Detecting Link Hijacking by Web Spammers

Young-joo Chung, Masashi Toyoda, and Masaru Kitsuregawa

No Institute Given

Since current search engines employ link-based ranking algorithms as an important tool to decide a ranking of sites, web spammers are making a significant effort to manipulate the link structure of the Web, so called, link spamming. Link hijacking is an indispensable technique for link spamming to bring ranking scores from normal sites to target spam sites. In this paper, we propose a link analysis technique for finding link hijacked sites using modified PageRank algorithms. We performed experiments on the large scale Japanese Web archive and evaluated the accuracy of our method. Detection precision of our approach was improved about 25% from a baseline approach.

Key words: Link analysis, Web spam, Information retrieval, Link hijacking

1 Introduction

In the last decade, search engines have been essential tools for information retrieval. People rely heavily on these tools to find information on the Web, and as a result, most Web sites get a considerable number of visitors via 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 results in the list [1], obtaining a high ranking in the search results became crucial for the success of sites.

Web spamming is the behavior that manipulates web page features to get a higher ranking than the page deserves. Web spamming techniques can be categorized into term spamming and link spamming. *Term spamming* manipulates textual contents of pages by repeating specific keywords that are not related with page contents and by adding irrelevant meta-keywords or anchor text. Search engines which employ textual relevance to rank pages will return these manipulated pages at the top of the result list. *Link spamming* manipulates the link structure of the Web to mislead link-based ranking algorithms such as PageRank [4]. Link-based ranking algorithms consider a link as an endorsement for pages. Therefore, spammers create numerous false links and construct an artificially interlinked link structure, so called a spam farm, where all pages link to the target spam page in order to centralize its link-based importance.

Links from external normal pages to spam pages are needed in order to attract the attention of search engines and feed ranking scores to spam farms. These links that are created without any agreements of page owners are called *hijacked link*. To hijack link, spammers post comments including URLs to spam pages on public bulletin boards, buy expired domains and sponsor pages. Hijacked

links affect link-based ranking algorithms significantly, when they are pointing to spam farms containing a large amount of spam pages.

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

- Hijacked sites are prone to be attacked continuously by various spammers (e.g. by repetitive spam comments on blogs). Observing such sites will be helpful for the prompt detection of newly created spam sites that might not be filtered by existing anti-spam techniques. Since spam detection has been an arms race, it is important to find out sites with new spamming methods.
- Once we detect hijacked sites, we can modify link-based ranking algorithms to reduce the importance of newly created links from hijacked pages in those sites. It makes the algorithms robust to newly created spam. Though it might temporally penalize links to normal sites, we can correct their importance after we invent spam detection methods for novel spamming techniques.
- Crawling spam sites is a sheer waste of time and resources. Most crawlers have spam filters, but such filters cannot quickly adapt themselves to new spamming methods. By reducing the crawling priority of new links from hijacked pages in detected sites, we can avoid collecting and storing new spam sites, until spam filters are updated.

In order to identify hijacked sites, we consider characteristics of the link structure around hijacked sites. As Figure 1 indicates, hijacked sites are supposed to have a certain number of links to both normal and spam sites, and exist at the boundary of them. To detect this boundary, we take account of trustworthiness and spamicity of whole sites. Normal sites would have high trustworthiness and low spamicity, and in contrast, spam sites would have low trustworthiness and high spamicity. These relations will be reversed at the link between normal sites and spam sites, or where link hijacking occurs. Based on this idea, we detect the point where trustworthiness and spamicity are reversed in order to extract hijacked sites.

In addition, we focus on the fact that hijacked sites have links pointing to both normal and spam sites. Out-neighbors of normal sites will show much more trustworthiness than spamicity, and vice versa. Thus, it would be assumed that overall trustworthiness and spamicity in out-neighbors of hijacked sites are equivalent compared to those of normal or spam.

Trustworthiness and spamicity of a site can be evaluated by some link-based ranking algorithms such as modified versions of PageRank. For each site, we calculate white and spam scores using two different modified PageRanks. Intuitively, these scores represent the degree of trustworthiness and spamicity of sites.

