon U^{Coll} + (1 - \epsilon) M^{Coll} \tag{2}$$

where U^{Coll}, $U_{ij}^{Coll} = 1/q$, and M^{Coll} is

$$M^{Coll} : M_{ij}^{Coll} = \frac{N_{ij}^{Coll}}{\sum_k N_{ik}^{Coll}} \tag{3}$$

The ranking value vector of collections, denoted as V^{Coll}, is the stationary distribution of transition matrix T^{Coll}. Compare to the mathematical representation of PageRank discussed in Section 3.1, we can see they are in the same mathematical framework.

When the collection evaluation V^{Coll} is given, we can define the hyperlink evaluation as follows

$$v_{ij} = \begin{cases} \frac{\delta \times S(V_{Coll(j)}^{Coll})}{C_i} & if : Coll(i) = Coll(j) \\ \frac{S(V_{Coll(j)}^{Coll})}{C_i} & otherwise \end{cases} \tag{4}$$

where C_i is the normalization coefficient such that the sum of v_{ij} is 1 respect to index j, and $S : R \rightarrow R$ is an increasing function to scale the results of collection evaluation. This link evaluation corresponds to the following surfer behavior in a random walk: during the walk, the surfer can "see" the collection information of its neighbors. When he makes the transition selection, he prefers to jumping to the page whose collection is more trustable. A great difference between this evaluation to Metric 1 is that the pages in highly valuable collection gain higher values. In fact, the directory links in highly valuable collections tend to give stronger information reference and association. A notable web directory collection is much more helpful for people to navigate the web than a discussion board served to a small community group. If Web Local Aggregation happens in both collections, the effect in the latter would be worse than that in the former. In this context, Metric 2 is an improvement of Metric 1.

2.3 Metric 3: minimal back-distance (MinBD) based evaluation

This metric directly identifies the cycle information in the web graph and evaluates the links accordingly. In Circular Contribution, we know the mutual contribution of two nodes is affected by the length of the path between them. The smaller the length is, the greater the contribution is. Thus, for each link $i \rightarrow j$, the backward contribution of node j to node i is dominated by the minimal length of all possible paths from node j to node i in the web graph. We call this length *minimal back-distance (MinBD)*. Denote $MinBD_{ij}$ as the minimal back distance from j to i. For each link $i \rightarrow j$ the mathematical representation of this metric is

$$v_{ij} = \begin{cases} \frac{f(\Omega)}{C_i} & if\ MinBD_{ij} \geq \Omega \\ \frac{f(MinBD_{ij})}{C_i} & otherwise \end{cases} \tag{5}$$

where Ω is a threshold, C_i is the normalization coefficient such that the sum of v_{ij} is 1 respect to index j, $f : R \rightarrow R$ is an increasing function satisfying $f(0) \geq 0$. A threshold Ω is set because MinDB could be infinite if there exists no backward path. Any finite MinDB is less than or equal to the diameter of the web inferred from the definition of MinDB. Albert et al. [1] theoretically estimated that the web diameter is 19 in 1999. Broder et al. [5] experimentally determined that the diameter of the web is at least 28. Although the web increases dramatically, it is shown that the diameter of web increases very slowly with the degree of the logarithm of the web size [1]. Thus, the threshold Ω is set as 30 which can fit the current web size quite well.

Let's consider the random walk guided by this link evaluation. Suppose the web surfer can "see" further than the neighbors at any page. An effective web walk strategy should explore the web as much as possible and not go back to the visited local pages frequently. When determining the next transition, the surfer preferentially jump to the linked pages with higher MinDB. Consequently, the probability that the surfer goes back to the current node is reduced compared with uniform transition selection. Furthermore, as those links with small MinDB are evaluated less, the Circular Contribution effect is reduced, which can effectively alleviate the negative effect of Web Local Aggregation whose web graph contains many short cycles.

People may argue why not comprehensively evaluate all cycles in the web graph other than those with minimal length. The computation and memory cost of evaluating all cycles in the web graph is very high considering the huge size of the web. As a trade-off, our evaluation employs MinDB only which can be efficiently computed with the algorithm presented in Section 2.3.1.

2.3.1 Compute MinDB efficiently

The biggest problem in Metric 3 is that there is no good algorithm to compute minimal back distance very efficiently. At a glance, this is a all-source shortest path problem, which can be solved in $O(|V| \times |E|)$ where $|V|$ and $|E|$ are the number of vertices and number of edges of a given graph. Consider the size of global web, an accepted algorithm MUST run linear or almost linear as PageRank or HITS does. As we are only interested in computing minimal back distance of neighboring pages, it is not necessary to compute all-source shortest path. Here we present an algorithm which can compute MinDB efficiently.

