The Chinese University of Hong Kong
Department of Computer Science and Engineering

Ph.D. -- Term Paper

Cover Sheet

Title: Extending Link-based Algorithms for Similar Web Pages with Neighborhood Structure

Name: LIN Zhenjiang

Student I.D.: 05267230

Contact Tel. No.: 64157520 Email A/C: zjlin@cse.cuhk.

Supervisor(s): Michael R. Lyu, Irwin King

Marker(s): Lai-Wan Chan, Christopher C. Yang

Mode of Study: ☐ Part-time ☑ Full-time

Submission Date: 6 Dec 2006

Term: 2006-2007

Field(s): Computer Science and Engineering

Presentation Date:

Time:

Venue:

Abstract

The problem of finding similar pages to a given web page arises in many web applications such as search engine and web document classification. In this paper, we focus on the link-based similarity measures which compute web page similarity solely from the hyperlinks of the Web. We first propose a simple but important model called the Extended Neighborhood Structure (ENS), which defines a *bi-directional* (in-link and out-link) and *multi-hop* neighborhood structure. Based on the ENS model, several existing similarity measures are extended, which include PageSim, SimRank, Co-citation, and Bibliographic coupling. Moreover, theoretical analyses show that the extended PageSim is an online, incremental, scalable and stable algorithm, which is especially suitable for the Web. We test the algorithms on two datasets: a web graph crawled from the website of our department and a citation graph crawled from Google Scholar. Experimental results show that the performance of the extended algorithms is significantly improved and the extended PageSim outperforms all the others in our tests.

Keywords: similarity measure, web mining, link analysis, PageRank

1 Introduction

The World Wide Web (WWW, or simply "the Web") has grown into a giant warehouse with tremendous amount of information available online in the past two decades. Along with this growth, a wide variety of tools and applications have been or are being developed to extract valuable knowledge from the Web. One of the most famous mining tools is the Search Engine, which typically searches web pages related to the query keywords provided by users.

Unlike the *keyword searching* above, *instance searching* searches by instance rather than by keywords. That is, it takes a web page as the input and returns a list of related (or similar) web pages to this page. For example, for a query such as "www.cnn.com", the searching result would be such web pages related to news as "www.usnews.com" or "news.bbc.co.uk". One advantage of instance searching is that users can find the related web pages to the web page they are interested in, without having to worry about selecting the right keywords.

The problem of finding similar pages to a given web page arises in many web applications. One example is the "similar pages" service of Google. Each time users click on the 'Similar Pages' link for a search result (the URL of a web page), Google automatically searches the Web for pages that are related to this result. Web document classification (such as Yahoo! Directory) categorizes web pages into a hierarchical structure according to the degree of similarity between web pages. Web community identification is another important web application. One approach to identifying *web community*, which is a collection of web pages sharing a common topic [8, 26], is based on the similarity between web pages.

In the fields of information retrieval (IR) and recommender systems, the problem of finding similar objects has been studied extensively for many years, and a variety of domain-specific similarity measures have been developed. In traditional IR, text-based similarity measures based on *matching text* may be employed to find similar documents to a given query in a document corpus. For collaborative filtering in a recommender system [9, 17], similar users may be grouped by *users' preferences*. In particular, several link-based algorithms have been suggested to exploit the similarity information hidden in the link structure of graph, such as Co-citation [31], Companion [7], SimRank [14], and PageSim [23]. Further methods arise from graph theory, such as similarity measure that is based on network flows [24].

One major problem of the link-based algorithms is that they usually do not make full use of the structural information of the graph. For example, Co-citation only considers direct neighbors. SimRank is a multi-hop algorithm, but it considers only one direction. We believe that a well-designed algorithm should take into account as much link information as possible to produce high quality results.

On the other hand, we have to develop more specific algorithms for the Web, because traditional IR techniques are prevented from being applied to the Web directly by the following special characteristics of the Web:

1. **Huge**: The Web is probably the largest database in history. Studies have estimated that the volume of indexable web pages exceeded two billion at the end of last century [3, 20]. Recently, the authors of [10] estimated the size of the indexable Web to at least 11.5 billion pages as of the end of January 2005.

2. **High Dynamics**: Unlike books in library, web pages continue to

change even after they are initially created and indexed by search engines [5]. In [29], the authors suggested basically there are two dimensions of web dynamics: *growth dynamics* which indicates that the Web grows in size, and *update dynamics* which indicates that both the content and the link structure of the Web are constantly changed.

3. **Fast Growing**: The rapid expansion of the Web is another issue. Studies revealed that the Web grows at an exponential rate [20, 29]. The growth rate has been estimated to be roughly one million pages per day [20].

4. **Untrustworthy**: The Web is an untrustworthy world due to the fact that its contents, including textual content of web pages and hyperlinks between web pages, are prone to be manipulated, or *spammed*. *Spammers* on the Web use various techniques to "mislead search engines and give some pages higher ranking than they deserve" [11]. This action is called *web spamming* [11]. Some experts consider web spamming the single most difficult challenge web searching is facing today [12].

Motivation and Contributions: To develop efficient and flexible similarity measures which make full use of the structural information of the Web to produce high quality results motivates our research work. In this paper, we focus on the link-based similarity measures which compute similarity scores between web pages solely from the hyperlink structure modeled by the web graph, with vertices representing web pages and directed edges representing hyperlinks between pages. The main contributions of this paper are as follows.

1. A simple but important model called the Extended Neighborhood Structure (ENS) is proposed. This model defines a *bi-directional* (in-link and out-link) and *multi-hop* neighborhood structure. This model is designed for helping link-based algorithms make full use of the link information of a graph.

2. Several similarity measures are extended based on the ENS model. The performance of the extended algorithms improves significantly, which illustrates the effectiveness of the ENS.

3. The extended PageSim is an online, incremental, scalable, and stable algorithm which is especially suitable for the Web.

The rest of the paper is organized as follows. In Section 2, we present the related work on similarity measures and link-based algorithms. In Section 3, the ENS model is introduced and several link-based similarity measures are extended based on this model. Section 4 gives more details on the analysis of the extended PageSim algorithm. The experimental results on two datasets as well as discussions are shown in Section 5. Finally, we conclude our work and propose directions of future work in Section 6.

2 Related Work

A variety of text-based similarity measures have been proposed in the field of information retrieval, such as the *cosine similarity* and the *TFIDF* (Term Frequency-Inverse Document Frequency) model [30]. A problem of the text-based methods is that generally they require large storage and long computing time due to the need of full-text comparison. Moreover, they are prone to be manipulated by *keyword spamming*. These limitations prevent pure text-based similarity measures from being applied to the *huge* and *untrustworthy* Web directly.

