A Method for Finding Link Hijacking Based on Modified PageRank Algorithms

Young joo Chung Masashi Toyoda Masaru Kitsuregawa

Institute of Industrial Science, University of Tokyo 4-6-1 Komaba Meguroku, Tokyo, JAPAN
E-mail: {raysylph, toyoda, kitsuregawa}@tkl.iis.u-tokyo.ac.jp

Abstract As the search result ranking is getting important for attracting visitors and yielding profits, more and more people are now trying to mislead search engines in order to get a higher ranking. Since link-based ranking algorithms important tools for current search engines, web spammers are making a significant effort to manipulate the links structure of the Web, namely, link spamming. Link hijacking is an essential technique for link spamming. By link hijacking, spammers can make search engines believe that normal sites endorse spam sites. In this paper, we propose a link analysis technique for finding link-hijacked sites using modified PageRank algorithms. We performed experiments on our large scale Japanese web archive and evaluated the accuracy of our method.

Keyword Link analysis, Web spam, Information retrieval

1. INTRODUCTION

In the last decade, search engines have been the essential tools for information retrieval. As more and more people rely heavily on search engines to find information in the Web, most of web sites obtain a considerable number of visitors from search engines. Since the increase in visitors usually means the increase in financial profit, and approximately 50% of search engine users look at no more than the first 5 result in the list [1], obtaining high rankings in search results becomes crucial for the success of sites.

Web spamming is defined as the behavior of manipulating the web page features to get a higher ranking than it deserves. Web spamming technique can be categorized into term spamming and link spamming [2]. *Term spamming* is the behavior to manipulate textual contents of pages. Repeating specific keywords and adding irrelevant meta-keywords or anchor texts that are not related with page contents are typical term spam techniques. Search engines that use textual relevance to rank pages will show these manipulated pages at the top of the result list. *Link spamming* is the behavior of manipulating the link structure of the Web to mislead link-based ranking algorithms such as PageRank [3]. For example, spammers can construct a *spam farm*, an artificially interlinked link structure, with a purpose of centralizing link-based importance scores to target spam pages [4]. In addition to building spam farms, spammers should make links from external reputable pages to target spam pages, even though the authors of the external pages do not want to link to them. This behavior is called *link hijacking*. Posting comments including URLs to spam pages on public bulletin board and inserting advertisements on normal pages by sponsoring are frequently used hijacking methods. Hijacked links mislead link-based ranking algorithms which consider the link as human judgment about web pages. Hijacked pages could make a significant impact on ranking algorithms, since hijacked links are usually connected to a large number of spam farms where reputation of normal sites would leak out in large quantities.

In this paper, we propose a novel method for detecting web sites that are hijacked by spammers. Most of previous researches have focused on demoting or detecting spam, and as far as we know, there was no study on detecting link hijacking that is important in the following situations:

- In link-based ranking algorithms, we can reduce the weight of hijacked links. This will drop ranking scores of a large number of spam sites connected to hijacked sites, and improve the quality of search results.
- The hijacked sites will be continuously attacked by spammers (e.g. by repetitive spam comments on blogs), if their owners do not devise a countermeasures. By observing those hijacked sites, we can detect newly created spam sites promptly.
- Crawling spam sites is a sheer waste of time and resources. We can avoid collecting and storing numerous spam pages by stopping crawling at hijacked link.

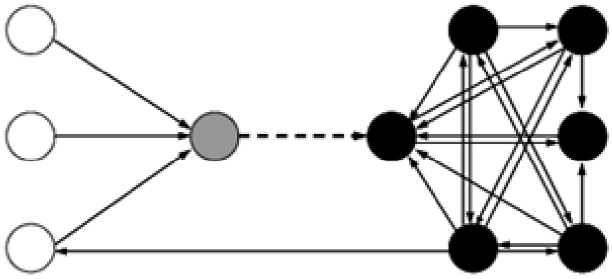


Figure 1 Link structure around a hijacked site.