For simplicity, we assume the web graph is a directed unweighted graph without any link from and to the same page. Our algorithm starts with a depth-first-search (DFS), repeatedly updates the minimal back distance in the DFS when new cycles are encountered. We color each graph node with WHITE, GREY and BLACK representing whether it is *unvisited*, *visited but not finished* and *finished* in DFS. For each page i, the algorithm stores a list of *cycle ancestors*, each of which locates in the DFS path from root page to page i and there exists a cycle containing the *cycle ancestor* and i. And for each *cycle ancestor* of page i, the algorithm stores the minimal distance from i to the *cycle ancestor* in convenience of future update of MinDB in DFS. The algorithm picks a root page and runs recursively on the following procedure:

```
DFS (page i)
BEGIN
  Color page i as GREY
  FOR each link i->j
    IF page j is WHITE    /* tree link */
      DFS(j)
    ELSE IF page j is GREY   /* back link */
      UPDATE_DFSPATH(j,i)
    ELSE IF page j is BLACK   /* cross link */
      FOR each cycle ancestor k of page j DO
        IF k is GREY
          UPDATE_DFSPATH(k,i)
  Color page i as BLACK
END
```

where the procedure UPDATE_DFSPATH() is defined as

```
UPDATE_DFSPATH(page p, page q)
BEGIN
  FOR each link k->l along DFS path p to q DO
    update MinDB of link k->l
    store cycle ancestor p in node l
    update minimal distance from l to p
END
```

The procedure UPDATE_DFSPATH() here is rather scratch. The actual procedure involves many conditional checks to correctly update MinDB. The correctness of this algorithm comes from the following lemmas:

LEMMA 1. *In DFS, if a back link (a link pointing to a GREY node) is found, a new cycle is found.*

LEMMA 2. *In DFS, if a cross link (a link pointing to a BLACK node) $i \rightarrow j$ is found, there exists a cycle containing this link if and only if there is a cycle containing j with ancestor of a GREY node.*

The first lemma is obvious. We state the proof of the second one here:

PROOF. For an inter link $i \rightarrow j$, if there is a cycle containing j with ancestor of a GREY node k, then there exists a DFS path from k to i, denoted as $k \rightsquigarrow i$. Denote the arc from j to k in the cycle containing j and k as $j \rightsquigarrow k$. We see $i \rightarrow j$, $j \rightsquigarrow k$ and $k \rightsquigarrow i$ is a cycle containing the link $i \rightarrow j$. On the other hand, if there exists a circle C within all the links found so far in DFS, which contains link $i \rightarrow j$, C must contain at least one GREY node because the page i is the current node in DFS and it must be linked by its parent, which is a GREY node. Furthermore, the nodes with minimum DFS depth in such a cycle must also contain a GREY node. Otherwise, their exists a link on this cycle satisfying that it links from a BLACK node to a GREY node and the BLACK node has smaller DFS depth. This contradicts with the property of DFS. Proof done. □

The main structure of this algorithm is a DFS. It is highly possible there exist more than one DFS tree. This situation will not damage our algorithm as any cross-link between different DFS trees will never be in a cycle. The actual web graph may contain a huge number of cycles. As we only store the cycle ancestors for each page and the minimum distances from the page to cycle ancestors, we avoid to repeatedly check all possible cycles in computation. This saves a lot of computation and memory to improve the efficiency of our algorithm. Since DFS is certainly linear, the computing complexity of a single DFS node depends on its DFS depth of the graph. If the graph is malformed, and the DFS search ends with a very large depth, the computing cost may be high with high average DFS depth. Empirically, we compare this algorithm with another linear, but very limited algorithm described as follows:

In the DFS of the graph, for each page, we use breath-first-search to search a limited steps, say up to depth 4. From current literature and our own data set, the average number of outer links per page is about 8. This will contribute a constant factor $8^4 \sim 4000$. Our experiments in Section 5 show our algorithm is much faster than this one. Our experiments also show that even the strongly connected component of the web is significantly large, the DFS depth is still relatively small due to the sparse nature of the web.

3. MODIFY STOCHASTIC PROCESS USING HYPERLINK EVALUATION

Supported by the three metrics of hyperlink evaluation, we propose a new ranking strategy named *Evaluation-based Web Ranking*. The basic idea here is that by applying link evaluation to the stochastic ranking algorithms, the new algorithms can improve ranking effectiveness compared with the original ones. We first review the mathematical frameworks of three different algorithms here: PageRank [19], HITS [15] and SALSA [16]. Then we discuss how to apply hyperlink evaluation into these ranking frameworks to produce new improved algorithms. The improved algorithms are independent with what actual link evaluation metric is used.

3.1 The mathematical framework of three existing ranking algorithms

3.1.1 PageRank algorithm

A web graph of n pages can be represented using an n-by-n adjacency matrix A, where $A(i, j)$-element is 1 if page i links to page j, and 0 otherwise. The PageRank algorithm first constructs a probability transition matrix M by normalizing each row of adjacency matrix A to sum to 1. The idea of this algorithm is inferred from random walk within a graph. When a person is at page i, with probability $(1 - \epsilon)$, it uniformly picks a URL link from this page and transits to the target page of this link. With probability ϵ, it jumps to any other page in the web with uniform probability. The final transition matrix T is

$$T = \epsilon U + (1 - \epsilon)M \tag{6}$$

where U is an n-by-n matrix of uniform transition probabilities having $U_{ij} = 1/n$ for all $1 \leq i, j \leq n$. The vector of PageRank scores p is then defined to be the stationary distribution of this Markov chain. The stationary distribution is the eigenvector of the transition matrix and satisfies

$$(\epsilon U + (1 - \epsilon)M)^T p = p \tag{7}$$

The vector p gives the importance scores which can be ordered as ranks.