The link-based similarity functions were first proposed in the field of bibliometrics, which studies the citation patterns of scientific articles, and infers relationships between articles from their cross-citations [16, 32]. Two notable algorithms are *Co-citation* and *bibliographic coupling*. The Co-citation algorithm measures similarity between two articles based on the number of articles which cite both of them. In the bibliographic coupling algorithm, similarity is based on the number of articles cited by both of the two articles.

A number of link-based similarity measures have been proposed in the past few years. The *Maximum Flow/Minimum Cut* and *Authority* algorithms were developed for measuring the similarity of scientific papers in a citation graph [24]. The SimRank algorithm was proposed to measure similarity of the structural context "in any domain with object-to-object relationships" [14]. It is a recursive refinement of co-citation based on the assumption that "two objects are similar if they are referenced by similar objects". *Jaccard measure* [13] and *Adamic/Ada* [2] were also applied to the link prediction problem underlying social network evolution using only link information in [22]. We refer to the article [22], which contains an exhaustive list of link-based similarity measures.

Web structure mining has been largely influenced by research in the fields

of social network and citation graph. In recent years, Web link structure has been widely used to exploit important information inherent in the Web. One successful link-based algorithm is *PageRank* [28], which is the "heart" of Google search engine. PageRank assigns a global "importance" or "authority" score to each web page solely based on the structural information of the Web. The intuition behind PageRank is "a page has high rank if the sum of the ranks of its backlinks (in-links) is high" [28]. As reported in [25], web spamming seems to be the driving force behind the evolution of search engines in their effort to provide quality results. The success of PageRank is mainly based on the more sophisticated anti-spamming solution it provided [25].

Unlike PageRank, *HITS*, another link-based algorithm, computes two different scores for each web page: a hub score and an authority score. A page with a high authority score is one linked to by many good hubs, and a page with a high hub score is one that links to many good authorities. The authority and hub scores are mutually reinforced, and they can be computed recursively. Following the success of the PageRank and HITS, other link-based algorithms have also been developed, such as *SALSA* [21], *pSALSA* [4], and *PHITS* (Probabilistic HITS) [6], etc.

Recently, the link-based similarity measures have been suggested over the web graph. However, as we mentioned before, a link-based similarity measure has to be designed carefully to fit for the special characteristics of the Web.

3 Extending Similarity Measures

We believe that, to produce high quality results, a well-designed link-based algorithm should make full use of the structural information of the web graph. However, almost all existing similarity measures are either single-directional or just 1-hop, which limits their performance.

In this section, we first propose the Extended Neighborhood Structure (ENS) model which defines a generalized neighborhood structure on graph. Based this model, several existing similarity measures are extended. Experimental results in Section 5 show that the extended algorithms outperform the original ones, which serve to illustrate the effectiveness of this model. Moreover, the extended PageSim algorithm introduced later is designed according to the intuition in Section 3.1, and it performs the best among all the algorithms tested in this paper.

3.1 The Extended Neighborhood Structure Model

Recent research has suggested that there are large amounts of valuable information hidden in the vast link structure of the Web. For example, a web page linking to another page usually implies some kind of relationship between them. This is because the fact that generally authors of web pages would like to link their pages to those pages which they think are related to theirs.

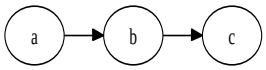


Figure 1: Interpretation of the ENS model

Consider the graph in Figure 1, why does page a link to page b? Maybe the reason is "a is interested in b", or "a is familiar with b", or else. No matter what the reason is, the basic fact is that at least a knows b. Of course, b may not know a since there's no hyperlink from b to a. It is very much like the relationship between people. Therefore, a web page may have two kinds of neighbors: *in-link neighbors* (those who know it) and *out-link neighbors* (whom it knows). In Figure 1, a is b's in-link neighbor and b is a's out-link neighbor. Now, we can come up with a simple and straightforward intuition on web page similarity: *similar web pages have similar neighbors*. Or in other words, to know a web page, know its neighbors!

On the other hand, page c is a 2-hop indirect out-link neighbor of a, which implies page a may not be so familiar with c as with b. This assumption is reasonable and can be thought as the familiarity decreases along links (both in-links and out-links).

Therefore, the concept of neighborhood is now extended in two aspects: *bi-direction* and *multi-hop*. Although the intuition of similarity is still "similar web pages have similar neighbors", its meaning is generalized, since the "neighbors" here refer to the bi-directional and multi-hop neighbors instead of single-direction or direct neighbors. This model is based on the natural hypothesis that a link-based algorithm likely improves its accuracy by considering more structural information of the graph. In the following experiments, we'll show that the ENS model really works.

In the following of this paper, the notation $Sim(a, b)$ represents the similarity score of web pages a and b, which is produced by the similarity mea-

sure in the context. Now we define the web graph model, which will be used throughout this paper.

3.2 Web Graph Model

We model the Web as a directed graph $G = (V, E)$ with vertices V representing web pages $v_i (i = 1, 2, \cdots, n)$ and directed edges E representing hyperlinks among the web pages. $I(v)$ denotes the set of in-link neighbors of pages v and $O(v)$ denotes the set of out-link neighbors of page v.

Definition 1 *Let $path(u_1, u_s)$ denote a sequence of vertices $u_1, u_2, \ldots, u_s$ such that $(u_i, u_{i+1}) \in E$ $(i = 1, \cdots, s-1)$ and u_i are distinct. It is called a **path** from u_1 to u_s.*

Definition 2 *Let $length(p)$ denote the **length** of path p, and define $length(p) = |p| - 1$, where $|p|$ is the number of vertices in path p.*

Definition 3 *Let $PATH(u, v)$ denote the set of all possible paths from page u to v.*

3.3 Extended Co-citation and Bibliographic Coupling

Co-citation and bibliographic coupling are two 1-hop and single-directional algorithms. Their intuitions and definitions are as follows.

1. **Co-citation:** the more common in-link neighbors two pages have, the more similar they are. Therefore, $Sim(a, b) = |I(a) \cap I(b)|$.
2. **Bibliographic coupling:** the more common out-link neighbors two pages have, the more similar they are. Therefore, $Sim(a, b) = |O(a) \cap O(b)|$.

We can easily construct a bi-directional algorithm called Extended Co-citation and Bibliographic Coupling (ECBC) as follows.

1. **ECBC:** the more common neighbors two pages have, the more similar they are. Therefore,

$$Sim(a, b) = \alpha |I(a) \cap I(b)| + (1 - \alpha) |O(a) \cap O(b)|,$$

where $\alpha \in [0, 1]$ is a user-defined constant.

