Site Level Noise Removal for Search Engines

André Luiz da Costa Carvalho[1], Paul - Alexandru Chirita[2], Edleno Silva de Moura[1]
Pável Calado[3],Wolfgang Nejdl[2]
[1]Federal University of Amazonas, Av. Rodrigo Octávio Ramos 3000, Manaus, Brazil
andre@ufam.edu.br,edleno@dcc.ufam.edu.br
[2]L3S and University of Hannover, Deutscher Pavillon Expo Plaza 1, 30539 Hannover, Germany
{chirita,nejdl}@l3s.de
[3]IST/INESC-ID, Av. Prof. Cavaco Silva, 2780-990 Porto Salvo, Portugal
pavel.calado@tagus.ist.utl.pt

ABSTRACT

The currently booming search engine industry has determined many online organizations to attempt to artificially increase their ranking in order to attract more visitors to their web sites. At the same time, the growth of the web has also inherently generated several navigational hyperlink structures that have a negative impact on the importance measures employed by current search engines. In this paper we propose and evaluate algorithms for identifying all these noisy links on the web graph, may them be spam or simple relationships between real world entities represented by sites, replication of content, etc. Unlike prior work, we target a different type of noisy link structures, residing at the site level, instead of the page level. We thus investigate and annihilate site level mutual reinforcement relationships, abnormal support coming from one site towards another, as well as complex link alliances between web sites. Our experiments with the link database of the TodoBR search engine show a very strong increase in the quality of the output rankings after having applied our techniques.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

General Terms

Algorithms, Experimentation

Keywords

PageRank, Link Analysis, Noise Reduction, Spam

1. INTRODUCTION

The popularity of search engines has thoroughly increased over the past years. And so has the amount of information they are indexing. At the same time, upon searching this overwhelming quantity of data, people usually view only the top answers returned for each query [20]. It is thus very important to provide these responses with the best quality possible. Alas, currently this is not an easy task.

WWW 2006, May 23–26, 2006, Edinburgh, Scotland.
ACM 1-59593-323-9/06/0005.

Search engines adopt several different sources of evidence to rank web pages matching a user query, such as textual content, title of web pages, anchor text information, or the link structure of the web. This latter measure is one of the most useful sources of evidence adopted. To extract information from the link structure, search engines use algorithms that assess the quality (or popularity) of web pages by analyzing the linkage relationships among them. The success of this strategy relies on the assumption that a link to a page represents a vote from a user that sustains the quality of that page.

In spite of the success of link analysis algorithms, many link structures created using web hyperlinks lead these algorithms to provide wrong conclusions about the quality of web pages. This phenomenon happens because links that cannot be interpreted as votes for quality sometimes negatively affect the link analysis results. We name these here as *noisy links*. A subset of them has already been addressed exhaustively in prior work, namely the nepotistic links (also called spam links), i.e., the links *intentionally* created to artificially boost the rank of some given set of pages, usually referred to as *spam pages* [27]. In fact, given the importance of search engines in modern society, many online organizations currently attempt to artificially increase their rank, since a higher rank implies more users visiting their pages, which subsequently implies an increased profit. Thus, a new industry specialized in creating spam information has emerged, called Search Engine Optimization (SEO) [17].

Even though it is very important to diminish the noise effect induced by spammers (i.e., creators of spam), many other noisy links may appear in the web graph and should be detected as well. For instance, two sites from companies of the same group may be strongly interconnected by links that do not represent votes for quality. Such links are created due to a relationship between the entities represented by both sites and not by the intention to vote for the quality of their pages, as assumed by the link analysis algorithms. In fact, we argue that most of the noisy links on the web are created in a non-intentional way. Therefore, devising algorithms to detect noise in general (which also includes spam) is better than devising algorithms considering only spam attacks over the web graph.

In this paper we propose a site-level approach for detecting generic noisy links on the web. Previous algorithms have focused on identifying noise only by analyzing page level relationships, which clearly misses some of the higher level noise, generated between a group of sites. We investigate three main types of site level

relationships: mutual reinforcement (in which many links are exchanged between the two sites), abnormal support (where most of one site's links are pointing to the same target site), and link alliances (in which several sites create complex link structures that boost the PageRank score of their pages). When the relation between such sets of sites is considered suspicious, we assume that the links between them are noisy and penalize them accordingly. Finally, it is important to note that this new approach is complementary to the existing page level approaches, and both strategies should be adopted simultaneously for identifying noisy links in a search engine database.

We studied the impact of our noise removal techniques on PageRank [25], as it is probably the most popular link analysis algorithm for computing the reputation of pages over the web. Our experiments have shown an improvement of 26.98% in Mean Reciprocal Rank [19] for popular bookmark queries, 20.92% for randomly selected bookmark queries, and up to 59.16% in Mean Average Precision [4] for topic queries. Furthermore, 16.7% of the links from our collection were considered as noisy, while many of them could not be considered nepotistic, thus demonstrating the importance of searching for noisy links in search engine databases instead of searching only for spam.

The paper is organized as follows: Section 2 presents some background information and discusses the related work. Section 3 describes our site-level noise detection approach and the algorithms we proposed. We extensively evaluate these algorithms and analyze their performance in Section 4. Finally, Section 5 draws some conclusions and discusses future work.

2. BACKGROUND AND RELATED WORK

2.1 Background

Preliminaries. Link analysis algorithms are founded on the representation of the web as a graph. Throughout the paper we will refer to this graph as $G = (V, E)$, where V is the set of all web pages and E is a set of directed edges $< p, q >$. E contains an edge $< p, q >$ iff a page p links to a page q. $In(p)$ represents the set of pages pointing to p (the *in-neighbors* of p) and $Out(p)$ the set of pages pointed to by p (the *out-neighbors* of p). We denote the *p-th* component of a vector $\mathbf{v}$ as $v(p)$. Finally, let A be the normalized adjacency matrix corresponding to G with, $A_{ij} = \frac{1}{|Out(j)|}$ if page j links to page i and $A_{ij} = 0$ otherwise.

We now take a closer look at the PageRank algorithm. Since it is one of the most popular link analysis algorithms, we chose to use it to evaluate the impact of our noise removal methods.

PageRank [25] computes web page reputation scores starting from the idea that "a page should receive a high rank if the sum of the ranks of its in-neighbors is also high". Given a page p, the PageRank formula is:

$$PR(p) = (1 - c) \cdot \sum_{q \in In(p)} \frac{PR(q)}{\|Out(q)\|} + \frac{c}{\|V\|} \qquad (1)$$

The damping factor $c < 1$ (usually 0.15) is necessary to guarantee convergence and to limit the effect of rank sinks [9].