The rest of this paper proceeds as follows. In Section 2, we review background knowledge for PageRank and link spamming. Section 3 introduces modified PageRank algorithms and several approaches to detect or demote link spamming. Section 4 presents our method for detecting hijacked sites. In Section 5,

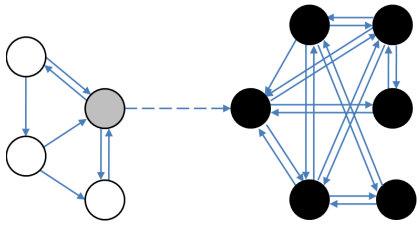

Fig. 1. Link structure around a hijacked site. White, gray, and black nodes represent normal, hijacked and spam sites respectively. A dashed link from the hijacked site to a spam site is the hijacked link.

we report the experimental result of our algorithm. Finally, we conclude and summarize the result of our approach.

2 Background

2.1 Web Graph

The entire Web can be considered as a directed graph. We can denote the Web as $G = (V, E)$, where V is the set of nodes and E is a set of directed edges $< p, q >$. Node v can be a page, host or site.

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 $\|V\|$, the number of total nodes on the Web.

2.2 PageRank

PageRank [4] is one of the most well-known 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:

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

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 an effect of rank sink. $\mathbf{d}$ is a uniformly distributed random vector. Instead of following links to next pages, we can jump from a page to a random one chosen according to distribution $\mathbf{d}$.

2.3 Link Spamming

After the success of Google which adopted PageRank as the main ranking algorithm, PageRank became a primary target of link spammers. Z. Gyöngyi et

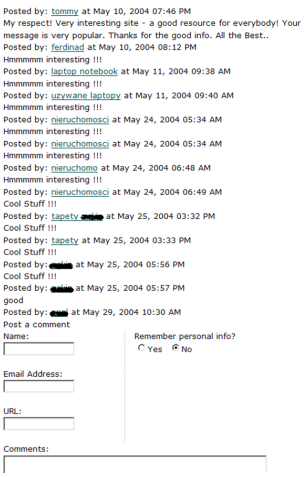

Fig. 2. Spam comments on blog.

Fig. 3. Link hijacking by advertisement.

al. studied about link spam in [6] and introduced an optimal link structure to maximize PageRank score, a *spam farm*. The spam farm consists of a target page and boosting pages. All boosting pages link to the target page in order to increase the rank score of it. Then, the 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 [10].

In addition to constructing the internal link structure, spammers make external links from outside of spam farms to attract search engines and provide PageRank scores to the target page. To make links from non-spam pages to spam pages, various hijacking techniques are exploited. Spammers send trackbacks that lead to spam sites, or post comments including links pointing to spam pages. Expired domains can be bought by spammers, and then changed to spam sites. Spammers can also sponsor Web sites to insert advertisements of spam sites on their pages.

Examples of link hijacking are shown in Figure 2 and 3. Figure 2 illustrates spam comments on blogs. Spammers post comment containing a large amount of links to spam sites. By this, they can create massive links to spam sites easily in a short period. Figure 3 shows a page containing many links to spam pages. Although it is the page about java and linux, it contains links pointing to a lot of sport sites which are densely connected together. Note that major search engines and blog services employ counter-measures like `rel="nofollow"` tags, which is attached to hyperlinks that should be ignored by link-based ranking algorithms [15]. However, there still exist a number of Web services that do

not support such means, and hijacking techniques like buying expired domains cannot be penalized by `"nofollow"` tag.

3 Previous Work

3.1 TrustRank and Anti-TrustRank

To improve the PageRank algorithm, Gyöngyi et al. presented the TrustRank algorithm [8]. The basic intuition of TrustRank is that good pages seldom link to spam pages. People trust good pages, and can trust pages pointed to by good pages. Like this, trust can be propagated through the link structure of the Web. Therefore, in TrustRank, a list of highly trustworthy pages is created as a seed set and each of these pages is assigned a non-zero initial trust score, while all the other pages on the Web have initial values of 0. After computation, good pages will get a decent trust score, and spam pages get a lower trust scores.

The matrix notation of TrustRank is following:

$$\mathbf{t} = \alpha \cdot \mathbf{T} \times \mathbf{t} + (1 - \alpha) \cdot \mathbf{d}^{\tau}$$

where $\mathbf{t}$ is TrustRank score vector, α is decay factor(0.85), and $\mathbf{d}^{\tau}$ is a random jump distribution vector where

$$d_p{}^{\tau} = \begin{cases} 1/\|S\|, & \text{if } p \text{ is in trust seed set } S \\ 0, & \text{otherwise} \end{cases} .$$

Krishnan et al. proposed Anti-TrustRank to find out spam pages [11]. Instead of selecting good pages as a seed set, Anti -TrustRank starts score propagation from spam seed pages. Each spam page is assigned Anti-Trust score and this score is propagated along incoming links.