3.4 Extended SimRank

SimRank is a fixed point of the recursive definition: *two pages are similar if they are referenced by similar pages.* Numerically, for any web page u and v, this is specified by defining $Sim(u, u) = 1$ and

$$Sim(u, v) = \gamma \cdot \frac{\sum_{a \in I(u)} \sum_{b \in I(v)} Sim(a, b)}{|I(u)||I(v)|}$$

for $u \neq v$ and $\gamma \in (0, 1)$. If $I(u)$ or $I(v)$ is empty, then $Sim(u, v)$ is zero by definition. The SimRank iteration starts with $Sim_0(u, v) = 1$ for $u = v$ and $Sim_0(u, v) = 0$ for $u \neq v$. The SimRank score between u and v is defined as $lim_{k \to \infty} Sim_k(u, v)$.

SimRank is a multi-hop algorithm, but it is not bi-directional. We extend the intuition of SimRank to be "two pages are similar if they have similar neighbors". Accordingly, SimRank can be extended to

$$Sim(u, v) = \gamma \cdot (\sum_{a \in I(u)} \sum_{b \in I(v)} Sim(a, b) + \sum_{a \in O(u)} \sum_{b \in O(v)} Sim(a, b))$$

$$\times (|I(u)||I(v)| + |O(u)||O(v)|)^{-1}.$$

The proof of the convergence of the extended SimRank is omitted here.

3.5 Extended PageSim

PageSim can be regarded as a "weighted multi-hop" version of Co-citation algorithm. First, it takes the common in-link information of 1-hop as well as multi-hop neighbors into account to improve the quality of the result. Moreover, since not all pages are equally important on the Web, it is possible that a citation of an authoritative web page may be more important than that of several obscure pages. Therefore, PageSim also considers the importance of web pages.

The PageSim algorithm can be simply described as: each web page propagates its "feature information" to its (multi-hop) out-link neighbors along hyperlinks. Once the propagation of all web pages finished, the similarity between two pages is measured by the "feature information" that they have in common.

In PageSim, a web page linking to others is considered as it introducing itself to them. Its out-link neighbors then propagate its self-introduction

to their own out-link neighbors. Naturally, the introduction information decreases along with the propagation. As a result, two pages are similar if they received common introductions (they also have their own introduction). By this way, PageSim successfully avoid the situations such as two linking-to-each-other pages are decided to be not similar if they are not introduced by intermediate pages in SimRank.

For a web page, the volume of its "feature information" is measured by its PageRank score or other score that represents the importance of this page. Obviously, each web page has to preserve a space, which is called the *feature vector*, to store the "feature information" of its own as well as all the others. The mathematical definitions of PageSim is given as follows.

Definition 4 *Let $PR(v)$ denote the PR score of page v. Let $PG(u,v)$ denote the PR score of page u that propagates to v through $PATH(u,v)$. We define*

$$PG(u,v) = \begin{cases} \sum_{p \in PATH(u,v)} \frac{d \cdot PR(u)}{\prod_{w \in p, w \neq v} |O(w)|} & v \neq u, \\ \\ PR(u) & v = u, \end{cases} \tag{1}$$

where $d \in (0,1]$ is a decay factor and $u, v \in V$.

Definition 5 *Let $\overrightarrow{FV}(v)$ denote the Feature Vector of page v.*

$$\overrightarrow{FV}(v) = (PG(v_i, v))^T = (PG_i(v))^T, i = 1, \cdots, n,$$

where $v, v_i \in V$.

Definition 6 *Let $PS(u,v)$ denote the PageSim score of pages u and page v. We define*

$$PS(u,v) = \sum_{i=1}^{n} \frac{min(PG_i(u), PG_i(v))^2}{max(PG_i(u), PG_i(v))}, \tag{2}$$

where $u, v \in V$.

The detailed explanations of the above definitions are given in [23]. In short, there are two stages in the PageSim algorithm: *PR score propagation stage* and *PS score computation stage*. Equations (1) and (2) correspond to the processes in these two stages respectively.

Extended PageSim (EPS): In PageSim, the "feature information" of web pages propagate along only out-links, and the consequent PS scores are

actually "out-link" PS scores. In EPS, we also propagate along in-links (with decay factor $1 - d$) and produce the "in-link" PS scores. This is because we consider the in-links complement to out-links. Considering the in-link propagation may help increase the quality of searching results. The EPS score of two pages is hence defined by the sum of "in-link" and "out-link" PS scores of them. We denote the EPS score of pages u and v by $EPS(u, v)$. Certainly, the storage requirement of EPS is doubled since we also need to store the "feature information" propagated through in-links.

Moreover, we adopt the Jaccard measure [13], which is commonly used in IR to measure the similarity between two vectors, to calculate the similarity scores of the Extended PageSim (EPS). For example, to calculate the "out-link" PS score of u and v, we use Equation (3) instead of Equation (2).

$$PS(u, v) = \frac{\sum_{i=1}^{n} min(PG(v_i, u), PG(v_i, v))}{\sum_{i=1}^{n} max(PG(v_i, u), PG(v_i, v))}, \tag{3}$$

where $u, v \in V$.

In EPS algorithms, the PR score propagation process of a web page is encapsulated in the *PR_prop* sub-function, and the calculation of EPS score between two pages is in the *PS_calc* sub-function. Since these sub-functions are rather straightforward, we omit them to make the paper tidy.

Algorithm 1 Extended PageSim (EPS) Algorithm

```
1: Input: G: web graph G(V, E) with known PageRank scores.
2: Output: EPS_{n×n}: EPS score matrix (EPS(v_i, v_j))_{n×n}.
3: procedure EPS(G)
4:     // Stage 1: PageRank score propagation
5:     for i ← 1, n do
6:         call PR_prop(G,v_i)                    ▷ propagate PR score of v_i
7:     end for
8:     // Stage 2: PageSim score calculation
9:     for i ← 1, n do
10:        for j ← 1, n do
11:            call PS_calc(v_i,v_j)              ▷ calculate EPS(v_i,v_j)
12:        end for
13:    end for
14:    return EPS score matrix
15: end procedure
```

3.6 Case Study and Summary

We have extended several link-based similarity measures based on the ENS model. In this part, we give some simple cases to illustrate a major limitation of the original algorithms. That is, they may produce incorrect result on web page similarity in some common situations. We summarize these algorithms as well as their extended versions at the end of this part.