3.1.2 HITS algorithm

The HITS algorithm points that a page has high "authority" weight if it is linked to by many pages with high "hub" weight, and that a page has high hub weight if it links to many authoritative pages. Given a set of n web pages , the HITS algorithm firstly forms the n-by-n adjacency matrix A, whose (i, j)-element is 1 if page i links to page j, and 0 otherwise. It iterates the following equations:

$$a^{t+1} = A^T \cdot h^t \tag{8}$$

$$h^{t+1} = A \cdot a^{t+1} \tag{9}$$

where a and h are the vectors of authority values and hub values.

For each iteration, a normalization step is then applied, so that the vectors a and p become unit vectors in some norm. Kleinberg [15] proves that for a sufficient number of iterations the vectors a and h converge to the principle eigenvectors of the matrices $A^T A$ and AA^T.

A more stable HITS is given by Ng et al. [18]. It introduces a small uniform jump probability to any other page for each transition as PageRank algorithm does. The mathematical representation of the stable HITS algorithm is:

$$a^{t+1} = \epsilon \overrightarrow{1} + (1 - \epsilon)A^T \cdot h^t \tag{10}$$

$$h^{t+1} = \epsilon \overrightarrow{1} + (1 - \epsilon)A \cdot a^{t+1} \tag{11}$$

3.1.3 SALSA algorithm

The SALSA algorithm is built in the same framework of authority and hub like HITS algorithm. It normalizes the adjacency matrix A of the web graph by rows and columns respectively to obtain new transition matrices for authority and hub computation. Denote by R the matrix which results by dividing each nonzero entry of A by the sum of the entries in its row, and by L the matrix which results by dividing each nonzero element of A by the sum of the entries in its column. Then the SALSA algorithm iterates the following equations:

$$a^{t+1} = L^T \cdot h^t \tag{12}$$

$$h^{t+1} = R \cdot a^{t+1} \tag{13}$$

where a and h are the vectors of authority values and hub values. The computation model is very similar to HITS algorithm given in Section 3.1.2. The authority and hub vectors converge to the principle eigenvectors of matrices $L^T R$ and RL^T.

3.2 PageRank-based algorithm using link evaluation

In PageRank algorithm, the small coefficient of uniform jumping probability, ϵ, counts for an important factor of PageRank algorithm [18] and can be used to bias PageRank value distribution [13]. We propose two levels of Evaluation-based Web Ranking for PageRank-like algorithm: the first level of evaluation is that the PageRank transition probabilities can be evaluated by link evaluation; the second level of evaluation is that the uniform jumping probability ϵ for each page can be evaluated by the link evaluation on that page. These findings lead to the discovery of two improved algorithms in the mathematical framework of PageRank algorithm: *fixed ϵ algorithm* and *evaluation-based non-fixed ϵ algorithm.*

3.2.1 Level 1: fixed ϵ algorithm

The original PageRank transition matrix M in Section 3.1.1 is modified as

$$\tilde{M} : \tilde{M}_{ij} = v_{ij} \tag{14}$$

Where v_{ij} is any one of the link evaluations given in Section 2. Since the link values have been normalized, $\tilde{M}$ is a well formed transition matrix. Clearly, we replace the original matrix M with $\tilde{M}$ in Section 3.1.1 and apply the fixed uniform jumping probability to form the final transition as Eq.(6) does.

3.2.2 Level 2: evaluation-based non-fixed ϵ algorithm

The hyperlink evaluation is the first level evaluation in reducing the local aggregation of the web. In the second level, we evaluate the overall quality of all out links from a page, then decide how to choose ϵ. For any page i, we calculate the average of hyperlink values of all out links on page i. Because the normalization of hyperlink evaluations in Section 3 is performed on different bases for pages with different number of out links, we use the non-normalized evaluation to calculate average quality of links on a single page. The average quality of the out links on page i is defined as

$$AvgQual(i) = \frac{\sum_{j:i \to j} V_{ij}}{N_i} \tag{15}$$

where $V_{ij} = v_{ij} \times C_i$ is the unnormalized link value and N_i is the number of outlinks on page i. The uniform jumping probability for page i, ϵ_i, is then evaluated as

$$\epsilon_i = \epsilon_{base} + \epsilon_{adjust} \times g(AvgQual(i)) \tag{16}$$

where ϵ_{base} is the global base and ϵ_{adjust} is the adjustment range. $g()$ is an decreasing function and maps $AvgQual(i)$ into $[0, 1]$. For example, if $AvgQual(i)$ is in $[1, \infty]$, $g(x) = \frac{1}{x}$ satisfies. The greater value $AvgQual(i)$ has, the smaller probability it will jump to an arbitrary page.

Let p^t be the ranking vector after iteration t. Iteratively computing rank is shown as follows,

$$p_i^{t+1} = \sum_j \frac{\epsilon_j \times p_j^t}{n} + \sum_{j:j \to i} (1 - \epsilon_j) \times v_{ji} \times p_j^t \tag{17}$$

where n is the total number of pages in the web graph. To reduce complexity, the first term must be calculated only once for each iteration. And after each iteration, vector p^t must be normalized.

3.3 HITS-based algorithm using link evaluation

To modify stochastic process in HITS ranking system, we can apply the link values as the weights to HITS algorithm. The non-zero entries of new transition matrix $\tilde{A}$ is defined as

$$\tilde{A} : \tilde{A}_{ij} = \frac{v_{ij}}{\max_{j:i \to j} v_{ij}} \tag{18}$$

where $i \to j$ is a link in the web graph. All entries of $\tilde{A}$ are within [0,1] in this mathematical representation. Let N_i be the number of out links on page i. In uniform link evaluation, link values are $v_{ij} = 1/N_i$ in the normalized form, and $\max_{k:i \to k} v_{ik}$ is also $1/N_i$. Therefore the new algorithm under uniform link evaluation is exactly the original HITS algorithm. Replacing A^T and A in the stable HITS algorithm Eq.(10) and Eq.(11) with $(\tilde{A})^T$ and $\tilde{A}$, we get our improved HITS-based algorithm.