3.2 Core-based PageRank

Core-based PageRank was suggested by Gyöngyi et al. [10]. When we have a seed set S, we describe a core-based PageRank score of a page p as $\mathbf{PR}'(p)$. A core-based PageRank score vector $\mathbf{p}'$ is :

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

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

$$d_p{}^{\nu} = \begin{cases} 1/n, & \text{if } p \text{ is in seed set } S \\ 0, & \text{otherwise} \end{cases} .$$

Core-based PageRank is different from TrustRank by random jump vector. Core-based PageRank adopts a random jump distribution $1/n$, which is normalized by the number of whole Web site, instead of $1/\|S\|$.

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

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

Z. Gyöngyi et al. mentioned a core-based PageRank with a spam seed set in [10]. They refer to blending $\mathbf{PR}^+$ and $\mathbf{PR}^-$ (e.g. compute weighted average) in order to detect spam pages. However, this view is different from ours. We think $\mathbf{PR}^+$ and $\mathbf{PR}^-$ separately and focus on the change in scores through links to discover hijacked links.

3.3 Other Approaches

Several approaches have been also suggested for the purpose of detecting and demoting link spam.

To demote spam pages and make PageRank resilient to link spamming, Wu et al. complemented TrustRank with topicality in [9]. They computed TrustRank score for each topic to solve a bias problem of TrustRank.

To detect link spam, Benczur et al. introduced SpamRank [12]. SpamRank checks PageRank score distributions of all in-neighbors of a target page. If this distribution is abnormal, SpamRank regards a target page as spam and penalizes it. Gyöngyi et al. suggested Mass Estimation in [10]. They evaluated *spam mass*, a measure of how many PageRank scores a page gets through links from spam pages. Saito et al. employed a graph algorithm to detect Web spam [13]. They extracted spam pages by the strongly connected component decomposition and used them as a seed set to separate spam pages from non-spam pages.

Du. et al. discussed an effect of hijacked links on the spam farm in [7]. They introduced an extended optimal spam farm by dropping the assumption of [6] that leakage by link hijacking is constant. Although Du. et al. considered link hijacking, they did not studied features of hijacking and its detection, which is different from our work.

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

4 Link Hijacking Detection

Based on characteristics of the change in trustworthiness and spamicity around hijacked sites, we compute a hijacked score of sites.

To begin with, we assign white score and spam score to whole Web sites. We employ the notion of $\mathbf{White}(p)$ and $\mathbf{Spam}(p)$, which represent the degree of trustworthiness of site p and the degree of spamicity of site p, respectively. For example, TrustRank score and core-based PageRank score with a white seed set can be used as white score. Anti-TrustRank score and core-based PageRank score with spam seeds are available for spam scores.

With these scores, we define the *score reversal* relation. Hijacked site p would have a higher white score than its spam score, and spam site would have lower white score than its spam score. These features can be described with *relative trust*, **RT**.

$$\mathbf{RT}(p) = \log\big(\mathbf{White}(p)\big) - \log\big(\mathbf{Spam}(p)\big) - \delta \ .$$

We used log value since PageRank score obeys power law distribution. If the log difference between white score and spam score of site p is bigger than threshold δ, p is more likely to be normal. In contrast, $\mathbf{RT}(p)$ lower than 0 implies p might be spam. When we use a higher δ value, we consider the white score of hijacked sites is much higher than its spam score. Therefore, our algorithm will choose a site nearer to normal side as a hijacked. When we use a lower δ value, we regard the white score of a hijacked site is lower, so select sites close to spam side as hijacked.

As we see in Section 1, if p is hijacked, there must be spam sites in out-neighbors of p by definition. In addition to this, we take account of only sites with a lower white score and higher spam score than those of p. With this condition, we can check sites that are more likely to be spam than p is. We will call this relation *score reversal*.