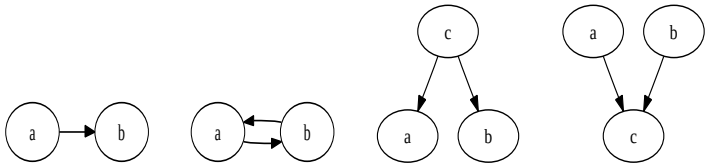


Figure 2: Case Study

In Figure 2, we list four kinds of common relationship between pages. In each of them, we know that a and b are related pages, therefore $Sim(a, b) \neq 0$. However, some algorithms incorrectly calculate that $Sim(a, b) = 0$ which means a and b are unrelated. We list the results on the relationship between a and b produced by the algorithms in Table 2, with "+" representing a is related to b and "-" otherwise. The properties of each algorithm are listed in Table 3, with "-" representing "NO" and "+" representing "YES". The algorithms include Co-citation (CC), Bibliographic coupling (BC), Extended Co-citation and Bibliographic Coupling ($ECBC$), SimRank (SR), Extended SimRank (ESR), PageSim (PS), and Extended PageSim (EPS).

Table 2 shows that: (a) the extended algorithms can measure more cases than the original ones; (b) only EPS can measure $Sim(a, b)$ correctly in all cases, since it takes both bi-direction and multi-hop structural information into account, as shown in Table 3.

In Table 3, ESR is also a multi-hop and bi-directional algorithm. However, it decide how similar two pages are by their "common similar neighbors" only. That is, to be similar pages, two pages have to have similar neighbors, or they have to be introduced to each other by intermediate pages. If they don't, they are not similar even they know each other (linking to each other as shown in Figure 2(2)). While in EPS, the introducers are not necessary since each page can introduce itself by propagating its "feature information".

Table 1: Sim(a, b)

Case	CC	BC	SR	PS
1	-	-	-	+
2	-	-	-	+
3	+	-	+	+
4	-	+	-	-
5	-	-	-	+

Table 2: Simple Case Study

Case	CC	BC	ECBC	SR	ESR	PS	EPS
1	-	-	-	-	-	+	+
2	-	-	-	-	-	+	+
3	+	-	+	+	+	+	+
4	-	+	+	-	+	-	+

Table 3: Properties of the Algorithms

Properties	CC	BC	ECBC	SR	ESR	PS	EPS
bi-direction	-	-	+	-	+	-	+
multi-hop	-	-	-	+	+	+	+

4 Analysis of Extended PageSim

Although, from the definitions in Section 3.5 we can see that the major difference between Extended PageSim (EPS) and PageSim is that EPS is bi-directional whereas PageSim is not, their performance is even more different. Experimental results show that EPS outperforms PageSim significantly. Actually, the performance results of extended algorithms are all better than those of original ones in our experiments. All of these results serve to show that the Extended Neighborhood Structure model really works.

In this section, we look more insight into the EPS. We first review the pruning technique that we employed in [23]. After complexity analysis, we show the properties of the EPS algorithm.

4.1 The Pruning Technique

The pruning technique is based on the observation that the volume of information propagated in distance usually drops quickly. Therefore, by pruning the radius of propagation, we may improve the efficiency of algorithm without reducing its precision significantly. It is actually a tradeoff between efficiency and precision. This technique can be applied in most multi-hop algorithms, such as SimRank and PageSim.

4.2 Complexity Analysis

The EPS can certainly adopt pruning technique too. Suppose the average number of one web page's neighbors is $k = (\sum_{i=1}^{n} |I(v_i)| + |O(v_i)|)/n$ and the radius of propagation is $r \in \mathcal{N}$. The time complexity of *PS_ prop* is hence $O(C)$, where $C = k^r$ is constant with respect to n.

The space complexity also benefits from pruning technique. Although the two feature vectors of a web page is designed to store PR scores of all web pages, the size of them should be far less than n. Because on the huge Web, it is unlikely that a web page receives PR scores of all the pages, especially when the radius of propagation is "pruned". It is easy to conclude that the expectation of one feature vector's size is also $O(C)$. As a result, the time complexity of *PS_calc* function is $O(C)$ too.

Therefore, by adopting the pruning technique, the space complexity of EPS, the time complexity of propagating all of n web pages's PR scores, and

the time complexity of computing all of the EPS scores related to a query page are all $O(Cn)$.

Apparently, the key factor of the complexities of EPS is the propagation radius r, since large r results in huge C which may dramatically increase the running time of the algorithm. Therefore, finding a small r while preserving the precision is an important task. The experiments conducted in Section 5 show that $r = 3$ is such an empirical propagation radius. On the other hand, the average number of in-links per web page was measured at about 8 [19], and the average number of out-links per web page was measured at about 7.2 [18]. Therefore, we have $C = k^r \approx 16^3 = 4096$. Considering the huge n, $C \approx 4096$ indeed indicates that EPS is efficient in both time and storage.

4.3 Characteristics of the EPS

Based on the complexity analysis, next we analyze the following special characteristics of EPS. We show that EPS is an online, incremental, scalable, and stable algorithm, which is especially suitable for the Web.

Online: We know that there are two stages in EPS: propagation stage and calculation stage. When the PR score propagation finishes, the EPS score of any two pages can be calculated by calling *PS_calc* based on their feature vectors only. However, no matter what the complexity of *PS_calc* is, calculating the EPS scores of all n^2 page-pairs of the Web is really a tough task.

Fortunately, based on the assumption of propagation radius pruning, precomputing all the n^2 EPS scores can be avoided. As we know, a web page only stores $O(C)$ web pages' PR scores. On the other hand, a web pages's PR score can only be propagated to at most $O(C)$ pages. Therefore, the number of pages which may contain common PR scores with a query page is at most $O(C) \cdot O(C) = O(C^2)$,which is also the time complexity of finding all of these pages. Obviously, we just need to calculate the EPS scores between these $O(C^2)$ pages and the query page, since only them are possibly similar to the query page. We have known that $C \approx 4096$. Given a query page, it is possible to compute the corresponding $O(C^2)$ EPS scores *online* with a powerful computing environment. As a result, huge amounts of storage space is saved.

Incremental: The Web is highly dynamic, which means there are large numbers of perturbations (or changes) on the web graph. We refer "changes" to web pages or hyperlinks added or removed from the Web. Due to the huge

volume of the Web, incremental algorithms are surely preferred to update the results of algorithm efficiently.

In EPS, since the PR score propagation of each page is independent, we just need to re-propagate those pages which are "influenced" by perturbations. Clearly the change of a hyperlink will influence the two pages it links. For each of the two pages, its structural change will consequently influence $O(C)$ pages which propagate their feature information to it. So the change of a hyperlink will actually influence $2 + 2 \cdot O(C) = O(C)$ pages. Consequently, the change of a page a will influence about $(I(a) + O(a))O(C)$ web pages.