In order to find out hijacked sites, we consider the characteristics of the link structure around hijacked site which is illustrated in Figure 1. White, gray and black nodes represent normal, spam and hijacked sites, respectively. A dashed link from the hijacked site to a spam site is a hijacked link. While a hijacked site has links pointing to spam sites, it is rarely pointed to by the spam sites because spam sites have few incentives to share link-based importance score with hijacked sites. Consequently, we can see a significant change in the link structure between spam and hijacked sites.

Suppose a walk starting from a spam site by following links backward. In the first few steps, we are in the middle of the spam farm, and we could see that visited sites are pointed to by many other spam sites. When we reach one of the hijacked sites, however, we would notice that the site is no longer pointed to by spam sites.

Such kind of changes in the link structure can be estimated by some modified versions of PageRank. For each page, we calculate white and spam scores using two different modified PageRank. Intuitively, these scores mean that the degree of trustworthiness and spamicity of a site. Hence, the spam score of a site in spam farms might overwhelm its white score, and the white score of a hijacked site might overwhelm its spam score. With this observation, we consider the inverse search of the Web graph from sample spam sites. We would find out hijacked sites during the walk where the order of the spam value and trust value is reversed.

We tested our method and evaluated the precision of it on large-scale graph of the Japanese Web archive including 5.8 million sites and 283 million links. The rest of this paper proceeds as follows. In Section 2, we review background knowledge for PageRank and link spamming. Section 3 introduces several approaches to detecting or demoting link spamming. Section 4 presents our method to detect hijacked sites. In Section 5, we report experimental result of our algorithm. Finally, we discuss result of our approach.

2. BACKGROUND

2.1 WEB GRAPH

Link-based ranking algorithms consider the entire Web as a directed graph. We can denote the Web as $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E})$, where $\boldsymbol{V}$ is the set of all node. v can be a page, site of host. $\boldsymbol{E}$ is a set of directed edges $<p,q>$. Each node has some incoming links(inlinks) and outgoing links(outlinks). $In(p)$ represents the set of nodes pointing to p(the in-neighbors of p) and $Out(p)$ is the set of nodes pointed to by p(the out-neighbors of p). We will use n to describe $\|\boldsymbol{V}\|$, the number of total nodes on the Web.

2.2 PAGERANK

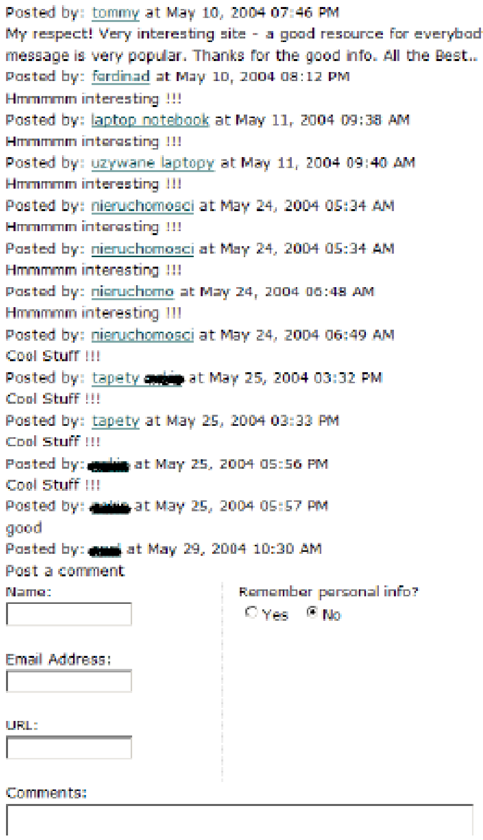
PageRank [3] is one of the most famous link-based ranking algorithms. The basic idea of PageRank is that a web page is important if it is linked by many other important pages. This recursive definition can be showed as following matrix equation:

$$\mathbf{p} = \alpha \cdot \mathbf{T} \cdot \mathbf{p} + (1 - \alpha) \cdot \mathbf{d}$$

Where $\mathbf{p}$ is PageRank score vector, $\mathbf{T}$ is transition matrix. $T(p, q)$ is $1/\|Out(q)\|$ if there is a link from node q to node p, and 0 otherwise. The decay factor $\alpha < 1$ (usually 0.85) is necessary to guarantee convergence and to limit the effect of rank sink. $\mathbf{d}$ is a uniformly random distribution vector. We can jump from a page to a random page chosen according to distribution $\mathbf{d}$ without following outlinks.