With the score reversal relation, we define the hijacked score. First, we create $R(p)$, a set of sites which have the score reversal relation with site p.

$$R(p) = \left\{ r \;\middle|\; \begin{array}{l} r \in Out(p) \wedge \mathbf{RT}(r) < 0 \ \ \wedge \\ \mathbf{White}(r) < \mathbf{White}(p) \wedge \mathbf{Spam}(r) > \mathbf{Spam}(p) \end{array} \right\} .$$

Then, we define a set H of hijacked candidates. A hijacked site h will have higher trustworthiness than its spamicity, and will have at least one out-neighbor node that is in the score reversal relation with it.

$$H = \big\{ h \;\big|\; \mathbf{RT}(h) \geq 0 \ \wedge \ R(h) \ \neq \phi \big\} \ .$$

Finally, we compute the hijacked score of h. Two different hijacked detection methods are designed.

As a naive approach, we considered only the total trustworthiness difference between a hijacked candidate and sites in the score reversal relation with it. This can be described as following:

$$\mathbf{H}_{rev}(h) = \sum_{r \in R(h)} \big\{ \log\big(\mathbf{White}(h)\big) - \log\big(\mathbf{White}(r)\big) \big\} \ .$$

We found out that when both trustworthiness and spamicity of around a hijacked site are considered, a higher detection precision can be achieved. Overall out-neighbor trustworthiness of a hijacked site will be obtained by the average **RT** of total normal outnodes. In the same manner, spamicity of out-neighbors will be computed by the average $-\mathbf{RT}$ of all spam outnodes. To obtain out-neighbor trustworthiness and spamicity of a hijacked site, we divide $Out(h)$ into

a set of normal sites $nOut(h)$ and a set of spam sites $sOut(h)$.

$$nOut(h) = \left\{ n \;\middle|\; n \in Out(h) \wedge \mathbf{RT}(n) \geq 0 \right\} ,$$
$$sOut(h) = \left\{ s \;\middle|\; s \in Out(h) \wedge \mathbf{RT}(s) < 0 \right\} .$$

The following is the improved hijacked score $\mathbf{H}_{all}(h)$.

$$\mathbf{H}_{all}(h) = \frac{\sum_{n \in nOut(h)} |\mathbf{RT}(n)|}{\|nOut(h)\| + \lambda} \cdot \frac{\sum_{s \in sOut(h)} |\mathbf{RT}(s)|}{\|sOut(h)\| + \lambda} .$$

While λ is a smoothing factor which alleviates an effect of **RT** when the number of normal or spam out-neighbors is very small.

$\mathbf{H}_{all}(h)$ increases as both trustworthiness and spamicity grow. When either trustworthiness or spamicity is getting lower, $\mathbf{H}_{all}(h)$ decreases since site h seems to be a spam or normal site.

5 Experiments

To test our method, we prepared a large Web data set. White seed set and spam seed set were also generated to compute white and spam scores. As for computations of white and spam scores, we used a core-based PageRank score $\mathbf{PR}^{+}$ and $\mathbf{PR}^{-}$ pair. With white and spam scores, we obtained two kinds of hijacked scores and evaluated the precisions of results.

5.1 Data Set and Seed 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 [14], except that it focuses on pages written in Japanese. Pages outside the .jp domain are collected if they were written in Japanese. We used a 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, our dataset contains fairly amount of pages in English or other languages. The percentage of Japanese pages is estimated to be 60%. This snapshot is composed of 96 million pages and 4.5 billion links.

We use an unweighted site level graph of the Web, in which nodes are Web sites and edges represent the existence of links between pages in different sites. In the site graph, we can easily find dense connections among spam sites that cannot be found in the page level graph. To build the site graph, we first choose the representative page of each site that has 3 or more inlinks from other sites, and whose URL is within 3 tiers (e.g. http://A/B/C/). Then, pages below each representative page are contracted to one site. Finally, edges between two sites are created when there exist links between pages in these sites. The site graph built from our snapshot includes 5.8 million sites and 283 million links. We call this dataset Web graph in our experiments.

To compute white and spam scores, we constructed trust seed set and spam seed set. We used manual and automated selecting methods for both seed sets.

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

For the spam seed set, we choose 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 a manual check. In addition, we used automatically extracted spam sites obtained by [13]. Saito et al. obtained this large spam seed set by following steps. First, they extracted strongly connected components (SCC) from the Web graph. Since spam sites tend to construct a densely connected link structure, it could be assumed that spam sites form SCC. In addition to detecting spam located at the fringe of the Web, Saito et al. counted up maximal cliques in the largest SCC, or a core. Cliques whose sizes are less than 40 were extracted from the core and about 8,000 spam sites were obtained. Finally, they used these spam sites as a reliable spam seed set and expanded them by a minimum cut technique that separates links between spam and non-spam sites. Since this method showed a high extraction precision, we used their spam sites as seed sites. Finally, Total 580,325 sites are used as a spam seed set.