We can see that, in EPS, the influenced pages by a perturbation are just a small portion of the huge Web. Re-propagating only the influenced pages will certainly improve the update efficiency.

Scalable: First, we know that the time complexity of *online* EPS is linear with respect to n, given a fixed or bounded r. So the *online* EPS is efficient. Second, EPS inherits parallelism property, because the feature information of each page is propagated *independently*. This property is very important since EPS can be implemented to utilize the computing power and storage capacity of tens to thousands of computers interconnected with a fast local network. These two properties are essential for EPS to be capable of handling the huge and fast growing Web.

Stable: The stability of EPS is based on two aspects: the stability of PageRank and the "localism" of EPS. First, in [27], the authors proved that the perturbed PR scores will not be far from the original as long as the perturbed web pages did not have high overall PR scores. This means that PageRank scores are fairly stable since web pages which have high PR scores are only a tiny part of the Web. Second, due to the pruning technique, web pages only propagate PR scores to their nearby neighbors, which means a small change of the Web only influences on the feature vectors of nearby web pages. Based on these two facts, which can be concluded as "*propagating stable PR scores locally*", the EPS is a stable algorithm.

Spamming Resistance: During the past few years, web spamming has became such a big problem that spamming-resistant algorithms are certainly preferred. In [23], the ability of spamming resistance of PageSim has been analyzed. The analysis is also applicable to the extended PageSim. However, we would like to leave the complicated spamming issue for our future work.

We list some other properties of EPS below, which can be easily deduced from the definitions in Section 4. For any web pages u and v,

1. The EPS scores are *symmetric*, i.e., $EPS(u, v) = EPS(v, u)$;

2. Each page is most similar to itself, i.e., $EPS(u, u) = \max_{v \in V} EPS(u, v)$;

3. $EPS(u, v) \in [0, 1]$.

5 Experimental Results

We have proposed the Extended Neighborhood Structure (ENS) model and extended several link-based similarity measures, including Co-citation, bibliographic coupling, SimRank, and PageSim, based on this model. In this section, we report on some preliminary experimental results. The primary purpose is to show that the ENS model indeed helps link-based similarity measures improve their accuracy. Moreover, since the Extended PageSim (EPS) is one of the focus in this paper, we conducted more tests on it. The tests include estimating the empirical value of propagation radius r and testing the effect of the decay factor d on the result of EPS.

5.1 Datasets

We tested the algorithms on two types of graphs: one is a web graph crawled from our department, and the other is a citation graph crawled from Google Scholar [1]. All text in our datasets are in English.

1. **CSE Web (CW) dataset** is a set of web pages crawled from the web site of CSE department at CUHK (*http://www.cse.cuhk.edu.hk*), which contains about 22,000 web pages and 180,000 hyperlinks. The average numbers of in-links and out-links are 8.6 and 7.7 respectively.

2. **Google Scholar (GS) dataset** contains a citation graph of 20,000 articles which were crawled through public interface of Google Scholar search engine [1], with vertices representing articles and directed edges representing citations between articles (directed edge (u, v) exists if and only if article u cites v). To obtain this dataset, we first submitted keyword "web mining" to the Google Scholar which returned 50 related articles as a result. Then we crawled the remaining articles by following

the "Cited By" hyperlinks of the search results using Breadth-First Search algorithm. (The"Cited By" hyperlink of an article goes to a list of articles citing it.) What Google Scholar interests us is its "Related Articles" function which provides users a list of related articles to the original result. These "Related Articles" can be used as ground truth in the following experiments.

5.2 Ground Truth and Evaluation Methods

For any vertex v in graph G, a similarity measure A would produce a list of top N vertices most similar to v (excluding v itself), which is denoted by $top_{A,N}(v)$. Let the number $score_{A,N}(v)$ denote the average score to v of the $top_{A,N}(v)$. Thereby, we consider the average number of $score_{A,N}(v)$ for all $v \in V$ as the quality of the top N results produced by algorithm A, which is denoted by $\Delta(A, N)$. That is, $\Delta(A, N) = (\sum_{v \in V} score_{A,N}(v))/n$.

A good evaluation of the similarity measures is difficult without performing extensive user studies or having a reliable ground truth. In this paper, we would use two different evaluation methods. For the CW dataset, we use the cosine TFIDF, a traditional text-based similarity function, as rough metrics of similarity. For the GS dataset, we use the "Related Articles" provided by Google Scholar as ground truth.

(1) Cosine TFIDF Similarity: The cosine TFIDF similarity score of two web pages u and v is just the cosine of angle between TFIDF vectors of the pages [15], which is defined by

$$TFIDF(u, v) = \frac{\sum_{t \in u \cap v} W_{tu} \cdot W_{tv}}{\|u\| \cdot \|v\|},$$

where W_{tu} and W_{tu} are TFIDF weights of term t for web pages u and v respectively. $\|v\|$ denotes the length of page v, which is defined by $\|v\| = \sqrt{\sum_{t \in v} W_{tv}^2}$.

Therefore, for the CW dataset, we define

$$score_{A,N}(v) = \frac{1}{N} \sum_{u \in top_{A,N}(v)} TFIDF(u, v),$$

and $\Delta^T(A, N) = \Delta(A, N)$ which measures the average cosine TFIDF score of top N similar web pages returned by algorithm A.

(2) Related Articles: For an article v in citation graph G, the list of its "Related Articles" returned by Google Scholar is denoted by $RA(v)$. We define

$$related_N(v) = \{\texttt{top N related articles } v_i | v_i \in RA(v) \cap V\}.$$

The precision of similarity measure A at rank N is:

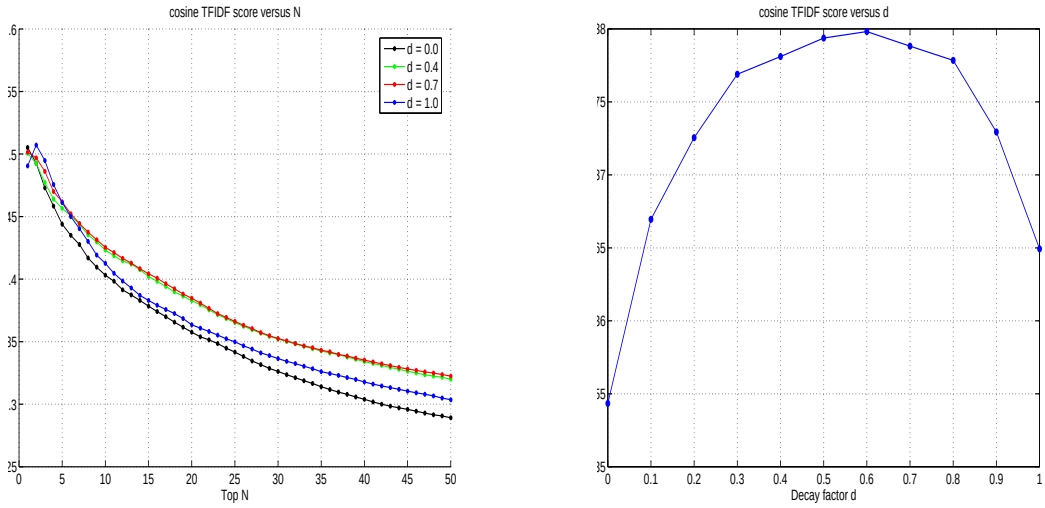
$$precision_{A,N}(v) = \frac{|top_{A,N}(v) \cap related_N(v)|}{|related_N(v)|}.$$

Therefore, for the GS dataset, we simply define

$$score_{A,N}(v) = precision_{A,N}(v),$$

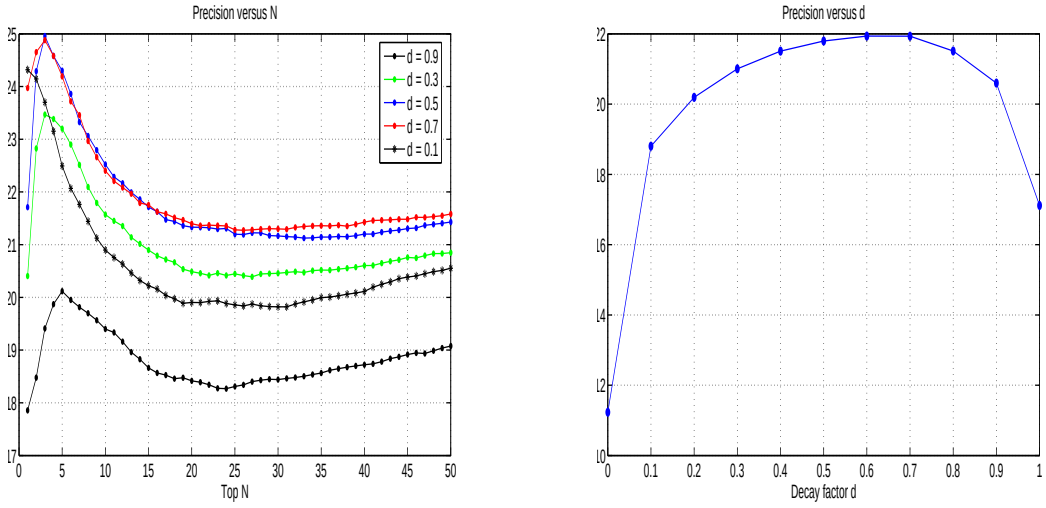
and $\Delta^P(A, N) = \Delta(A, N)$ which measures the average precision of algorithm A at top N.

5.3 Results on the Decay Factor of EPS



(a) (b)

Figure 3: Estimation of the optimal decay factor d on CW dataset



(a) (b)

Figure 4: Estimation of the optimal decay factor d on GS dataset

Without adopting pruning technique, we experimented with various values for the decay factor d of EPS. We found that $d \approx 0.6$ to be the best setting for our datasets. The results of CW and GS datasets are shown in Figure 3 and Figure 4 respectively.

Figure 3(a) plots the curves of $\Delta^T(A, N)$ for different d. Figure 3(b) shows the "overall performance" of EPS for different d, in which the average values of the curves in Figure 3(a) are on the y-axis, and their corresponding decay factors are on the x-axis.

Figure 4(a) plots the curves of $\Delta^P(A, N)$ for different d. In Figure 4(b), the average values of the curves in Figure 4(a) are on the y-axis, and the corresponding decay factors are on the x-axis.

From the figures we can see that: (a) both figures showed the optimal value of d is around 0.6; (b) since $d = 1.0$ corresponds to the original PageSim, the results of EPS outperform those of the original PageSim.

5.4 Results on the Propagation Radius of EPS

We also test the effects of the propagation radius r on the results of EPS and get an empirical radius. The results of CW and GS datasets are shown in Figure 5 and 6 respectively. In these Figures, "$r = n$" means no radius

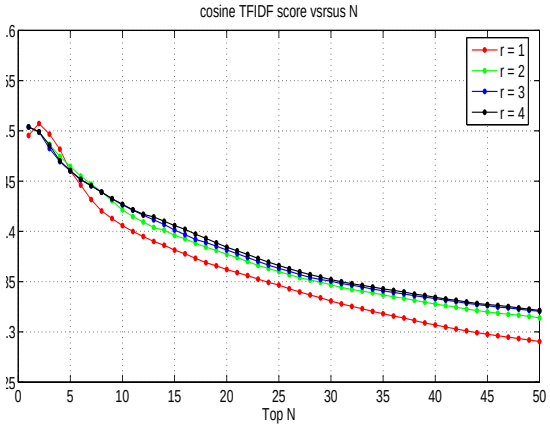


Figure 5: Empirical Radius r of CW dataset

pruning is applied to EPS.

Figure 5 plots the curves of $\Delta^T(A, N)$ for different r. First, it shows that the quality of the results increases with r. Second, the curve of $r = 3$ is very close to the "$r = n$" curve of EPS. Therefore, we can choose $r = 3$ to be the propagation radius in practice.

Figure 6 plots the curves of $\Delta^P(A, N)$ for different r. What it agrees with Figure 5 is that $r = 3$ is a good approximation. But what is more interesting is that the quality of the results *decreases* with r. We think that the possible reasons may be:

(1) The citation graph of GS dataset is incomplete. First, we crawled the articles along "inverse" citation direction. This means, for any article, we only know who cites it (its in-links), but we don't know all of its references (out-links). It is different from the Web, in which we usually only know the out-links of web pages. Second, the downloaded articles are only 1/4 to 1/3 of the articles found by crawler, which is similar to the Web.

(2) The Google Scholar search engine probably takes *direct* citation as more important. This is the most possible reason since the result of $r = 1$ is much better than others. We do not think Google Scholar is perfect, nevertheless, it is a useful tool to measure the *relative* performance of similarity functions.