2.2 Related work

Most of the previous works on noise are focused on solving the problem of detecting spam, instead of detecting noise in a general sense. Good solutions for the spam detection problem are important and difficult because they need to deal with adversaries that continuously try to deceive search engine algorithms. As the search output is usually ordered using a *combination* of the various algorithms available to assess the quality of each result (e.g., PageRank, HITS, TFxIDF, etc.), spammers have devised specific schemes to circumvent each of these measures. Consequently, the search engines responded with detection or neutralization techniques. This caused the spammers to seek new rank boosting methods, and so on. Since our work here is focused on identifying noise on the link structure of web collections, we will present in this section only the most recent anti-spam techniques for link-based algorithms.

Ranking Based Approaches. The currently known types of artificial noisy link structures which could boost the rank of one or more web pages have been investigated by Gyögyi and Garcia-Molina. [16]. They manually build toy-scale link farms (networks of pages densely connected to each other) or alliances of farms and calculate their impact upon the final rankings. We used their results to design some of our spam fighting algorithms.

The seminal article of Bharat and Henzinger [7] has indirectly addressed the problem of noise detection on the web. Though inherently different from our approach, it is the only work we found in the literature that detects noise at a site level. Further, the authors provide a valuable insight into web noise link detection: They discovered the existence of "mutually reinforcing relationships" and proposed to assign each edge an authority weight of $1/k$ if there are k pages from the one site pointing a single document from another site, as well as a hub weight of $1/l$ if a page from the first site is pointing to l documents residing all on the second site. Authors use this information to change HITS [21], a specific link analysis algorithm. We believe their solution can be used to complement our approach, and we intend to adapt it and integrate it into our algorithms in future work. Later, Li et al. [24] also proposed an improved HITS algorithm to avoid its vulnerability to *small-in-large-out* situations, in which one page has only a few in-links but many out-links. Nevertheless, their work only focuses on this specific problem, thus not tackling noise detection at all.

Another important work is SALSA [23], where the "Tightly-Knit (TKC) Community Effect" is first discussed. The organization of pages into such a densely linked graph usually results in increasing their scores. The authors proposed a new link analysis algorithm which adopts two Markov chains for traversing the web graph, one converging to the weighted in-degree of each page, for authority scores, and the other converging to its weighted out-degree, for hub scores. The approach resembles popularity ranking, which was also investigated by Chakrabarti [12] and Borodin et al. [8]. However, it does not incorporate any iterative reinforcement and is still vulnerable to some forms of the TKC effect [26].

Zhang et al. [29] discovered that colluding users amplify their PageRank score with a value proportional to $Out(1/\epsilon)$, where ϵ is the damping factor. Thus, they propose to calculate PageRank with a different ϵ for each page p, automatically generated as a function of the correlation coefficient between $1/\epsilon$ and $PageRank(p)$ under different values for ϵ. Their work is extended by Baeza-Yates et al. [3], who study how the PageRank increases under various collusion (i.e., nepotistic) topologies and prove this increase to be bounded by a value depending on the original PageRank of the colluding set and on the damping factor.

BadRank [2, 27] is one of the techniques supposed to be used by search engines against link farms. It is practically an inverse PageRank, in which a page will get a high score if it points to many

pages with high BadRank, as depicted in the formula below:

$$BR(p) = (1-c) \cdot \sum_{q \in In(p)} \frac{BR(q)}{\|Out(q)\|} + c \cdot IB(p) \qquad (2)$$

The exact expression of $IB(p)$ is not known, but it represents the initial BadRank value of page p as assigned by spam filters, etc. The algorithm is complementary to our approaches. We therefore plan to investigate in a further work the idea of propagating the badness score of a page as an extension on top of the algorithms presented in this paper.

TrustRank [18] proposes a rather similar approach, but focused on the good pages: In the first step, a set of high quality pages is selected and assigned a high trust; then, a biased version of PageRank is used to propagate these trust values along out-links throughout the entire web. The algorithm is orthogonal to our approaches: Instead of seeking for good pages, we attempt to automatically identify and penalize malicious links, thus decreasing the authority of the bad pages they point to. This ensures that good pages that are accidentally part of a malicious structure will be at most downgraded, but never dismissed from the possible set of query results (as in TrustRank).

SpamRank [5] resembles an "opposite TrustRank": First, each page receives a penalty score proportional to the irregularity of the distribution of PageRank scores for its in-linking pages; then, Personalized PageRank is used to propagate the penalties in the graph. The advantage over TrustRank is that good pages cannot be marked as spam, and comes at a cost of higher time complexity. Our approach is similar with respect to penalizing bad pages, but we build our set of malicious candidates much faster, by identifying abnormal link structures, instead of analyzing the distribution of PageRank scores for the in-linking pages of each page.

Finally, Wu and Davison [27] first mark a set of pages as bad, if the domains of n of their out-links match the domains of n of their in-links (i.e., they count the number of domains that link to and are linked by that page). Then, they extend this set with all pages pointing to at least m pages in the former set, and remove all links between pages marked as bad. Finally, new rankings are computed using the "cleaned" transition probability matrix. Their algorithm is complementary to our approach, as it operates at the lower level of web pages, instead of sites. In [28], the same authors build bipartite graphs of documents and their "complete hyperlinks"[1] to find link farms of pages sharing both anchor text and link targets (i.e., possibly automatically created duplicate links).

Other Approaches. While most of the web noise detection research has concentrated directly on the link analysis algorithms used within current search engines, another significant stream of activity was dedicated to designing innovative third party solutions to detect such unwanted hyperlinks. Kumar et al. [22] used bipartite graphs to identify web communities and marked as nepotistic those communities having several fans (i.e., pages contributing to the core of the bipartite graph with their out-links) residing on the same site. Roberts and Rosenthal [26] analyze the number of *web clusters* pointing to each target page in order to decrease the influence of TKCs. They propose several methods to approximate these clusters, but they evaluate their approach only on a minimal set of queries. A rather different technique is employed in [1], where the authors present a decision-rule classifier that employs 16 connectivity features (e.g., average level of page in the site tree, etc.) to detect web site functionality. They claim to have successfully used it to identify link spam rings as well, but no details are given about the importance of each feature for accomplishing this task.

[1]Hyperlinks having the anchor text attached to them.

Chakrabarti [11] proposed a finer grained model of the web, in which pages are represented by their Document Object Models, with the resulted DOM trees being interconnected by regular hyperlinks. The method is able to counter "nepotistic clique attacks", but needs more input data than our algorithms (which are based exclusively on link analysis). Also, since we specifically target noise removal, we are able to identify different types of link anomalies.