5.2 Types of Hijacking

In order to understand a layout of sites at the boundary of normal and spam, we randomly choose 1,392 sites near to spam seeds. These samples are checked by hand and classified into 4 categories; hijacked, normal, spam and unknown. Unknown sites were written in unrecognizable languages such as Chinese, Dutch, German and so on. Table 1 shows the result of classification. The 33% of total sites was identified as hijacked, and these 465 sites are divided into 8 types as follows.

- Blog sites with spam comments or trackbacks and public bulletin boards containing comments pointing to spam sites.
- Expired sites bought by spammers. Spammers can buy expired domains and use them for spam sites. Since Web sites tend to maintain links pointing to expired domains for a while, spammers are able to get links from them.
- Hosting sites that include spam sites of some customers.
- Normal sites that point to expired hijacking sites. Expired hijacking sites are turned into spam sites by spammers, so links from normal to these expired sites can be considered hijacked links.

– Free link registration sites that allow spammers to register links on them.
– Normal sites that create links to spam sites by mistakes. Authors of some sites make links pointing to spam sites by themselves. Since it is hard for non-experts to identify spam sites, they believe those spam sites are useful.
– Normal sites that contain advertising links pointing to spam sites. Spammers can insert links on normal sites by sponsoring them.
– Sites with public access statistics that show links to referrers. Spammers access such sites frequently, and then plant links to spam sites in the referrer list.

Table 2 shows the number of sites in each hijacking type. We can see that the most frequently used technique is blog and bbs hijacking. Expired hijacking is a quite popular technique among spammers, too. Particularly, domains for official sites of movies and singers are prone to be hijacked because they are used for a while, not permanently.

5.3 Evaluation

Using the white and spam seed sets, we computed core-based PageRank scores for white and spam scores. Hijacked scores were obtained as well, with different δ values. (See Section 4.)

Precision with a naive approach For δ from -2 to $+2$, we choose top 200 sites with high $\mathbf{H}_{rev}$ scores and checked them by hand. Detected samples are categorized into hijacked, normal, spam, and unknown. The detail is shown in Table 3. The best precision 42.5% was obtained when δ was 1.

Precision with a suggested approach With different δ values from -5.0 to 1.0, we computed $\mathbf{H}_{all}$ score and evaluated top 200 sites. In order to determine smoothing factor λ, we calculated $\mathbf{H}_{all}$ score of sample sites mentioned in Section 5.2 and examined the result. Since the best precision for top 200 sites was obtained when $\lambda = 40$, we used the same value for the whole sites evaluation.

Table 1. Types of evaluation sample site

Site type	Number of sites
Hijacked	465
Normal	345
Spam	576
Unknown	6
Total	1392

Table 2. Types of hijacking

Hijacking type	Number of sites
Blog and bbs	117
Expired sites	77
Hosting sites	64
Link to expired site	60
Link register sites	55
Link to spam by mistake	50
Advertisement to spam	32
Server statistics	10
Total	465

Table 3. Top 200 precision with a naive approach.

δ	-2	-1	0	1	2
Hijacked	67	74	83	85	72
Normal	14	22	42	69	97
Spam	112	97	70	42	27
Unknown	7	7	5	4	4
Total	200	200	200	200	200
Precision	33.5%	37%	41.5%	42.5%	36%

Table 4. Top 200 precision with a suggested approach

δ	-5	-4	-3	-2	-1	0	1
Hijacked	84	100	106	135	132	132	114
Normal	6	8	10	12	16	31	41
Spam	110	91	81	50	48	33	42
Unknown	0	1	3	3	4	4	3
Total	200	200	200	200	200	200	200
Precision	42%	50%	53%	67.5%	66%	66%	57%

The result is described in Table 4. We detected hijacked sites with the best precision of 67.5% when δ is -2.