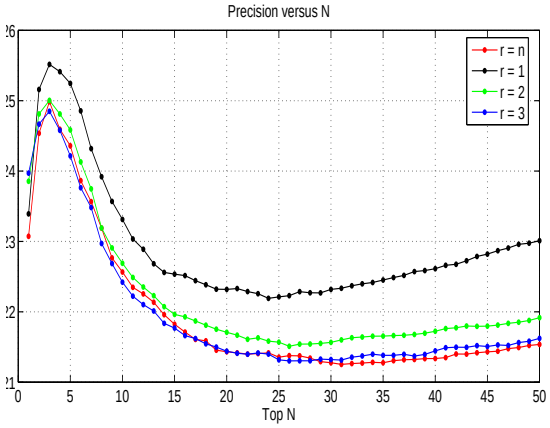


Figure 6: Empirical Radius r of GS dataset

5.5 Performance Evaluation of Algorithms

In this part, we evaluate the algorithms mentioned in this paper on the CW and GS datasets. These algorithms include Co-citation (CC), Bibliographic coupling (BC), Extended Co-citation and Bibliographic Coupling ($ECBC$), SimRank (SR), Extended SimRank (ESR), PageSim (PS), and Extended PageSim (EPS). The parameter settings of the algorithms are listed in Table 4.

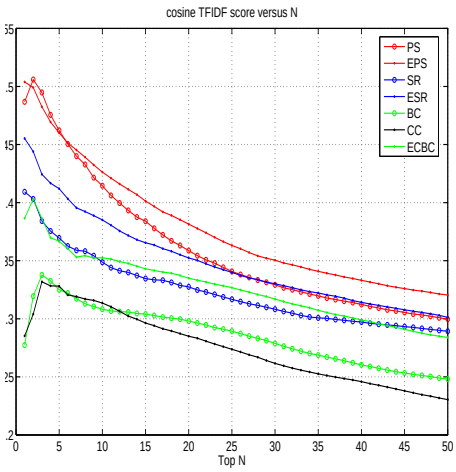
Table 4: Parameter Settings

ECBC	SR	ESR	PS	EPS
$\alpha = 0.5$	$C = 0.8$	$C = 0.8$	$r = 3$, $d = 0.5$	$r = 3$, $d = 0.6$

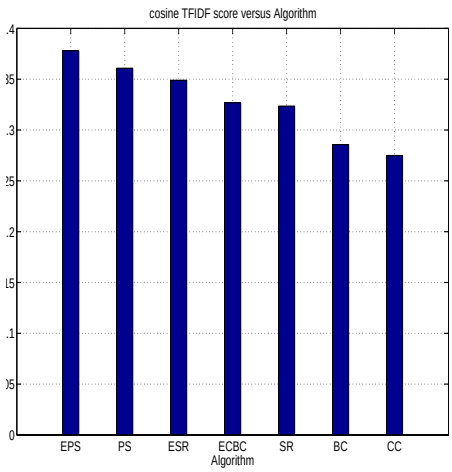
Figure 7(a) plots the curves of $\Delta^T(A, N)$ for different algorithms on the CW datasets. Figure 7(b) shows the the average values of the curves in Figure 7(a). Figure 8(a) plots the curves of $\Delta^P(A, N)$ for different algorithms on the GS datasets. Figure 8(b) shows the the average values of the curves in Figure 8(a). In Table 5, we list the performance improvement across all N for each extended algorithm.

From the results, we can see that:

1. The performance of the extended algorithms are significantly improved in almost all testing cases. This indicates that the ENS model really



(a) score curves



(b) overall performance

Figure 7: Performance of the algorithms on CW dataset

works.

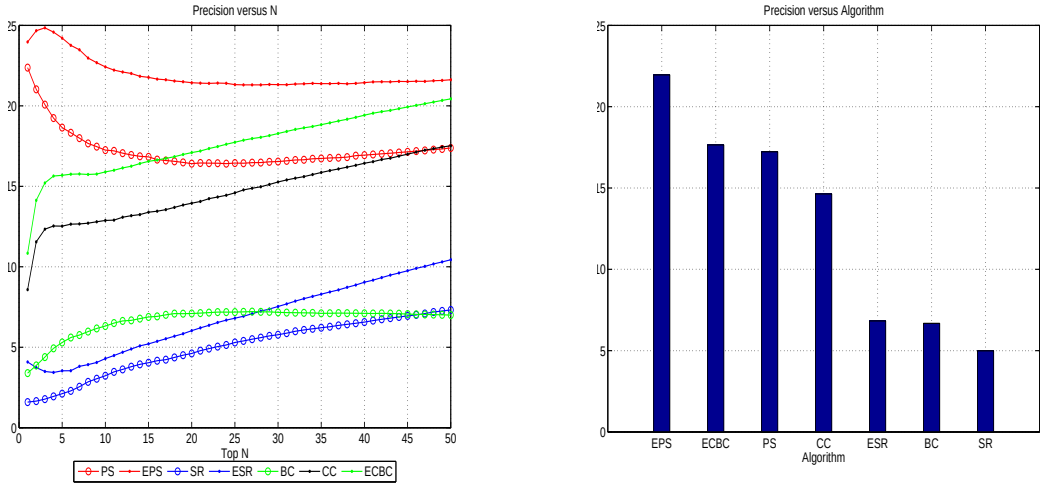
2. The EPS algorithm outperforms all the others in all test cases. It is because the EPS employs more structural information than others. It also confirms the effectiveness of the ENS model.

Table 5: Improvement of Overall Performance

Dataset	ECBC/CC	ECBC/BC	ESR/SR	EPS/PS
CW	18.87%	14.39%	7.86%	4.73%
GS	20.61%	164.21%	36.94%	27.44%

6 Conclusion and Future Work

Efficiently measuring similar between web pages is required in many web applications such as search engine and web document classification. In this paper, we focused on the link-based similarity measures which compute web page similarity solely form the hyperlinks of the Web. Our motivation is to develop efficient and flexible similarity measures which makes full use of



(a) score curves (b) overall performance

Figure 8: Performance of the algorithms on GS dataset

the structural information of the Web to produce high quality results. We propose the Extended Neighborhood Structure (ENS) model which defines a *bi-directional* (in-link and out-link) and *multi-hop* neighborhood structure. This model is designed for helping link-based algorithms make full use of the link information of a graph. Based on the ENS model, some existing similarity measures, including Co-citation, Bibliographic coupling, SimRank, and PageSim, are extended. In particular, the extended PageSim is an online, incremental, scalable, and stable algorithm which is especially suitable for the Web.