Fetterly et al. [15] use statistical measures to identify potential spam pages. Most of the features they analyze can be modeled by well known distributions, thus placing outliers in the position of potential spammers. After a manual inspection, the vast majority of them seemed to be spammers indeed. A related technique to detect spam pages is based on machine learning algorithms: Davison [13] uses them on several features of URLs (e.g., similar titles, domains, etc.) in order to identify nepotistic links on the web.

Before proceeding to discuss our algorithms, we should note that quite several other types of link noise exist besides spam. The most common is caused by mirror hosts and can be eliminated using algorithms such as those proposed by Broder et al. [10] or Bharat et al. [6]. Also, navigational links are intended to facilitate web browsing, rather than expressing true votes of trust. One work indirectly related to this type of links is [14], where the authors defined web documents as a "cohesive presentation of thought on a unifying subject" and proposed using these entities for information retrieval, instead of the regular web pages. Their work is however orthogonal to ours, as they seek to identify the correct web entities, whereas we propose solutions to remove noisy links from search engine databases.

3. SITE LEVEL NOISE REMOVAL

We argue here that many noisy links can be easily detected when the relationships between sites, instead of pages, are analyzed. Even though the current page centered approaches for detecting noise still hold (and will still be needed in the future), they may not be the best solution to deal with many practical situations. For example, a company having two branches with different sites will probably have only few of its pages involved in page level link exchanges, many other links between its two sites being thus regarded as true votes by the current approaches, even though they connect two entities having the same owner. Similarly, undetected replicated sites could exchange many links only at a high level, or could be linked only in one direction, towards the newer replica of the content. Finally, worst, automatically generated complex site level link spam structures may be missed by the page level approaches. Therefore, we propose detecting noise at a *site level* rather than on page level. More specifically, we propose the following three approaches: (1) Identifying mutual reinforcement relations between web sites, (2) identifying relationships between sites where one site has PageRank scores accumulated mostly from only one different site, and (3) penalizing alliances of sites that artificially promote some target site.

3.1 Site Level Mutual Reinforcement Relations

Our first site level approach to detect noisy links on web collections is based on the study of how connected are pairs of sites. Our assumption in this approach is that when two sites are strongly connected, they artificially boost their results in link analysis algorithms. We name this phenomenon as a *site level mutual reinforcement*. As one of the initial forms of noise, mutual reinforcement relations have been tackled as early as 1998 by Bharat and Henzinger [7]. However, all approaches proposed so far are centered around the web page as a unit item.

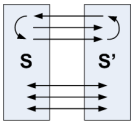

Figure 1: Example of site level link exchanges.

We therefore study the mutual reinforcement problem at the site level, because a considerable amount of noisy links between these type of web sites cannot be detected using approaches working at the page level. We thus consider all links between strongly connected sites as noise, including links between individual pages that are not suspicious per se. These links are considered noise because they can artificially boost the popularity rank of pages that belong to a pair of web sites whose relationship is considered suspicious.

One of the many possible examples of how this noise may affect the link analysis is depicted on the upper side of Figure 1, using cycles. The connection between two sites may create a set of cycles on the web graph containing pages from both of them. It is known that such cycle structures can boost the popularity of web pages [16], and since many cycles can arise from strongly connected sites, such alliances between sites may create anomalies in the final PageRank. Let us now discuss the two different algorithms we propose for detecting mutual site reinforcement relationships.

Bi-Directional Mutual Site Reinforcement (BMSR). This algorithm takes into account the number of link exchanges between pages from the two studied sites. We say that two pages p_1 and p_2 have a link exchange if there is a link from p_1 to p_2 and a link from p_2 to p_1. This first method tries to identify site pairs that have an abnormal amount of link exchanges between their pages. In these cases, we consider the pair as suspicious and all the links between its sites are considered noisy. The threshold to consider a pair of sites suspicious is set through experiments.

Notice that, as in the work of Wu and Davison [27], BMSR is based on link exchanges. However, while they perform a page level analysis of exchanges and they are interested in identifying spam pages exclusively, we perform a site level analysis, while being interested in identifying the more general phenomenon of noisy links.

Uni-Directional Mutual Site Reinforcement (UMSR). As sites are large structures, we also investigate the possibility of relaxing the notion of "link exchange" into "link density", i.e., counting all links between two sites, disregarding their orientation.

On the example from Figure 1, there are 3 link exchanges between the sites s and s' and the link density is 9 (link exchanges are also counted). In order to calculate these values, one needs to iterate over all pages, and for each page to increment the site level statistics every time a link exchange is found (see Algorithm 3.1.1 below, lines 5-8), for BMSR, or simply every time a link is encountered (Algorithm 3.1.1, lines 5-6, and 9), for UMSR. In order to keep clear the idea behind the algorithm, we did not include in the description below several straightforward performance optimizations, such as computing the BMSR for some sites s and s' only once, as it is the same with that for s' and s. Also, Algorithm 3.1.1 computes the link density as a measure of UMSR.

Algorithm 3.1.1. Detecting Link Exchanges at Site Level.

1: Let $BMSR(s, s')$ and $UMSR(s, s')$ denote the amount of link exchanges and the link density between sites s and s' respectively.
2: **For** each site s
3: **For** each site $s' \neq s$
4: $BMSR(s, s') = UMSR(s, s') = 0$
5: **For** each page $p \in V$, p residing on site s
6: **For** each page $q \in Out(p)$, q from site $s' \neq s$
7: **If** $p \in Out(q)$ **then**
8: $BMSR(s, s') = BMSR(s, s') + 1$
9: $UMSR(s, s') = UMSR(s', s) = UMSR(s, s') + 1$

Computing Page Ranks. Let us now see how we could use these measures to improve PageRank quality. An approach is depicted in Algorithm 3.1.2, which removes all links between all pairs of sites (s, s'), if the BMSR or UMSR values between them are above a certain threshold. In our experiments, we used 10, 20, 50, 100, 250 and 300 for link density (250 being best, yet still with poor performance), and 1, 2, 3 and 4 for link exchanges (with 2 having better results, indicating that most sites exchange incorrect votes, or links, with only a few partners, like a company with its branches).

Algorithm 3.1.2. Removing Site-Level Link Exchanges.

1: **For** each site s
2: **For** each site s'
3: **If** $*MSR(s, s') \geq \epsilon_{*MSR*}$ **then**
4: Remove all links between s and s'
5: Compute regular PageRank.