2.3 LINK SPAMMING

After the success of Google which adopted PageRank as the main ranking algorithm, PageRank became the main target of link spammers. Z. Gyöngyi et al. studied about link spam in [4] and introduced the optimal link structure to maximize PageRank Score, *spam farm*. A spam farm consists of a target page and boosting pages. All boosting pages link to a target page in order to increase the rank score of a target page. Then, a target page distributes its boosted PageRank score back to supporter pages. By this, members of a spam farm can boost their PageRank scores. Due to the low costs of domain registration and web hosting, spammers can create spam farms easily, and actually there exist spam farms with thousands of different domain names [9]. In addition to construct an internal link



Posted by: tommy at May 10, 2004 07:46 PM
My respect! Very interesting site - a good resource for everybody! Your message is very popular. Thanks for the good info. All the Best..
Posted by: ferdinad at May 10, 2004 08:12 PM
Hmmmmm interesting !!!
Posted by: laptop notebook at May 11, 2004 09:38 AM
Hmmmmm interesting !!!
Posted by: uzywane laptopy at May 11, 2004 09:40 AM
Hmmmmm interesting !!!
Posted by: nieruchomosci at May 24, 2004 05:34 AM
Hmmmmm interesting !!!
Posted by: nieruchomosci at May 24, 2004 05:34 AM
Hmmmmm interesting !!!
Posted by: nieruchomo at May 24, 2004 06:48 AM
Hmmmmm interesting !!!
Posted by: nieruchomosci at May 24, 2004 06:49 AM
Cool Stuff !!!
Posted by: tapety [redacted] at May 25, 2004 03:32 PM
Cool Stuff !!!
Posted by: tapety at May 25, 2004 03:33 PM
Cool Stuff !!!
Posted by: [redacted] at May 25, 2004 05:56 PM
Cool Stuff !!!
Posted by: [redacted] at May 25, 2004 05:57 PM
good
Posted by: [redacted] at May 29, 2004 10:30 AM
Post a comment
Name:
Remember personal info?
○ Yes ● No
Email Address:
URL:
Comments:

Figure 2 Spam comments on the blog

structure, spammers should create external links from outside of spam farms in order to provide additional PageRank score to the spam farm. We can see the real example of link hijacking in Figure 2.

To make links from non-spam sites to spam sites, spammers send trackbacks that lead to spam sites or, post comments including links pointing to target spam sites. In addition to posting spam comments or sending trackbacks, spammers can hijack links by various methods like creating pages that contain links to useful resource and links to target spam pages, or buying expired domains [4].

Because a large number of spam trackbacks and comments are created easily in a short period, link-based ranking algorithms like PageRank can be influenced seriously by link hijacking. Hijacked pages are hard to detect because their contents and domains are irregular [5].

3. RELATED WORK

Several approaches have been suggested in order to detect and demote link spam.

To demote spam pages and make PageRank resilient to link spamming, Gyöngyi et al. suggested TrustRank [6]. TrustRank introduced the concept of trust for web pages. In order to evaluate trust score of the entire Web, TrustRank assigns initial trust scores on some trust seed pages and propagates scores throughout the link structure. Wu et al. complemented TrustRank with topicality in [7]. They computed TrustRank score for each topic to solve the bias problem of TrustRank. Wu et al. also complemented TrustRank in [8] by propagating anti-trust from spam pages.