As both Table 3, 4 indicate, normal sites increase as δ increases. This is because with a higher δ, hijacked sites should have a higher white score. Likewise, as δ decreases, the proportion of spam sites increases. This means our algorithms become tolerant and consider sites with a relatively high spam score as hijacked. As for λ, we found out that as λ increases, $\mathbf{H}_{all}$ of spam sites decreases. However, if the value of λ exceeds 40, the number of spam sites in the top result hardly changes. The ratio of normal sites with high $\mathbf{H}_{all}$ remain stable regardless of λ.

We computed $\mathbf{H}_{rev}$ score with a TrustRank and Anti-TrustRank score pair and investigated the performance. However, the precision was far worse than that with a core-based PageRank pair.

6 Conclusion

In this paper, we proposed a new method for link hijacking detection. Link hijacking is one of the essential methods for link spamming and massive hijacked links are now being generated by spammers. Since link hijacking has a significant impact on link-based ranking algorithms, detecting hijacked sites and penalizing hijacked links is serious a problem to be solved.

In order to identify hijacked sites, we focused on characteristics of the link structure around hijacked sites. Based on the observation that white and spam score reversal occurs between hijacked sites and hijacking sites, we computed hijacked scores.

Experimental results showed that our approach is quite effective. Our best result for finding hijacked sites outperformed about 25% compared to a naive approach.

References

1. Nakamura, S.,Konishi, S.,Jatowt, A., Ohshima, H., Kondo, H., Tezuka, T., Oyama, S., Tanaka, K.: Trustworthiness Analysis of Web Search Results. In: 11th European Conference on Research and Advanced Technology for Digital Libraries. Budapest, Hungary (2007)
2. Ntoulas, A., Najork, M., Manasse, M. Fetterly,D.: Detecting Spam Web pages through Content Analysis. In: 15th International Conference on World Wide Web. Edinburgh, Scotland, UK (2006)
3. Fetterly, D., Manasse, M., Najork, M.: Spam, Damn Spam, and Statistics: Using Statistical Analysis to Locate Spam Web Pages. In: 7th International Workshop on the Web and Databases. Paris, France (2005)
4. Page, L., Brin, S., Motwani, R., Winograd, T.: The PageRank citation ranking: Bringing Order to the Web. Technical report,Stanford Digital Library Technologies Project, Stanford University, Stanford, CA, USA (1998)
5. Gyöngyi, Z., Garcia-Molina, H.: Web Spam Taxonomy. In: 1st International Workshop on Adversarial Information Retrieval on the Web, Chiba, Japan (2005)
6. Gyöngyi, Z., Garcia-Molina, H.: Link Spam Alliance. In : 31st International Conference on Very large Data Bases, Trondheim, Norway (2005)
7. Du, Y., Shi, Y., Zhao, X.: Using Spam Farm to Boost PageRank. In : 3rd International Workshop on Adversarial Information Retrieval on the Web. Banff, Alberta, Canada (2007)
8. Gyöngyi, Z., Garcia-Molina, H., Pedersen, J.: Combating Web spam with TrustRank. In : 30th International Conference on Very Large Data Bases. Toronto, Canada (2004)
9. Wu, B., Goel, V., Davison, B. D.: Topical TrustRank: Using Topicality to Combat Web Spam. In : 15th International Conference on World Wide Web. Edinburgh, Scotland, UK. (2006)
10. Gyöngyi, Z., Berkhin, P., Garcia-Molina, H., Pedersen, J.: Link Spam Detection Based on Mass Estimation. In : 32nd international conference on Very Large Data Base. Seoul, Korea (2006)
11. Krishnan, V., Raj, R.: Web Spam Detection with Anti-TrustRank. In : 2nd International Workshop on Adversarial Information Retrieval on the Web. Edinburgh, Scotland, UK. (2006)
12. Benczur, A., Csalogány, K., Sarlós, T., Uher, M.: SpamRank-fully automatic link spam detection. In : 1st International Workshop on Adversarial Information Retrieval on the Web. Chiba, Japan (2005)
13. Saito, H., Toyoda, M., Kitsuregawa, M., Aihara, K.: A Large-scale Study of Link Spam Detection by Graph Algorithms. In : 3rd International Workshop on Adversarial Information Retrieval on the Web. Banff, Alberta, Canada (2007)
14. Najork, M., Wiener, J. L.: Breadth-first Crawling Yields High-quality Pages. In : 10th international conference on World Wide Web, Hong Kong, Hong Kong (2001)
15. The Official Google Blog, `http://googleblog.blogspot.com/2005/01/preventing-comment-spam.html`