3.2 Site Level Abnormal Support

Another type of situation we consider as a noisy relation between sites is the *site level abnormal support*(**SLAbS**). It occurs when a single site is responsible for a high percentage of the total amount of links pointing to another site. This situation can easily arise within a web collection. For instance, and unfortunately, once the spammers have read the previous section, they could start to seek for new schemes that circumvent the algorithms we presented. A relatively simple approach they could take is to create chains of sites supporting each other through a limited number of links (see Figure 2 for an example). This is because their space of available choices is diminishing: Using too many links would make them detectable by our site level mutual reinforcement algorithms above, while using other structures than chains (e.g., hierarchy of sites) would visibly make their success more costly. We therefore propose the following axiom:

AXIOM 1. *The total amount of links to a site (i.e., the sum of links to its pages) should not be strongly influenced by the links it receives from some other site.*

In other words, for any site s, there should not be a site $s' \neq s$, whose number of links towards s is above a certain percentage of the total number of links s receives overall. In our experiments,

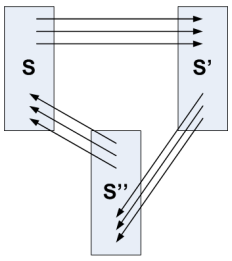

Figure 2: Example of site chains.

we tested with thresholds ranging from 0.5% up to 20% of the total number of links to s and the best results were achieved at 2%. Whenever such a pair of sites (s, s') is found, all links between them are marked as noisy links. Note that links from s to s' are also taken as noise because we consider the relationship between them suspicious. Finally, after this trimming process is over, we remove the detected noisy links and the regular PageRank algorithm is run over the cleaned link database. The approach is also summarized in Algorithm 3.2.

Algorithm 3.2.. Removing Site-Level Abnormal Support(SLAbS).

1: **For** each site s
2: let t be the total number of links to pages of s
3: **For** each site s' that links to s
4: let $t_{(s',s)}$ be the number of links from s' to s
5: $supp = t_{(s',s)}/t$
6: **If** $supp \geq \epsilon_{AS}$ **then**
7: Remove all links between s' and s
8: Compute regular PageRank.

3.3 Site Level Link Alliances

Another hypothesis we considered is that the popularity of a site cannot be supported only by a group of strongly connected sites. The intuition behind this idea is that a web site is as popular as diverse and independent are the sites that link to it. In fact, as we will see from the experiments section, our algorithm which detects and considers this concept of independence when computing PageRank gives a strong improvement in the overall quality of the final rankings.

Further, continuing the scenario discussed in the previous Section, suppose spammers do have enough resources available to build complex hierarchies of sites that support an end target site, as illustrated in Figure 3. These hierarchies have previously been named *Link Spam Alliances* by Gyöngyi and Garcia-Molina [16], but they did not present any solution to counteract them. Such structures would generate sites linked by a strongly connected community, thus contradicting our general hypothesis about the relation between diversity of sites that link to a site and its actual popularity.

Before discussing our approach, we should note that we do not address page level link alliances, i.e., hierarchies of pages meant to support an end target page, all pages residing on the same site, or on very few sites. These types of structures could be easily annihilated for example by using different weights for intra-site and inter-site links, or by implementing the approach presented by Bharat and Henzinger in [7], where every in-link of some page p is assigned the weight $1/k$ if there are k pages pointing to p (for link alliances distributed over several sites).

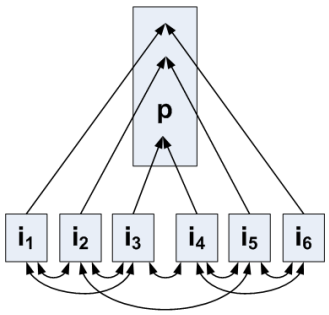

Figure 3: Example of link alliance spanning over several sites.

The more complicated situation is to find link alliances (intentional or not) over several sites, as the one depicted in Figure 3 (boxes represent sites). Our intuition is that these alliances would still have to consist of highly interconnected pages. More specifically, if a page p has in-links from pages i_1, i_2, ..., i_I, and these latter pages are highly connected, then they are suspect of being part of a structure which could deceive popularity ranking algorithms. We evaluate the degree of susceptivity using the following algorithm:

Algorithm 3.3.1.. Computing Link Alliance Susceptivity.

1: **For** each page p
2: Let Tot count the out-links of all pages $q \in In(p)$
3: Let $TotIn$ count the out-links of all pages $q \in In(p)$, such that they point to some other page from $In(p)$
4: **For** each page $q \in In(p)$
5: **For** each page $t \in Out(q)$
6: $Tot = Tot + 1$
7: **If** $t \in In(p)$ **then**
8: $TotIn = TotIn + 1$
9: Susceptivity(p) = $\frac{TotIn}{Tot}$.

Once the susceptivity levels are computed, we downgrade the in-links of every page p with $(1 - Susceptivity(p))$, uniformly distributing the remaining votes to all pages. This latter step is necessary in order to ensure the convergence of the Markov chain associated to the web graph, i.e., to ensure the sum of transition probabilities from each state st remains 1. The entire approach is also presented in Algorithm 3.3.2.

Algorithm 3.3.2.. Penalizing Site-Level Link Alliances.

1: Let $PR(i) = 1/\|V\|, \forall i \in \{1, 2, ..., \|V\|\}$
2: **Repeat** until convergence
3: **For** each page p
4: $PR(p) = (1 - Susceptivity(p)) \cdot (1 - c) \cdot \sum_{q \in In(p)} \frac{PR(q)}{\|Out(q)\|} + \frac{c}{\|V\|}$
5: $Residual = Susceptivity(p) \cdot (1 - c) \cdot \sum_{q \in In(p)} \frac{PR(q)}{\|Out(q)\|}$
6: **For** each page p'
7: $PR(p') = PR(p') + \frac{Residual}{\|V\|}$

4. EXPERIMENTS

4.1 Experimental Setup

We evaluated the impact of our noise removal techniques on the link database of the TodoBR search engine. This database consisted of a collection of 12,020,513 pages extracted from the Brazilian web, connected by 139,402,245 links. As it represents a considerably connected snapshot of the Brazilian web community, which is probably as diverse in content and link structure as the entire web, we think it makes a realistic testbed for our experiments.

In order to evaluate the impact of our algorithms within practical situations, we extracted test queries from the TodoBR log, which is composed of 11,246,351 queries previously submitted to the search engine. We divided these selected queries in two groups:

1. *Bookmark queries*, in which a specific web page is sought;
2. *Topic queries*, in which people are looking for information on a given topic, instead of some page.

Each query set was further divided in two subsets, as follows:

- *Popular queries*: Here, we selected the top most popular bookmark / topic queries found in the TodoBR log. These queries usually search for well known web sites and are useful to check what happens to these most commonly searched pages after the noise removal methods have been applied.
- *Randomly selected queries*: Here, we selected the queries randomly. These queries tend to search for less popular sites and show the impact of our techniques on pages that are probably not highly ranked by PageRank.