Table 1: Selected rank of PageRank

Rank	URL
1	www.nyu.edu
2	www.nyu.edu/bin/phfnyu
3	www.nyu.edu/library/bobst
4	www.stern.nyu.edu/acc/about
5	www.nyu.edu/gsas
6	www.law.nyu.edu
7	www.med.nyu.edu
8	monod.biomath.nyu.edu/index/papers.html
9	mchip00.med.nyu.edu/lit-med/archives/messages/date.html
10	mchip00.med.nyu.edu/lit-med/archives/messages/index.html
11	mchip00.med.nyu.edu/lit-med/archives/messages/author.html
12	mchip00.med.nyu.edu/lit-med/archives/messages/subject.html
13	www.law.nyu.edu/index.html
14	rmm-java.stern.nyu.edu/jmis/toppage/index.html
15	www.law.nyu.edu/search

3.4 SALSA-based algorithm using link evaluation

We keep the mathematical model of our algorithm consistent with that of SALSA provided in Section 3.1.3, but biased by our link evaluation. Two new normalized matrices, $\tilde{R}$ and $\tilde{L}$, are defined corresponding to matrices R and L defined in Section 3.1.3.

$$\tilde{R} : \tilde{R}_{ij} = v_{ij} \tag{19}$$

$$\tilde{L} : \tilde{L}_{ij} = \frac{N_i \times v_{ij}}{K_j} \tag{20}$$

where N_i is out-degree of page i, and K_j is the in-degree of page j. In uniform link evaluation, where $v_{ij} = 1/N_i$ in the normalized form, matrices $\tilde{R}$ and $\tilde{L}$ are exactly the same of R and L. Replacing R with $\tilde{R}$, L with $\tilde{L}$ respectively in the mathematical model of SALSA given by equations Eq.(12) and Eq.(13), we obtain our improved SALSA-based algorithm.

4. EXPERIMENTS

Our experiment data set is built on a crawl on the entire web of New York University in October, 2002. The crawler narrows its downloading to the URLs with host name ending with ".nyu.edu". After removing repetitions of web pages and non-text pages, the total number of URLs is 98,349, which is compatible with the number of results returned by Google [12] at the same time using query "-abc3dfg9 site:nyu.edu", which is about 102,000. The current size may have some increase due to the web expansion. These URLs make up a web graph consisting 723,380 hyperlinks. The ratio between number of hyperlinks and URLs is 7.4, which is remarkably in agreement with previous work [5, 9]. The study of the in-link and out-link distributions shows they obeys power law with power coefficients of 1.94 and 2.24. Both of them are in very good agreement with previous work studied by Broder el al. [5] and Barabasi and Albert [2]. Although the university web site is not significantly large, it has some good properties: unlike many commercial web sites, most of the web pages are stored statically on server side, not generated by web tools like CGI and ASP; the web pages are geographically close and well linked such that link-based algorithms can produce meaningful results; the university web site is proved experimentally to satisfy many known statistical properties of the global web such as in-degree and out-degree distribution. Basically, PageRank-based algorithms are run on different data sets as HITS-based or SALSA-based algorithms. For

Table 2: Selected authority rank of stable HITS

Rank	URL
1	apple.cs.nyu.edu/proteus/local/data/OnlineProc-/ACL02/MAIN/index.htm
2	apple.cs.nyu.edu/proteus/local/data/OnlineProc-/ACL02/EMNLP/index.htm
3	apple.cs.nyu.edu/proteus/local/data/OnlineProc-/ACL02/S2S/index.htm
4	apple.cs.nyu.edu/proteus/local/data/OnlineProc-/ACL02/DIALOG/index.htm
5	apple.cs.nyu.edu/proteus/local/data/OnlineProc-/ACL02/DEMOS/index.htm
6	apple.cs.nyu.edu/proteus/local/data/OnlineProc-/ACL02/WSD/index.htm
7	apple.cs.nyu.edu/proteus/local/data/OnlineProc-/ACL02/BIO/index.htm
	...
25	apple.cs.nyu.edu/proteus/local/data/OnlineProc-/ACL02/BIO/contents.htm
26	www.nyu.edu
27	apple.cs.nyu.edu/proteus/local/data/OnlineProc-/ACL02/WSD/contents.htm
28	www.nyu.edu/bin/phfnyu

PageRank-based algorithms, they can be applies directly on a well linked web graph. For HITS-based or SALSA-base algorithms, they are generally applied on topic related data sets and require to build the "base set" before doing iterations [15, 16]. But in Section 1.1 and Section 3.1.1, we have known that their computation models are independent with data set. Therefore, we apply HITS-base and SALSA-based algorithms directly on our data set for analyzing purpose. To make ranking unique in each of such ranking systems, we choose *authority* ranking as the evaluation of importance of web pages because authority pages are computed based on in-links.