The algorithms and their extended versions were tested on two datasets: the CW web graph and GS citation graph. Experimental results show that the performance of the extended algorithms are significantly improved and the extended PageSim outperforms all the others in our tests.

There are a number of avenues for future work. Foremost, we need to extend more existing similarity measures based on the ENS model, and test the performance of the extended algorithms. We believe that by taking more link information into account, the performance of the link-based algorithms will be improved. Second, more extensive experiments are needed to evaluate the performance of the algorithms. This includes testing more datasets and comparing with more existing approaches. Third, spamming-resistance is

another direction for link-based algorithms, since web spamming is one of the most difficult challenges web searching is facing today [12]. Finally, we believe that a practical algorithm on the Web has to be hybrid, so integrating link-based similarity measures with other (text-based) methods is another direction of our future work.

7 Acknowledgment

This work is supported by grants from the Research Grants Councils of the HKSAR, China (Project No. CUHK4205/04E and Project No. CUHK4235/04E) and is affiliated with the VIEW Technologies Laboratory and the Microsoft-CUHK Joint Laboratory for Human-centric Computing & Interface Technologies.

References

[1] http://scholar.google.com/.

[2] L. Adamic and E. Adar. Friends and neighbors on the Web. *Social Networks*, 25(3):211–230, 2003.

[3] K. Bharat and A. Broder. Mirror, mirror on the Web: a study of host pairs with replicated content. In *WWW '99*, pages 1579–1590, NY, USA, 1999. Elsevier North-Holland, Inc.

[4] A. Borodin, G. O. Roberts, J. S. Rosenthal, and P. Tsaparas. Finding authorities and hubs from link structures on the World Wide Web. In *WWW '01: Proceedings of the 10th international conference on World Wide Web*, pages 415–429, NY, USA, 2001. ACM Press.

[5] B. Brewington, G. Cybenko, R. Stata, K. Bharat, and F. Maghoul. How dynamic is the Web? In *WWW '00: Proceedings of the 9th Conference on World Wide Web*, Amsterdam, Netherlands, May 2000. ACM Press.

[6] D. Cohn and H. Chang. Learning to probabilistically identify authoritative documents. In *ICML '00: Proceedings of the 17th International Conference on Machine Learning, Stanford, CA, USA*, 2000.

[7] J. Dean and M. R. Henzinger. Finding related pages in the World Wide Web. *Computer Networks (Amsterdam, Netherlands: 1999)*, 31(11–16):1467–1479, 1999.

[8] G. Flake, S. Lawrence, and C. L. Giles. Efficient identification of web communities. In *KDD '00: Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 150–160, Boston, MA, USA, 2000.

[9] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry. Using collaborative filtering to weave an information tapestry. *Commun. ACM*, 35(12):61–70, 1992.

[10] A. Gulli and A. Signorini. The indexable web is more than 11.5 billion pages. In *Special interest tracks and posters of the 14th international conference on World Wide Web*, pages 902–903, NY, USA, 2005. ACM Press.

[11] Z. Gyöngyi and H. Garcia-Molina. Web spam taxonomy. In *the First International Workshop on Adversarial Information Retrieval on the Web*, 2005.

[12] M. R. Henzinger, R. Motwani, and C. Silverstein. Challenges in web search engines. *SIGIR Forum*, 36(2):11–22, 2002.

[13] A. K. Jain and R. C. Dubes. *Algorithms for clustering data*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1988.

[14] G. Jeh and J. Widom. SimRank: a measure of structural-context similarity. In *KDD '02: Proceedings of the 8th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 538–543, New York, NY, USA, 2002. ACM Press.

[15] T. Joachims. A probabilistic analysis of the rocchio algorithm with TFIDF for text categorization. In *ICML '97: Proceedings of the Fourteenth International Conference on Machine Learning*, pages 143–151, San Francisco, CA, USA, 1997. Morgan Kaufmann Publishers Inc.

[16] M. Kessler. Bibliographic coupling between scientific papers. *American Documentation*, 14(10–25), 1963.

[17] J. A. Konstan, B. N. Miller, D. Maltz, J. L. Herlocker, L. R. Gordon, and J. Riedl. Grouplens: applying collaborative filtering to usenet news. *Commun. ACM*, 40(3):77–87, 1997.

[18] R. Kumar, P. Raghavan, S. Rajagopalan, D. Sivakumar, A. Tomkins, and E. Upfal. The Web as a graph. In *Proc. 19th ACM SIGACT-SIGMOD-AIGART Symp. Principles of Database Systems, PODS*, pages 1–10. ACM Press, 15–17 2000.

[19] S. R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. Extracting large-scale knowledge bases from the Web. In *The VLDB Journal*, pages 639–650, 1999.

[20] S. Lawrence and C. L. Giles. Accessibility of information on the Web. *Intelligence*, 11(1):32–39, 2000.

[21] R. Lempel and S. Moran. SALSA: the stochastic approach for link-structure analysis. *ACM Trans. Inf. Syst.*, 19(2):131–160, 2001.

[22] D. Liben-Nowell and J. Kleinberg. The link prediction problem for social networks. In *12th International Conference on Information and Knowledge Management*, pages 556–559. ACM, November 2003.

[23] Z. Lin, I. King, and M. R. Lyu. PageSim: A novel link-based similarity measure for the World Wide Web. In *WI '06: Proceedings of the 5th International Conference on Web Intelligence. To appear*, 2006.

[24] W. Lu, J. Janssen, E. Milios, and N. Japkowicz. Node similarity in networked information spaces. In *CASCON '01: Proceedings of the 2001 conference of the Centre for Advanced Studies on Collaborative research*, page 11. IBM Press, 2001.

[25] P. T. Metaxas and J. Destefano. Web spam, propaganda and trust. In *Proceedings of the First International Workshop on Adversarial Information Retrieval on the Web*, May 2005.

[26] M. Newman. Detecting community structure in networks. *The European Physical Journal B - Condensed Matter*, 38(2):321–330, 2004.

[27] A. Y. Ng, A. X. Zheng, and M. I. Jordan. Stable algorithms for link analysis. In *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 258–266, NY, USA, 2001. ACM Press.

[28] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank citation ranking: bringing order to the Web. Technical report, Stanford Digital Library Technologies Project, 1998.

[29] K. Risvik and R. Michelsen. Search engines and web dynamics. *Computer Networks*, 39(3):289–302, 2002.

[30] G. Salton. *Automatic Text Processing*. Addison-Wesley, 1989.

[31] G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., NY, USA, 1986.

[32] H. Small. Co-citation in the scientific literature: a new measure of the relationship between two documents. *Journal of the American Society for Information Science*, 24(265–269), 1973.