Then, 14 undergraduate and graduate computer science students (within different areas) evaluated the selected queries under various experimental settings. All of them were familiar with the Brazilian web pages and sites, in order to ensure more reliability to our experiments.

The bookmark query sets contained each 50 queries, extracted using the above mentioned techniques. All bookmark query results were evaluated using MRR (Mean Reciprocal Ranking), which is the metric adopted for bookmark queries on the TREC Conference[2] and is computed by the following equation:

$$MRR(QS) = \frac{\sum_{\forall q_i \in QS} \frac{1}{PosRelAns(q_i)}}{|QS|} \quad (3)$$

where QS is the set of queries we experiment on, and $PosRelAns(q_i)$ is the position of the first relevant answer in the rankings output for query q_i. MRR is the most common metric for evaluating the quality of results in bookmark queries. As it can be seen, its formula prioritizes methods that obtain results closer to the top of the ranking, adopting an exponential reduction in the scores (i.e., higher scores are better), as the position of the first relevant answer in the ranking increases. Also, MRR is very good at assessing the "real life" performance of the search engine, as the URLs most likely to be visited are those returned at the very top of the result list [20]. However, MRR is not sensible to pages having huge drops in position (e.g., from place 15 to place 40). Therefore, we also adopted another measure, mean position, (denoted MPOS in the tables to follow), which computes the average position of the first relevant answer in the output provided for each query. This metric results in a linear increase in the scores (higher is worse) as the position of the relevant answer increases.

For topic queries, we used two sets of 30 queries also selected from the TodoBR log as described previously. These different queries evaluate the impact of our noise removal algorithms when searching for some given topics. In this case, we evaluated the results using the same pooling method as used within the Web Collection of TREC [19]. We thus constructed query pools containing the first top 20 answers for each query and algorithm. Then, we assessed our output in terms of various precision based metrics. The *precision* of an algorithm is defined as the number of relevant results returned divided by the total number of results returned. For each algorithm, we evaluated the Mean Average Precision (MAP), the precision at the first 5 positions of the resulted ranking (P@5), as well as the precision at the top 10 output rankings (P@10). In all cases the relevant results were divided in two categories, (1) relevant and (2) highly relevant. Also, we processed all queries according to the user specifications, as extracted from the TodoBR log: phrases, Boolean conjunctive or Boolean disjunctive. The set of documents achieved for each query was then ranked according to the PageRank algorithm, with and without each of our link removal techniques applied. Finally, all our results were tested for statistical significance using T-tests (i.e., we tested whether the improvement over PageRank without any links removed is statistically significant).

In all the forthcoming tables, we will label the algorithms we evaluated as follows:

- **ALL LINKS**: No noise removal.
- **UMSR**: Uni-directional Mutual Site Reinforcement.
- **BMSR**: Bi-directional Mutual Site Reinforcement.
- **SLAbS**: Site Level Abnormal Support.
- **SLLA**: Site Level Link Alliances.
- Combinations of the above, in which every method is applied independently to remove (UMSR, BMSR, SLAbS) or downgrade (SLLA) links, and then PageRank is applied on the resulting "cleaned" graph. Several combinations have been tested, but due to space limitations, only the best will be presented here.

Algorithm specific aspects. Another important setup detail is how to divide the collection into web sites, as the concept of *web site* is rather imprecise. In our implementation, we adopted the host name part of the URLs as the keys for identifying individual web sites. This is a simple, yet very effective heuristic to identify sites, as pages with different host names usually belong to different sites, while those with identical host names usually belong to the same site.

As UMSR, BMSR and SLAbS all use thresholds to determine whether links between pairs of sites are noisy or not, it is important to tune such thresholds in order to adjust the algorithms to the collection in which they are applied. For the experiments we performed, we adopted the MRR results achieved for bookmark queries as the main parameter to select the best threshold. This metric was adopted because link information tends to have a greater impact on bookmark queries than on topic queries. Further, MRR can be calculated automatically, reducing the cost for tuning. The best parameters for each method depend on the database, the amount of noisy information and the requirements of the search engine where they will be applied.

Table 1 presents the best thresholds we found for each algorithm using MRR as the tuning criteria. These parameters were adopted in all the experiments presented.

4.2 Results

Bookmark Queries. We evaluated the bookmark queries in terms of Mean Reciprocal Rank (MRR) and Mean Position (MPOS) of the first relevant URL output by the search engine. Table 2 shows the MRR scores for each algorithm with popular bookmark queries. The best result was achieved when combining all

[2]http://trec.nist.gov/

Method	Threshold
UMSR	250
BMSR	2
SLAbS	2%

Table 1: Best thresholds found for each algorithm using MRR as the tuning criteria.

Method	MRR	Gain [%]	Significance
ALL LINKS	0.3781	-	-
UMSR	0.3768	-0.53%	*No*, 0.34
BMSR	0.4139	9.48%	*Highly*, 0.008
SLAbS	0.4141	9.5%	*Yes*, 0.04
SLLA	0.4241	12.14%	*Yes*, 0.03
BMSR+SLAbS	0.4213	11.40%	*Yes*, 0.02
SLLA+BMSR	0.4394	16.20%	*Highly*, 0.01
SLLA+SLAbS	0.4544	20.17%	*Highly*, 0.003
SLLA+BMSR+SLAbS	**0.4802**	**26.98%**	***Highly*, 0.001**

Table 2: Mean Reciprocal Rank (higher is better) for popular bookmark queries.

the noise detection methods proposed, showing an improvement of 26.98% in MRR when compared to PageRank. The last column shows the T-test results, which indicate the statistical significance of the difference in results[3] for each database, when compared to the ALL LINKS version (i.e., PageRank on the original link database). The only method that had a negative impact on MRR was the UMSR, which indicates that many unidirectional relations between sites are rather useful for the ranking. This was also the only algorithm for which the T-test indicated a non-significant difference in the results (p-values lower than 0.25 are taken as marginally significant, lower than 0.05 are taken as significant, and lower than 0.01 as highly significant).

Table 3 presents the Mean Position of the first relevant result (MPOS) achieved for popular bookmark queries under each of the algorithms we proposed. The best combination remains *SLLA+BMSR+SLAbS*, with a gain of 37.00%. Thus, we conclude that for popular bookmark queries the combination of all methods is the best noise removal solution. Also, individually, Site Level Link Alliance (SLLA) produced the highest increase in PageRank quality.