We have found that Web Local Aggregation in the NYU web does indeed have a negative effect on the rankings output by the standard algorithms. Table 1 shows the top 15 pages given by PageRank algorithm by setting $\epsilon = 0.15$ in Eq.(6). The pages ranked from 9 to 12 are the directory pages of a message board archive in the medical school of NYU. Each of the directory pages hosts the links to the same collection of 958 message pages of different orders. And each message page has links to the four directory pages and links to previous and following messages. These directory pages are ranked unexpectedly high in the whole collection of about 100,000 pages, as compared to their actual significance, intuitively judged. However, since there are many short cycles in this collection, the Circular Contribution effect is significant and produces this inaccurate ranking distribution. Table 2 shows the selected top ranked pages using stable HITS algorithm by setting $\epsilon = 0.2$ in Eq.(10) and Eq.(11). A local collection describing a project called "Proteus" takes top 25 positions higher than the portal page of NYU. This collection contains some mirrors of several online conference proceedings which gain high ranking in the mutual reinforcement environment.

In support of our theoretical analysis in the previous sections, we implement the following algorithms: the original PageRank, stable HITS, and SALSA algorithms; PageRank-based Level 1 evaluation algorithms using link evaluations of Metric 1 and Metric 2; PageRank-based Level 1 and Level 2 evaluation algorithms using link evaluation of Metric 3; HITS-based and SALSA-base algorithms using link evaluations of Metric 1, Metric 2 and Metric 3. There are 13 different algorithms in total. We use the following abbreviations to denote them in the same order: PR, HT, SA, PRM1, PRM2, PRL1M3, PRL2M3, HTM1, HTM2, HTM3, SAM1, SAM2, SAM3. All these algorithms are applied on the same data set stated before. The precise algorithms use the following configurations: in PageRank-based algorithms we set $\epsilon = 0.15$ as Google does; in HITS-based algorithms we adopt the model of stable algorithm of Eq.(10) and Eq.(11) and set $\epsilon = 0.2$ as Ng el al. [18] does; in link evaluations we set the in-collection coefficient $\delta = 0.2$ in both Metric 1 and Metric 2, $S : S(x) = \sqrt[3]{x}$ in Eq.(4) to scale collection rank in Metric 2, and $f : f(x) = \sqrt{x}$ in Eq.(5) to scale MinDB values in Metric 3; in PageRank-base algorithm using Level 2 evaluation we set $\epsilon_{base} = 0.1$, $\epsilon_{adjust} = 0.1$, and $g : g(x) = 1/x$ since $AvgQual()$ maps into $[1, \infty)$. Some of these configurations are made to compatible with previous works and some are made for computational correctness. For example, with the chosen ϵ_{base} and ϵ_{adjust}, the first term in Eq.(16) , $\sum_j \frac{\epsilon_j \times p_j^t}{N}$, converges to 0.151 after tens of iterations, which is almost equivalent to the same setting of ϵ in the original PageRank algorithm. In computing MinDB of links used in Metric 3, we compare our algorithm proposed in Section 2.3.1 with the limited algorithm also described in the same section. The time expense of the first algorithm on our data set takes less than 10% of the time cost by the limited one. This shows empirically the algorithm to compute MinDB is efficient. The depth-first-search starting at the university main page ends with maximum depth of 1645. As all URL can be reached from this portal page according to the crawl, this number is relatively small compared with the size of whole collection.

Given those improved ranking algorithms, the greatest challenge is how to evaluate them compared with the original one. Kendall's τ distance measure can be utilized to compare ranked list and its effectiveness in applying to web ranking has been discussed in [11, 13]. This measure is very limited for our evaluation purpose because it can only tell how different ranking agree with one another. We use an *aggregation-tracing* method to evaluate the effectiveness for different ranking algorithms in avoiding negative effect of Web Local Aggregation: first, we manually identify several sub collections of local aggregation in our data set; second, we trace how their ranks change for different ranking systems.

The manually selected sub collections of local aggregation are given in Table 3. Collection MCHIP and END98-01 have similar structure which contains four directory links hosting the same message archives. PROT is the collection taking the top positions in the ranking computed by HITS algorithm. For in-bound links entering the collection and out-bound links leaving the collection, only those within NYU web are counted. In the collection of ARTG, 844 outbound links point to www.cs.nyu.edu/artg/telecom/fall99/index.html, which links to the directory page of collection ARTG. A common property of these collections is that the number of intra-links within the collections is very large. The ratios of intra-links versus collection size are 13.5, 8.5, 14.8, 15.2, 15.0, 15.0, 8.2, displayed in the same order of the selected collections in Table 3. These ratios are larger than the average ratio 7.4 of our data set. The percentages of intra-links respect to the involved links (i.e. the sum of intra-links, in-bound links and out-bound links) are 99.6%, 99.6%, 99.2%, 99.2%, 97.5%, 95.1% and 81.0%, which are also displayed in the same order of the selected collections in Table 3. The local structures of these collections comply with our description of Web Local Aggregation. In the following discussions, we'll see that the