To detect link spam, Benczúr et al. introduced SpamRank [10]. SpamRank checks PageRank score distributions of all in-neighbors of a target page. If this distribution is abnormal, SpamRank regards a target page as a spam and penalizes it. Krishnan et al. proposed Anti-TrustRank to find out spam pages [11]. As the inverse-version of TrustRank, Anti-TrustRank propagates Anti-Trust score through inlinks from seed spam pages. Gyöngyi et al. suggested Mass Estimation in [9]. They evaluated spam mass, a measure of how many PageRank score a page get through links from spam pages. Saito et al. employed a graph algorithm to detect web spam [15]. They extracted spam seed from the strongly connected component (SCC) and used them to separate spam sites from non-spam sites. Becchetti et al. computed probabilistic counting over the Web graph to detect link spam in [19].

Some studies are done to optimize the link structure for fair ranking decision. Carvalho et al. proposed the idea of noisy links, the link structure that has a negative impact on the link-based ranking algorithms [12]. By removing these noisy links, they improved the performance of link-based ranking algorithm. Qi et al. also estimated the quality of links by similarity of two pages [13].

Du. et al. discussed the effect of hijacked links on the spam farm in [5]. They suggested an extended optimal spam farm by dropping the assumption of [4] that leakage by link hijacking is constant. Although they consider link hijacking, they did not mention the real features of hijacking and its detection, which is different from our approach.

As we reviewed, although there are various approaches to link spam, the link hijacking has never been explored closely. In this paper, we propose a new approach to discovering hijacked link and pages. With our approach, we would contribute to a new spam detection technique and improve the performance of link-based ranking algorithms.

4. DETECTING LINK HIJACKING

4.1 Core-based PageRank

To decide whether each page is a trustworthy page or a spam page, previous approaches used biased PageRank and biased inverse PageRank with white or spam seed set [6][11]. In this paper, we adopted a core-based PageRank proposed in [9]. When we have a seed set $\boldsymbol{S}$, we describe a core-based score of a page p as $\mathbf{PR}'(p)$. A core-based PageRank score vector $\mathbf{p}'$ is:

$$\mathbf{p}^{'} = \alpha \cdot \mathbf{T} \cdot \mathbf{p}' + (1-\alpha) \cdot \mathbf{d}^{\boldsymbol{S}}$$

where a random jump distribution $\mathbf{d}^{\boldsymbol{S}}$ is:

$$\mathbf{d}_p{}^{S} = \begin{cases} 1/n, \text{if } p \text{ is in seed set } \boldsymbol{S} \\ 0, \quad \text{otherwise} \end{cases}$$

We adopted a core-based PageRank instead of TrustRank because a core-based PageRank is independent on the size of a seed set compared to TrustRank which uses a random jump distribution of $1/\|\boldsymbol{S}\|$ instead of $1/n$.

In this paper, we use two types of core-based PageRank scores.

- $\mathbf{p}^{+}$ = a core-based PageRank score vector with a trust seed set $\mathbf{S}^{+}$.
- $\mathbf{p}^{-}$ = a core-based PageRank score vector with a spam seed set $\mathbf{S}^{-}$.

Z. Gyöngyi et al. mentioned a core-based PageRank with a spam seed set in [9]. They focused on blending $\mathbf{p}^{+}$ and $\mathbf{p}^{-}$(e.g. compute weighted average) in order to detect spam pages. However, this view is different from ours. We think $\mathbf{p}^{+}$ and $\mathbf{p}^{-}$ independently and focus on the change in scores through links to discover hijacked pages.

4.2 Link Hijacking Detection Algorithm

Based on the characteristics of links structure around hijacked pages, we observe the changes core-based PageRank score $\mathbf{PR}^{+}$ with white seeds and $\mathbf{PR}^{-}$ with spam seeds during an inverse graph traversal starting from spam seed sites.