After having evaluated our techniques on popular bookmark queries, we tested their performance over the randomly selected set. The MRR results for this scenario are displayed in Table 4. Again, the best outcome was achieved when combining all the noise detection methods proposed, with an improvement of 20.92% in MRR, when compared to PageRank. Note that an improvement is harder to achieve under this setting, since the web pages searched are not necessarily popular, and thus many of them may have just a few in-going links and consequently a low PageRank score. Therefore, as removing links at the site level might also have the side effect of a further decrease of their PageRank score, they could become even more difficult to find. This is why both site level mutual reinforcement algorithms (BMSR and UMSR) resulted in a negative impact in the results, indicating that *some* site level mutual reinforcement might not necessarily be a result of noise (at least the uni-directional type of reinforcement). Similar results have been observed when computing the Mean Position of the first relevant result, instead of the MRR (see Table 5). Individually, SLLA is still the best algorithm, whereas the best technique overall is again the combined SLLA+BMSR+SLAbS.

[3]Recall that statistical significance is not computed on the average result itself, but on each evaluation evidence (i.e., it also considers the agreement between subjects when assessing the results). Thus, smaller average differences could result in a very significant result, if the difference between the two algorithms remains relatively constant for each subject.

Method	MPOS	Gain [%]	Significance
ALL LINKS	6.35	-	-
UMSR	6.25	1.57%	*No*, 0.34
BMSR	5.37	18.25%	*Yes*, 0.04
SLAbS	5.84	8.72%	*No*, 0.26
SLLA	5	27.06%	*Highly*, 0.003
BMSR+SLAbS	5.63	12.89%	*Minimal*, 0.12
SLLA+BMSR	4.84	31.17%	*Highly*, 0.01
SLLA+SLAbS	4.68	35.86%	*Highly*, 0.002
SLLA+BMSR+SLAbS	**4.62**	**37.29%**	***Yes*, 0.04**

Table 3: Mean position of the first relevant result obtained for popular bookmark queries.

Method	MRR	Gain [%]	Signific., p-value
ALL LINKS	0.3200	-	-
UMSR	0.3018	-5.68%	*Highly*, 0.01
BMSR	0.3195	-0.17%	*No*, 0.45
SLAbS	0.3288	2.73%	*No*, 0.31
SLLA	0.3610	12.81%	*Yes*, 0.04
BMSR+SLAbS	0.3263	-2.19%	*No*, 0.36
SLLA+BMSR	0.3632	13.47%	*Yes*, 0.03
SLLA+SLAbS	0.3865	20.78%	*Yes*, 0.017
SLLA+BMSR+SLAbS	**0.3870**	**20.92%**	***Yes*, 0.016**

Table 4: Mean Reciprocal Rank (higher is better) for randomly selected bookmark queries.

Topic Queries. As mentioned earlier in this section, we evaluated topic queries using precision at the top 5 results (P@5) and at the top 10 results (P@10), as well as mean average precision (MAP). We first turn our attention to the experiment in which the output URLs assessed both as relevant and highly relevant are considered as good results. Table 6 presents the evaluation for the most popular 30 topic queries under this scenario. All results were tested for significance, and in both P@5 and P@10 no method manifested a significant gain or loss. Even so, in both P@5 and P@10 we see that BMSR has a slight gain over UMSR. SLLA exhibited the greatest gain in P@5, but the results were relatively similar for all algorithms in P@10. As for MAP, most of the results (except for SLAbS, BMSR, and their combination) had significant gain on MAP, when compared with the database without noise removal. Finally, SLAbS performance was rather poor. However, this behavior of SLAbS was recorded only with this kind of queries, where it is also explainable: Some very popular sites might indeed get an abnormal support from several of their fans; some would consider this as noise, but our testers apparently preferred to have the ranks of these sites boosted towards the top. The best individual method was SLLA and the best combination was SLLA with BMSR, which was better than the combination of all three methods due to the negative influence of SLAbS.

The same experiment was then performed for the 30 randomly selected topic queries. Its results are depicted in Table 7. Here, SLLA remains a very effective individual algorithm, but SLAbS shows even better results. This indicates that an abnormal support for less popular sites usually appears as a result of noise. Moreover,

Method	MPOS	Gain [%]	Significance
ALL LINKS	8.38	-	-
UMSR	8.61	-2.71%	*Highly*, 0.01
BMSR	8.28	1.28%	*No*, 0.27
SLAbS	8.23	1.80%	*Minimal*, 0.24
SLLA	7.42	12.89%	*Minimal*, 0.11
BMSR+SLAbS	8.02	4.09%	*No*, 0.36
SLLA+BMSR	7.27	15.21%	*Minimal*, 0.07
SLLA+SLAbS	7.12	17.61%	*Highly*, 0.01
SLLA+BMSR+SLAbS	**7**	**19.76%**	***Highly*, 0.005**

Table 5: Average mean position of the first relevant result for randomly selected bookmark queries.

Method	P@5	P@10	MAP	Signif. for MAP
ALL LINKS	0.255	0.270	0.198	-
UMSR	0.255	0.282	0.207	*Highly*, 0.0031
BMSR	0.260	0.285	0.198	*No*, 0.3258
SLAbS	0.226	0.262	0.185	*Minimal*, 0.0926
SLLA	0.275	0.270	0.227	*Highly*, 0.0030
BMSR+SLAbS	0.226	0.276	0.200	*No*, 0.3556
SLLA+SLAbS	0.245	0.255	0.216	*Yes*, 0.0429
SLLA+BMSR	**0.270**	**0.273**	**0.231**	***Highly*, 0.0031**
SLLA+BMSR+SLAbS	0.245	0.259	0.223	*Yes*, 0.0129

Table 6: Precision at the first 5 results, at the first 10 results, and Mean Average Precision considering *all* the relevance judgments for popular topic queries.

due to this special behavior of our algorithms, under this setting the main contributor to the combined measures was SLAbS, thus yielding the best MAP score for BMSR+SLAbS.

Before concluding this analysis, we also measured the quality of our methods under the same setting, but considering only the highly relevant output URLs as good results (recall that our subjects evaluated each URL as irrelevant, relevant and highly relevant for each query). For the popular topic queries (Table 8), the performance of the individual methods was similar to the scenario that considered both relevant and highly relevant results, with the main difference being that here SLAbS gains about 12% over the database without noise removal, instead of losing. This is because the sites previously discovered due to noise (i.e., those being very popular, but also abnormally supported by some fans) were considered only relevant by our testers, and thus not included in this more strict experiment. Finally, for the randomly selected queries (Table 9), SLAbS again showed the best individual performance (just as in the sibling experiment considering both kinds of relevance judgments), with the overall top scores being achieved for SLLA+BMSR+SLAbS and BMSR+SLAbS.