Table 3: Humanly identified sub collections of Web Local Aggregations

Abbrev.	URL prefix	dir. URL suffix (.html)	Size	Intra-links	In-bound	Out-bound
MCHIP	mchip00.med.nyu.edu/lit-med/archives/messages/	index, date, subject, author	962	12967	2	45
PROT	apple.cs.nyu.edu/proteus/local/data/OnlineProc/	ACL02/MAIN/index.htm, etc.	875	7469	6	26
END98	endeavor.med.nyu.edu/pipermail/lit-med/1998/	index, date, subject, author	193	2850	10	12
END99	endeavor.med.nyu.edu/pipermail/lit-med/1999/	index, date, subject, author	255	3876	10	21
END00	endeavor.med.nyu.edu/pipermail/lit-med/2000/	index, date, subject, author	223	3336	10	76
END01	endeavor.med.nyu.edu/pipermail/lit-med/2001/	index, date, subject, author	345	5162	10	255
ARTG	www.cs.nyu.edu/artg/telecom/fall99/lecture_notes/	lecture_notes	504	4148	3	966

Table 4: Effectiveness of improved PageRank-based algorithms than PageRank algorithm

Coll.	Avg(PR)	ADiff(PRM1,PR)	ADiff(PRM2,PR)	ADiff(PRL1M3,PR)	ADiff(PRL2M3,PR)
MCHIP	12015.7	1744.2	2084.8	5927.8	5743.2
PROT	59670.4	-1985.2	-2490.5	-8751.0	-8144.7
END98	14199.5	1963.6	2490.5	3768.1	3507.8
END99	14485.9	1986.4	2547.7	3854.7	3717.7
END00	15666.9	1700.9	2243.4	2543.1	2388.9
END01	17634.0	1459.0	1850.9	1863.7	1719.1
ARTG	25024.9	65.1	299.4	539.3	916.9
Coll.	Top(PR)	HDiff(PRM1,PR)	HDiff(PRM2,PR)	HDiff(PRL1M3,PR)	HDiff(PRL2M3,PR)
MCHIP	9	9	26	25	24
PROT	496	127	108	55	56
END98	139	67	82	59	59
END99	95	60	70	54	55
END00	129	55	66	42	40
END01	73	44	53	42	38
ARTG	47	6	7	20	21

top rank of these collections are very high in the overall ranking given by three original algorithms as we predicted in Section 1. Tracing the rank change of these local aggregations will give us the clue how our algorithms can improve ranking quality.

In measuring the change of different ranking, we use the average difference $ADiff$ and difference of highest rank $HDiff$ to evaluate the rank change of a sub collection between two different ranking distributions. Let Φ be a sub collection of local aggregation, $R_1(i)$ and $R_2(i)$ be the ranks of page $i \in \Phi$ in two different ranking distributions. The measures of $ADiff$ and $HDiff$ are defined as follows:

$$ADiff(\Phi, R_2, R_1) = \frac{\sum_{i\in\Phi}(R_2(i) - R_1(i))}{Size(\Phi)} \qquad (21)$$

$$HDiff(\Phi, R_2, R_1) = \min_{i\in\Phi} R_2(i) - \min_{i\in\Phi} R_1(i) \qquad (22)$$

Suppose the negative effect of Web Local Aggregation happens on collection Φ in rank R_1. If the value of $ADiff(\Phi, R_2, R_1)$ or $HDiff(\Phi, R_2, R_1)$ is positive, it shows such negative effect has been alleviated by R_2, otherwise not. The larger such value is, the more effective the ranking R_2 is. Table 4, Table 5, and Table 6 give our main results. Analyzing the data in these tables, we draw the following conclusions as our major results of this paper:

PageRank and HITS algorithms are much more sensitive to Web Local Aggregation than SALSA algorithm. In PageRank and HITS algorithms, all selected collections except PROT have average rank within top 27% of the whole data set. The highest rank of MCHIP given by PageRank is 9 as we have pointed out in Table 1. The top rank of PROT given by HITS is 1 which is surprisingly bad as pointed out by Table 2. But all the average ranks given by SALSA algorithm for the collections other than MCHIP are lower than the top 55% of the total data set, which shows SALSA algorithm is much less affected by Web Local Aggregation. Even for MPICH, it is 25% lower than that of PageRank and HITS. The top ranked pages of all selected collections, i.e. the Top(PR), Top(HT), Top(SA) in Table 4, 5, 6, are all ranked very high in the whole collection. All of them are in the top 496 pages and are within top 0.5% of the whole data set. And the average for Top(PR), Top(HT) and Top(SA) are 141.0, 153.3, and 120.7 respectively.

PageRank and HITS algorithms give similar rank distribution. The average ranks of the selected collections given by PageRank and HITS algorithms obey the same order: MCHIP < END98-01 < ARTG < PROT. For the four collections of END98-01, the average ranks are close to one another both in the ranking given by PageRank and HITS algorithms. The standard variations of them are 1401.8 and 491.5, and are within 9% and 3% of the cumulative average of Avg(PR) and Avg(HT). The average of $|Avg(PR) - Avg(HT)|$ for selected collections is 5184.1, which counts for 5.3% of the size of total data set.