As long as we are in a spam farm, the visiting site should have a high $\mathbf{PR}^{-}(q)$ and a low $\mathbf{PR}^{+}(q)$. When we reach at a hijacked site, it should have a lower $\mathbf{PR}^{-}(p)$ and a higher $\mathbf{PR}^{+}(p)$, since it is hardly pointed to by spam sites.

```
input : good seed set $S^{+}$, spam seed set $S^{-}$ , parameter $\delta$
output : set of hijacked sites of $H$

$H \leftarrow \emptyset$

Compute core-based PageRank score $\mathbf{PR}^{+}$ and $\mathbf{PR}^{-}$

for each site $s^{-}$ in $S^{-}$ do
    dfs($s^{-}$ , $H$)
end for

procedure dfs(s, $H$)
    if s is marked then
        return
    end if

    mark s
    if $\log(\mathbf{PR}^{+}(\mathbf{s})) - \log(\mathbf{PR}^{-}(\mathbf{s})) \geq \boldsymbol{\delta}$ then
        $H \leftarrow H \cup \{s\}$
        return
    end if

    for each site $t$ where $\{t | t \in In(s) \wedge \mathbf{PR}^{+}(s) < \mathbf{PR}^{+}(t)\}$
        dfs($t$, $H$)
    end for
end procedure
```

Figure 3 Link hijacking detection algorithm

By detecting this change in scores, we would find the hijacked sites. The algorithm is shown in Figure 3.

First, we compute $\mathbf{PR}^{+}(p)$ and $\mathbf{PR}^{-}(p)$ for each site p. Then start an inverse depth-first search from spam seed sites s^{-} whose scores are $\mathbf{PR}^{+}(\mathrm{s}^{-}) < \mathbf{PR}^{-}(\mathrm{s}^{-})$. The search from a site p is performed by selecting a site t whose scores are $\mathbf{PR}^{+}(p) < \mathbf{PR}^{+}(t)$. When it reached at a site q where scores are $\mathbf{PR}^{+}(q) > \mathbf{PR}^{-}(q)$, we output this site as a hijacked site, and stop the further search from this site.

We introduce parameter δ to adjust where we stop the search. When we use a higher δ value, a higher $\mathbf{PR}^{+}(p)$ score is required to stop the search, and we need a further search. When we use a lower δ value, we can stop the search earlier at a site with lower $\mathbf{PR}^{+}(p)$ score. δ can be modified from $-\infty$ to ∞.

5. EXPERIMENTS

5.1 Data set

To evaluate our algorithm, we performed experiments on a large-scale snapshot of our Japanese web archive built by a crawling conducted in May 2004. Basically, our crawler is based on breadth-first crawling [16], except that it focuses on pages written in Japanese. We collected pages outside the .jp domain if they were written in Japanese. We used a web site as a unit when filtering non-Japanese pages. The crawler stopped collecting pages from a site, if it could not find any Japanese pages on the site within the first few pages. Hence, this dataset contains fairly amount of English or other language pages. The amount of Japanese pages is estimated to be 60%. This snapshot is composed of 96 million pages and 4.5 billion links.

We use a site level graph of the Web, in which nodes are web sites and edges represent the existence of links between pages in different sites. The site graph built from our snapshot includes 5.8 million sites and 283 million links. We call this dataset web graph in this paper. Certain properties and its statistics of domains of our web graph are shown in Table 1 and 2.

Table 1 Properties of the web graph

Number of nodes	5,869,430
Number of arcs	283,599,786
Maximum of indegree (outdegree)	61,006 (70,294)
Average of indegree (outdegree)	48 (48)

Table 2 Domains in the web data

Domains	Numbers	Ratio(%)
.com	2,711,588	46.2
.jp	1,353,842	23.1
.net	436,645	7.4
.org	211,983	3.6
.de	169,279	2.9
.info	144,483	2.5
.nl, .kr, .us, etc.	841,610	14.3

5.2 Seed Set

To evaluate the trustworthiness and the spamicity of a site, we employed a core-based PageRank **PR**$^+$ and **PR**$^-$. Trust seed set and spam seed set are constructed for computation. We used manual and automated selection for both seed sets.