Conclusion. In order to make our results more clear, we also plotted their relative gain over regular PageRank (i.e., without noise removal). Figure 4 depicts this gain in percentage for bookmark queries and Figure 5 depicts it for topic queries. We first note that UMSR yielded negative results in three of the four experiments with bookmark queries, which makes it less preferable to its sibling BMSR, even though it performed better than the latter with topical queries. Also, we observe that SLAbS performed quite well under both broad experimental settings, but SLLA is clearly the best single approach for bookmark queries. Finally, all combined measures performed very well, with SLLA+BMSR+SLAbS being the best combination.

Method	P@5	P@10	MAP	Signif. for MAP
ALL LINKS	0.412	0.433	0.311	-
UMSR	0.442	0.442	0.333	*Highly*, 0.0030
BMSR	0.400	0.445	0.314	*No*, 0.3357
SLAbS	0.436	0.458	0.340	*Yes*, 0.0112
SLLA	0.461	0.455	0.327	*Yes*, 0.0125
BMSR+SLAbS	0.448	**0.470**	**0.358**	*Highly*, 0.0012
SLLA+BMSR	**0.485**	0.448	0.326	*Highly*, 0.0006
SLLA+SLAbS	0.461	0.461	0.354	*Minimal*, 0.0618
SLLA+BMSR+SLAbS	0.461	0.467	0.346	*Highly*, 0.0002

Table 7: Precision at the first 5 results, at the first 10 results, and Mean Average Precision considering *all* the relevance judgments for random topic queries.

Method	P@5	P@10	MAP	Signif. for MAP
ALL LINKS	0.152	0.141	0.112	-
UMSR	0.152	0.147	0.131	*Highly*, 0.0002
BMSR	0.152	0.150	0.127	*Highly*, 0.0022
SLAbS	0.152	0.147	0.126	*Yes*, 0.0172
SLLA	**0.162**	0.153	0.163	*Highly*, 0.00003
BMSR+SLAbS	0.152	**0.156**	0.128	*Highly*, 0.0016
SLLA+SLAbS	0.157	0.147	0.175	*Highly*, 0.00002
SLLA+BMSR	0.157	0.153	0.168	*Highly*, 0.00005
SLLA+BMSR+SLAbS	0.157	0.150	**0.179**	*Highly*, 0.00001

Table 8: Precision at the first 5 results, at the first 10 results, and Mean Average Precision considering *only the highly relevant* results selected by our subjects for popular topic queries.

4.3 Practical Issues

Amount of removed links. Even though the amount of removed links does not necessarily represent a performance increase in each algorithm, it is still interesting to see how much did they trim the original link structure. We thus present these values in Table 10 (recall that SLLA does not remove any links, but only downgrades them). We observe that BMSR has removed a relatively low amount of links, when compared to the other methods, which indicates that SLLA+SLAbS could be preferred in practical implementations when faster computations of the algorithm are desired, at the cost of a minimally lower output quality.

Scalability. Algorithms dealing with large datasets, such as the web, need to have a very low complexity in order to be applied in a real environment. All the algorithms we proposed in this paper have, in fact, a computational cost that is linear in the number of pages.

Both Mutual Reinforcement detection algorithms behave in a similar way, with UMSR being slightly less expensive than BMSR. The former needs a simple pass over all links and thus has the complexity $O(|E|)$. Let $M = Average_{p \in V}(Out(p))$, and assume the in-link information is present in the search engine database. If the in-links are in a random order, the complexity of BMSR is $O(|V| \cdot M^2)$, with M^2 being the cost of sequential searching. If the in-links are already sorted, then the complexity falls to $O(|V| \cdot M \cdot log(M))$.

SLAbS is very similar to UMSR. For each page p we update the statistics about its in-going links. Thus, if $P = Average_{p \in V}(In(p))$, then the computational complexity of SLAbS is $O(|V| \cdot P)$.

SLLA is based on the in-links of a page p that are not from the same site as p. Thus, the algorithm needs to calculate the amount of links from pages from $In(p)$ that point to other pages within $In(p)$.

Method	P@5	P@10	MAP	Signif. for MAP
ALL LINKS	0.170	0.179	0.187	-
UMSR	0.176	0.191	0.196	*Yes*, 0.0457
BMSR	0.170	0.185	0.195	*Minimal*, 0.0520
SLAbS	0.182	0.191	0.201	*Yes*, 0.0200
SLLA	0.164	0.185	0.194	*No*, 0.2581
BMSR+SLAbS	0.188	0.197	0.207	*Highly*, 0.0068
SLLA+BMSR	0.182	0.194	0.205	*Highly*, 0.0090
SLLA+SLAbS	0.182	0.206	0.203	*Yes*, 0.0180
SLLA+BMSR+SLAbS	**0.200**	**0.212**	**0.208**	*Highly*, 0.0012

Table 9: Precision at the first 5 results, at the first 10 results, and Mean Average Precision considering *only the highly relevant* results selected by our subjects for random topic queries.

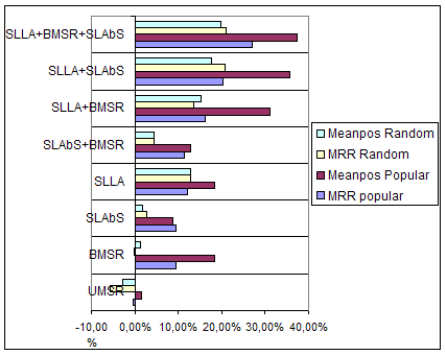

Figure 4: Relative gain (in %) of each algorithm in MRR and Mean Position for bookmark queries.

If the out-links or the in-links are already sorted, the complexity of this approach is $O(|V| \cdot M^2 \cdot log(M))$. Otherwise, the complexity is $O(|V| \cdot M^3)$, since a sequential search is now needed.

Finally, we note that all algorithms we proposed in this paper do a page-by-page processing, which greatly simplifies a any possible concurrent implementation.

5. CONCLUSIONS AND FUTURE WORK

In this paper we proposed using site level link analysis to detect the noisy links from search engine link databases. We designed and evaluated algorithms tackling three types of inappropriate site level relationships: mutual reinforcement, abnormal support and link alliances. Our experiments have showed an improvement of 26.98% in Mean Reciprocal Rank for popular bookmark queries, 20.92% for randomly selected bookmark queries, and up to 59.16% in Mean Average Precision for topic queries. Furthermore, our algorithms identified up to 16.7% of the links from our collection as noisy, while many of them could not be considered nepotistic, thus demonstrating that searching for noisy links in search engine databases is more important than searching only for spam.

While most of the algorithms we presented in this paper directly removed the identified malicious links, in future work we intend to investigate using different weights for various types of links, according to the relation they represent (i.e., inter-site or intra-site relation), as well as to their probability of representing a vote of importance. Additionally, we would like to study more complex (eventually automatic) approaches to tune up the parameter thresholds, instead of using the MRR scores resulted for bookmark queries.