The ranking systems using our hyperlink evaluations are very successful to improve ranking effectiveness in PageRank-based and HITS-based algorithms. Table 4 presents the results given by PageRank-based algorithms. Since PROT is least affected by Web Local Aggregation, we consider the collections other than PROT. Compared with the original PageRank algorithm, all new algorithms successfully improve the ranking biased by Web Local Aggregation. The magnitude of the improvement on $ADiff$ respect to link evaluations follows the same order: Metric 1 < Metric 2 < Metric 3. Furthermore, the positive values of $HDiff$ measure show that the highest ranks of these collections decline. For the

Table 5: Effectiveness of improved HITS-based algorithms than HITS algorithm

Coll.	Avg(HT)	ADiff(HTM1,HT)	ADiff(HTM2,HT)	ADiff(HTM3,HT)	Top(HT)	HDiff(HTM1,HT)	HDiff(HTM2,HT)	HDiff(HTM3,HT)
MCHIP	7394.9	3140.2	14296.7	16374.6	104	93	-24	279
PROT	40691.4	9603.3	17353.3	26145.8	1	0	607	2478
END98	19288.4	4935.3	8644.1	11957.2	273	291	-12	957
END99	18112.1	4729.9	10611.2	13041.2	211	237	-6	785
END00	18410.0	4776.7	10292.5	14193.1	250	248	-19	868
END01	18064.1	4560.1	12004.8	10513.1	133	167	37	650
ARTG	25825.3	6107.7	4602.6	16812.5	101	89	-4	1826

Table 6: Effectiveness of improved SALSA-based algorithms than SALSA algorithm

Coll.	Avg(SA)	ADiff(SAM1,SA)	ADiff(SAM2,SA)	ADiff(SAM3,SA)	Top(SA)	HDiff(SAM1,SA)	HDiff(SAM2,HT)	HDiff(SAM3,SA)
MCHIP	36960.3	-8972.3	-17672.0	5327.5	28	7	6	-4
PROT	58713.0	-312.2	-2588.9	-10373.3	229	69	123	61
END98	55176.6	-6373.2	-29034.9	11558.9	173	26	4	-25
END99	55601.6	-5468.4	-28752.1	11191.2	138	20	8	-22
END00	54706.6	-5698.4	-28133.2	-18.5	153	33	9	14
END01	55591.5	-4671.0	-27433.3	550.4	105	8	8	2
ARTG	60030.4	-7291.9	-28787.4	-1888.6	19	9	22	77

two-level evaluations proposed in Section 3.2, Level 2 evaluation has no improvement respect to Level 1 evaluation. This does not comply with our theoretical motivation and the answer still remains open for larger scale experimenting. Table 5 presents the results given by HITS-based algorithms. The improvement of new ranking distribution is significant. The top rank of PROT drops from 1st to 608th in HTM2 and 2479th in HTM3. The magnitude of the improvement on $ADiff$ respect to link evaluations follows the same order as PageRank-based algorithms: Metric 1 $<$ Metric 2 $<$ Metric 3. As for results induced by $Hdiff$ measure, the situation is not always better as shown for Metric 2 evaluation. The strength of HITS-based algorithm using Metric 3 is significant in the top rank decline for all selected collections.

SALSA-based ranking algorithms using our link evaluations have limited success to improve ranking effectiveness. Table 6 shows the results of similar experiments using SALSA-based algorithms. SAM1 and SAM2 does not improve the ranking effectiveness under *ADiff* measure. SAM3 has limited success to improve ranking on 4 of the 7 selected collections. The improvement under *HDiff* measure is sound in SAM1 and SAM2, but not SAM3. As we have pointed out, SALSA algorithm is much less affected by Web Local Aggregation. It is reasonable that out strategy could not make significant success since the original ranking system of SALSA is strong enough to resist the bias of Web Local Aggregation.

Our discussion focuses on alleviating the over-ranking problem. As a complement, we present some snapshots of top ranked pages given by improved algorithms as a demonstration how high quality pages win in these ranking systems. Table 7 presents 15 top ranked pages using algorithm PRL2M3. It preserves some high quality pages in Table 1 but removes many low quality pages. The page ranked second is the portal page to search NYU web, which is reasonable getting such high rank. Table 8 presents 15 top ranked pages using HTM3. The quality is not as good as PRL2M3, but it makes significant progress compared with top HITS ranking given by Table 2. We have already known SALSA algorithm is insensitive to local aggregation on the scale of average rank. It is interesting to know whether its top ranking is better than our improved algorithms, especially PRL2M3. Table 9 gives top ranked pages in SALSA. We see the results of PRL2M3 is still better than SALSA based on human judgement. This shows SALSA algorithm does not win all the time.

Table 7: Top ranked pages in PRL2M3

Rank	URL
1	www.nyu.edu
2	www.nyu.edu/bin/phfnyu
3	www.nyu.edu/gsas
4	www.nyu.edu/prospects.nyu
5	www.nyu.edu/library/bobst
6	www.law.nyu.edu
7	www.nyu.edu/cas
8	www.law.nyu.edu/index.html
9	www.nyu.edu/presidential.installation
10	www.nyu.edu/athletics/varsity_teams.html
11	www.nyu.edu/parentsday
12	www.nyu.edu/msep
13	www.nyu.edu/students.nyu
14	www.nyu.edu/alumni.nyu
15	www.nyu.edu/parents.guide

5. CONCLUSIONS AND FUTURE WORK

In the stochastic models of web ranking systems, web pages are ranked via co-citation and competition with each other. These models are often biased by Web Local Aggregation and produce bad ranking distribution. Empirically, such bias can be alleviated when the ranking are guided by some hyperlink evaluations. We address the issue of query-independent hyperlink evaluations that can be inferred from the web graph structure. A futher work based on our framework is the study of using query-oriented hyperlink evaluations to improve web ranking. There are some related works in this area [13, 20, 3, 4, 6, 8, 17, 14], but most of their approaches are addressed for only a certain ranking system and not applicable to different ranking systems. Ding et al. [10] has built a unified mathematical framework for PageRank, HITS and SALSA algorithms.