In order to generate a trust seed set, we computed PageRank score and performed a manual selection on top 1,000 sites with high PageRank score. Well-known sites (e.g. Google, Yahoo!, MSN and goo), authoritative university sites and well-supervised company sites are selected as white seed sites. After manual check, 389 sites are labeled as trustworthy sites. To make up for small size of a seed set, we extracted sites with specific URL including .gov (US governmental sites) and .go.ip (Japanese governmental sites). Finally, we have 40,396 sites as trust sites.

For spam seed set, we chose sites with high PageRank score and checked manually. Sites including many unrelated keywords and links, redirecting to spam sites, containing invisible terms and different domains for each menu are judged spam sites. We have 1,182 sites after manual check. In addition, we used automatically extracted seed sites obtained by analyzing strongly connected components and cliques [15]. Finally, Total 580,325 sites are used for a spam seed set.

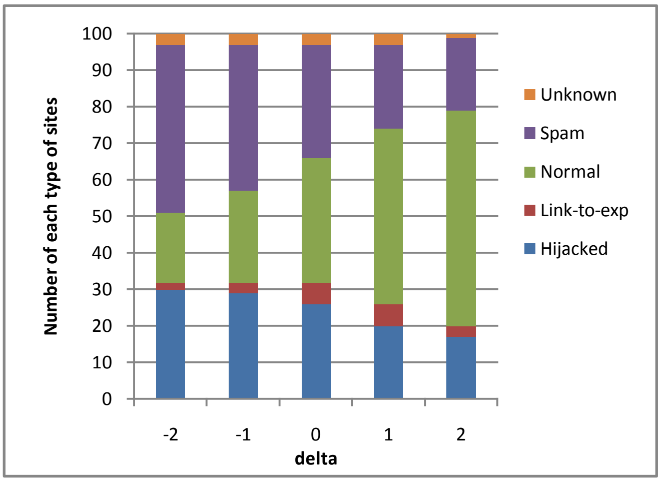


Figure 4 Number of sites of each type with different δ

5.3 Evaluation

Using the trust and spam seed sets, we extracted lists of potential hijacked sites with different δ values from -2.0 to 2.0. (See the algorithm in Section 4.2). After we had the lists, we sorted them in the descending order of Anti-TrustRank scores. We chose Anti-TrustRank since sites with high Anti-TrustRank scores tend to have many links to spam sites and consequently can be considered to be influential.

5.3.1 *Types of hijacking*

We first looked through several hundreds of sites in those lists, and investigated suspicious sites whether they are hijacked or not. As a result, we obtained various types of hijacking sites. In addition to well known link hijacking methods like spam comments, trackbacks and buying expired domains, spammers create links to their spam sites by accessing normal sites with public access statistics log that shows links to referrer sites. Spammers are also able to obtain a link from hosting company sites by being a client of companies. We regard normal sites with direct links to expired-domain hijacked sites as hijacked sites, since spammers employ sites with expired domains as a spam site.

5.3.2 *Precision of hijack detection*

Figure 4 shows the number of each hijacking type in the top 100 results using different δ values. We categorized detected samples into spam, normal, normal site with direct link to expired-hijacked sites, hijacked sites and finally, unknown. Sites written in unrecognizable languages such as Chinese, German and Italian were judged unknown.

We can find from 17 to 30 hijacked sites with different δ. We can detect the most hijacked sites (30 sites) with the lowest δ value. This means that hijacked sites tend to be judged to be spam sites, which means normal sites might take a disadvantage in the ranking due to link hijacking. In addition, we can find from 2 to 6 normal sites that point directly to hijacked sites. When we include these sites into hijacked sites, about 32% of sites in the top 100 results are related to link hijacking. Considering the difficulty of detecting hijacked sites with diverse contents and complex structure on the web, this is quite encouraging.

6. CONCLUSION

In this paper, we proposed a new method for link hijacking detection. Link hijacking is an essential method for link spamming and many hijacked links are now being generated by spammers. Since link hijacking could have a significant impact on link-based ranking algorithms and disturb assigning global importance of sites, detecting hijacked sites and penalizing hijacked links are the serious problems to be solved.