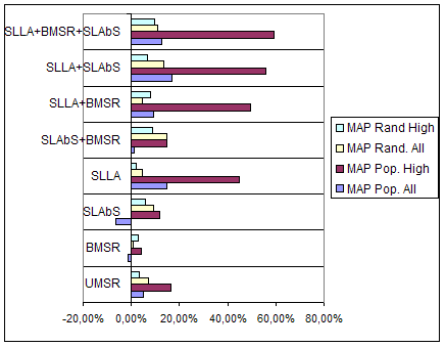

Figure 5: Relative gain (in %) in MAP for all algorithms for topic queries, considering only highly relevant results as relevant (High), and considering both relevant and highly relevant answers as relevant (All).

Method	Links Detected	% of Total Links
UMSR	9371422	7.16%
BMSR	1262707	0.96%
SLAbS	21205419	16.20%
UMSR+BMSR	9507985	7.26%
BMSR+SLAbS	21802313	16.66%

Table 10: Amount of links removed by each of our algorithms.

6. ACKNOWLEDGMENTS

This work is partially supported by GERINDO Project–grant MCT/CNPq/CT-INFO 552.087/02-5, SIRIA/CNPq/CT-Amazônia/55.3126/2005-9, CNPq individual grant 303576/2004-9 (Edleno S. de Moura) and FCT project IR-BASE, ref.POSC/EIA/58194/2004(Pável Calado)

7. REFERENCES

[1] E. Amitay, D. Carmel, A. Darlow, R. Lempel, and A. Soffer. The connectivity sonar: detecting site functionality by structural patterns. In *Proceedings of the 14th ACM Conference on Hypertext and Hypermedia*, pages 38–47, 2003.

[2] Badrank. http://en.efactory.de/e-pr0.shtml.

[3] R. Baeza-Yates, C. Castillo, and V. López. Pagerank increase under different collusion topologies. In *First International Workshop on Adversarial Information Retrieval on the Web*, 2005.

[4] R. A. Baeza-Yates and B. A. Ribeiro-Neto. *Modern Information Retrieval*. ACM Press / Addison-Wesley, 1999.

[5] A. A. Benczur, K. Csalogany, T. Sarlos, and M. Uher. Spamrank - fully automatic link spam detection. In *First International Workshop on Adversarial Information Retrieval on the Web*, 2005.

[6] K. Bharat, A. Z. Broder, J. Dean, and M. R. Henzinger. A comparison of techniques to find mirrored hosts on the WWW. *Journal of the American Society of Information Science*, 51(12):1114–1122, 2000.

[7] K. Bharat and M. R. Henzinger. Improved algorithms for topic distillation in a hyperlinked environment. In *Proc. of 21st ACM International SIGIR Conference on Research and*

Development in Information Retrieval, pages 104–111, Melbourne, AU, 1998.

[8] A. Borodin, G. O. Roberts, J. S. Rosenthal, and P. Tsaparas. Finding authorities and hubs from link structures on the world wide web. In *Proceedings of the 10th International Conference on World Wide Web*, pages 415–429, 2001.

[9] S. Brin, R. Motwani, L. Page, and T. Winograd. What can you do with a web in your pocket? *Data Engineering Bulletin*, 21(2):37–47, 1998.

[10] A. Z. Broder, S. C. Glassman, M. S. Manasse, and G. Zweig. Syntactic clustering of the web. *Comput. Netw. ISDN Syst.*, 29(8-13):1157–1166, 1997.

[11] S. Chakrabarti. Integrating the document object model with hyperlinks for enhanced topic distillation and information extraction. In *Proc. of the 10th International Conference on World Wide Web*, pages 211–220, 2001.

[12] S. Chakrabarti. *Mining the Web: Discovering Knowledge from Hypertext Data*. Morgan Kaufmann, 2003.

[13] B. Davison. Recognizing nepotistic links on the web. In *Proceedings of the AAAI-2000 Workshop on Artificial Intelligence for Web Search*, 2000.

[14] N. Eiron and K. S. McCurley. Untangling compound documents on the web. In *Proc. of the 14th ACM Conference on Hypertext and Hypermedia*, pages 85–94, 2003.

[15] D. Fetterly, M. Manasse, and M. Najork. Spam, damn spam, and statistics: using statistical analysis to locate spam web pages. In *WebDB '04: Proceedings of the 7th International Workshop on the Web and Databases*, pages 1–6, 2004.

[16] Z. Gyöngyi and H. Garcia-Molina. Link spam alliances. In *Proc. of the 31st International VLDB Conference on Very Large Data Bases*, pages 517–528, 2005.

[17] Z. Gyöngyi and H. Garcia-Molina. Web spam taxonomy. In *Proceedings of the Adversarial Information Retrieval held the 14th Intl. World Wide Web Conference*, 2005.

[18] Z. Gyöngyi, H. Garcia-Molina, and J. Pendersen. Combating web spam with trustrank. In *Proceedings of the 30th International VLDB Conference*, 2004.

[19] D. Hawking, E. Voorhees, N. Craswell, and P. Bailey. Overview of the trec8 web track. In *Eighth Text Retrieval Conference*, 1999.

[20] T. Joachims, L. Granka, B. Pan, H. Hembrooke, and G. Gay. Accurately interpreting clickthrough data as implicit feedback. In *Proc. of the 28th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2005.

[21] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, 1999.

[22] R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. Trawling the web for emerging cyber-communities. In *Proceeding of the 8th International Conference on World Wide Web*, pages 1481–1493, 1999.

[23] R. Lempel and S. Moran. The stochastic approach for link-structure analysis (SALSA) and the TKC effect. *Computer Networks (Amsterdam, Netherlands: 1999)*, 33(1–6):387–401, 2000.

[24] L. Li, Y. Shang, and W. Zhang. Improvement of hits-based algorithms on web documents. In *Proceedings of the 11th International Conference on World Wide Web*, pages 527–535, 2002.

[25] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford University, 1998.

[26] G. Roberts and J. Rosenthal. Downweighting tightly knit communities in world wide web rankings. *Advances and Applications in Statistics (ADAS)*, 3:199–216, 2003.

[27] B. Wu and B. Davison. Identifying link farm spam pages. In *Proceedings of the 14th World Wide Web Conference*, 2005.

[28] B. Wu and B. Davison. Undue influence: Eliminating the impact of link plagiarism on web search rankings. Technical report, LeHigh University, 2005.

[29] H. Zhang, A. Goel, R. Govindan, K. Mason, and B. van Roy. Improving eigenvector-based reputation systems against collusions. In *Proceedings of the 3rd Workshop on Web Graph Algorithms*, 2004.