Table 8: Top ranked authority pages in HTM3

Rank	URL
1	www.nyu.edu
2	www.nyu.edu/bin/phfnyu
3	www.nyu.edu/gsas
4	www.med.nyu.edu/ethics.html
5	www.law.nyu.edu
6	www.nyu.edu/library/bobst
7	www.med.nyu.edu/som/departments.html
8	www.nyu.edu/cas
9	www.nyu.edu/athletics/varsity_teams.html
10	www.stern.nyu.edu/acc/about
11	rmm-java.stern.nyu.edu/jmis/toppage/index.html
12	www.med.nyu.edu/webmastermail.html
13	www.cims.nyu.edu
14	www.law.nyu.edu/sitemap
15	www.nyu.edu/its

Table 9: Top ranked authority pages in SALSA

Rank	URL
1	www.nyu.edu
2	www.nyu.edu/bin/phfnyu
3	www.med.nyu.edu
4	www.law.nyu.edu
5	www.nyu.edu/gsas
6	www.stern.nyu.edu/acc/about
7	rmm-java.stern.nyu.edu/jmis/toppage/index.html
8	www.nyu.edu/library/bobst
9	www.law.nyu.edu/sitemap
10	mcrcr2.med.nyu.edu:8765/medind.html
11	www.med.nyu.edu/disclaimer.html
12	www.med.nyu.edu/ethics.html
13	www.med.nyu.edu/facstud.html
14	www.nyu.edu/cas
15	www.nyu.edu/athletics/varsity_teams.html

This encourages people to use the framework of our approach to target the issue in a more general way.

6. ACKNOWLEDGEMENTS

I would like to thank Professor Ernest Davis for invaluable comments and feedback. He also improved the write-up of this paper a lot. I would also like to thank many anonymous reviewers for providing invaluable suggestions on this work.

7. REFERENCES

[1] R. Albert, H. Jeong, and A. Barabasi. Diameter of the world-wide-web. *Nature*, 401:130–131, 1999.

[2] A. Barabasi and R. Albert. Emergence of scaling in random networks. *Science*, 286:509–512, 1999.

[3] K. Bharat and M. R. Henzinger. Improved algorithms for topic distillation in a hyperlinked environment. In *Proc. 21th ACM SIGIR*, 1998.

[4] A. Borodin, G. Roberts, J. Rosenthal, and P. Tsaparas. Finding authorities and hubs from link structure on the world wide web. In *Proc. 10th World Wide Web Conference*, 2001.

[5] A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, and J. Wiener. Graph structure in the web. *Computer Networks and ISDN Systems*, 30:309–320, 2000.

[6] S. Chakrabarti, B. Dom, D. Gibson, J. Kleinberg, P. Raghavan, and S. Rajagopalan. Automatic resource compilation by analyzing hyperlink structure and associated text. In *Proc. 7th World Wide Web Conference*, 1998.

[7] D. Chon and H. Chang. Learning to probabilistically identify authoritive documents. In *Proc. 17th International Conference on Machine Learning*, 2000.

[8] M. Diligenti, M. Gori, and M. Maggini. Web page scoring systems for horizontal and vertical search. In *Proc. 11th World Wide Web Conference*, 2002.

[9] S. Dill, R. Kumar, and K. McCurley. Self-similarity in the web. In *International Conference on Very Large Data Bases*, 2001.

[10] C. Ding, X. He, P. Husbands, H. Zha, and H. Simon. Pagerank, hits and a unified framework for link analysis. In *Proc. 25th ACM SIGIR*, 2002.

[11] C. Dwork, S. Kumar, M. Naor, and D. Sivakumar. Rank aggregation methods for the web. In *Proc. 10th International World Wide Web Conference*, 2001.

[12] http://www.google.com.

[13] T. Haveliwala. Topic-sensitive pagerank. In *Proc. 11th World Wide Web Conference*, 2002.

[14] H. Kao, S. Lin, J. Ho, and M. Chen. Entropy-based link analysis for mining web informative structures. In *Proc. 11th ACM CIKM*, 2002.

[15] L. Kleinberg. Authoritative sources in a hyperlinked environment. In *Proc. 9th ACM-SIAM Symposium on Discrete Algorithms*, 1998.

[16] R. Lempel and S. Moran. The stochastic approach for link-structure analysis (salsa) and the tkc effect. In *Proc. 9th International World Wide Web Conference*, 2000.

[17] L. Li, Y. Shang, and W. Zhang. Improvement of hits-based algorithms on web documents. In *Proc. 11th World Wide Web Conference*, 2002.

[18] A. Ng, A. Zheng, and M. Jordan. Stable algorithms for link analysis. In *Proc. ACM SIGIR*, 2001.

[19] L. Page, S. Brin, R. Motowani, and T. Winograd. The pagerank citation ranking: bringing order to the web. *Stanford Digital Library working paper*, 1997-0072, 1997.

[20] M. Richardson and P. Domingos. The intelligent surfer: probabilistic combination of link and content information in pagerank. In *Proc. Advances in Neural Information Processing Systems*, 2002.

[21] M. Steinbach. A comparison of document clustering techniques. In *Proc. 6th SIGKDD*, 2002.