In order to find out hijacked sites, we focused on the characteristics of the link structure around the hijacked sites. Based on the observation that hijacked sites are seldom linked by spam sites while they have many links to spam sites, we computed two types of core-based PageRank scores and monitored the change in two scores during the inverse walk from spam seed. Experimental result showed that our approach is quite effective. Our best result for finding hijacked sites was 32%.

REFERENCES

[1] S. Nakamura, S. Konishi, A. Jatowt, H. Ohshima, H. Kondo, T. Tezuka, S. Oyama and K. Tanaka. "Trustworthiness Analysis of Web Search Results, " Proc. the 11th European Conference on Research and Advanced Technology for Digital Libraries, 2007.

[2] Z. Gyöngyi and H. Garcia-Molina. "Web spam taxonomy," Proc. the 1st international workshop on Adversarial Information Retrieval on the Web, 2005.

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

[4] Z. Gyöngyi and H. Molina. "Link Spam Alliance," Proc. the 31st international conference on Very large Data Bases, 2005.

[5] Y. Du, Y. Shi and X. Zhao. "Using spam farm to boost PageRank," Proc. the 3rd international workshop on Adversarial information retrieval on the Web, 2007.

[6] Z. Gyöngyi, H. Garcia-Molina and J. Pedersen. "Combating web spam with TrustRank," Proc. the 30th international conference on Very Large Data Bases, 2004.

[7] B. Wu, V. Goel and B. D. Davison. "Topical TrustRank: using topicality to combat web spam," Proc. the 15th international conference on World Wide Web, 2005.

[8] H. Yang, I. King and M. R. Lyu. "Diffusion Rank: a possible penicillin for web spamming," Proc. the 30th annual international ACM SIGIR conference on Research and development in information retrieval, 2007.

[9] B. Wu, V. Goel, and B. D. Davison. "Propagating trust and distrust to demote web spam," Proc. the WWW2006 Workshop on Models of Trust for the Web, 2006.

[10] Z. Gyöngyi, P. Berkhin, H. Garcia-Molina and J.Pedersen. "Link Spam Detection Based on Mass Estimation," Proc. the 32nd international conference on Very Large Data Bases, 2006.

[11] V. Krishnan and R. Raj. "Web spam detection with Anti-trustRank," Proc. the 2nd international workshop on Adversarial Information Retrieval on the Web, 2006.

[12] A. Benczúr, K. Csaloganym T. Sarlos, M. Uher. "SpamRank-fully automatic link spam detection," Proc. the 1st international workshop on Adversarial Information Retrieval on the Web, 2005.

[13] A. Carvalho, P. Chirita, E. Moura and P. Calado. "Site level noise removal for search engines," Proc. the 15th international conference on World Wide Web. 2006.

[14] X. Qi, L. Nie and B. D. Davison. "Measuring similarity to detect qualified links," Proc. the 3rd international workshop on Adversarial Information Retrieval on the Web, 2007.

[15] R. Guha, R. Kumar, P. Raghavan and A. Tomkins. "Propagation of trust and distrust," Proc. the 13th international conference on World Wide Web, 2004.

[16] H. Saito, M. Toyoda, M. Kitsuregawa and K. Aihara. "A large-scale study of link spam detection by graph algorithms ," Proc. the 3rd international workshop on Adversarial Information Retrieval on the Web, 2007.

[17] M. Najork and J. L. Wiener. "Breadth-first crawling yields high-quality pages," Proc. the 10th international conference on World Wide Web, 2001.

[18] P. Metaxas and J. DeStefano. "Web spam, propaganda and trust," Proc. the 1st international workshop on Adversarial Information Retrieval on the Web, 2005.

[19] L. Becchetti, C. Castillo, D. Donato, S. Leonardi and R. Baeza-Yates. "Using rank propagation and probabilistic counting for link-based spam detection," Technical report, DELIS, 2006.