PAPER *Special Section on Info-Plosion*

A Method for Detecting Hijacked Sites by Web Spammer using Link-based Algorithms*

Young-joo CHUNG[†a)], **Masashi TOYODA**[†b)], ***Nonmembers***, ***and*** **Masaru KITSUREGAWA**[†c)], ***Member***

SUMMARY In this paper, we propose a method for finding web sites whose links are hijacked by web spammers. A hijacked site is a trustworthy site that points to untrustworthy sites. To detect hijacked sites, we evaluate the trustworthiness of web sites, and examine how trustworthy sites are hijacked by untrustworthy sites in their out-neighbors. The trustworthiness is evaluated based on the difference between the white and spam scores that calculated by two modified versions of PageRank. We define two hijacked scores that measure how likely a trustworthy site is to be hijacked based on the distribution of the trustworthiness in its out-neighbors. The performance of those hijacked scores are compared using our large-scale Japanese Web archive. The results show that a better performance is obtained by the score that considers both trustworthy and untrustworthy out-neighbors, compared with the one that only considers untrustworthy out-neighbors.

key words: *Link analysis, Web spam, Information retrieval, Link hijacking*

1. Introduction

In 2008, Google found one trillion URLs on the Web [1]. It is almost impossible to find necessary information from such a huge web space without search engines. Since approximately a half of search engine users look at no more than the first five results in the list [2], web sites need to get high rankings to attract visitors and yield profits. Given this situation, it is not surprising that web spammers appeared who try to boost the rankings of their sites using unfair ways.

Web spammers generally use two main techniques; 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 texts. *Link spamming* manipulates the link structure of the Web to mislead link-based ranking algorithms such as PageRank [5]. Since such algorithms consider a link as an endorsement to target pages, spammers construct *spam farms* [6], sets of densely inter-linked web sites, with a purpose of centralizing link-based importance scores to target spam sites

It is necessary for spammers to create links from reputable sites to their spam farms, since isolated spam farms hardly attract the attention of search engines and bring ranking scores to themselves. A link from a normal site to spam that is created without any agreement of the author of the normal site is called a *hijacked link*. Spammers can create hijack links by posting comments with links to their spam sites on public bulletin boards, by buying expired domains, and by sponsoring web sites. These hijacked links significantly affect link-based ranking algorithms when they are pointing to large spam farms.

In this paper, we propose a new method for detecting hijacked web sites. 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 appeared 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 sites attacked by new spamming methods.
- Once we detect hijacked sites, we can modify link-based ranking algorithms to reduce the importance of newly created links on hijacked pages in those sites. It makes the algorithms robust to new spam. This might penalizes links to normal sites temporally, but we can correct their importance after spam detection methods for new spamming techniques are invented.
- 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.

To identify hijacked sites, we consider characteristics of the *trustworthiness* of a hijacked site and its out-neighboring sites. Suppose that there is a path be-

[†]Institute of Industrial Science, The University of Tokyo. 4-6-1 Komaba, Meguro-ku, Tokyo, 153-8505, Japan

*This is an extended version of the paper that appeared in Proceedings of the 13th Pacific-Asia Conference on Knowledge Discovery and Data Mining, Bangkok, Thailand, 27-30 April 2009.

a) E-mail: chung@tkl.iis.u-tokyo.ac.jp
b) E-mail: toyoda@tkl.iis.u-tokyo.ac.jp
c) E-mail: kitsure@tkl.iis.u-tokyo.ac.jp

DOI: 10.1587/transinf.E0.D.1

tween normal and spam sites. As we walk through that path, the trustworthiness of the site on each step is expected to decrease, and at a certain site, it would become lower than some threshold. This occurs when a normal site points to spam sites. This means the normal site is possibly hijacked by the spam sites.

We evaluate the trustworthiness of a site using two modified versions of PageRank that calculate white and spam scores of the site. The white score is propagated only from normal seed sites, and the spam score is propagated only from spam seed sites. We consider a site is trustworthy when it has a high white score and a low spam score, and vice versa. In other words, the trustworthiness is the difference between the white and spam scores of a site. We define two hijacked scores that measure how likely a trustworthy site is to be hijacked based on the distribution of the trustworthiness in its out-neighbors.

The performance of those hijacked scores are compared using our large-scale Japanese Web archive. The results show that a better performance is obtained by the score that considers both trustworthy and untrustworthy out-neighbors, compared with the one that only considers untrustworthy out-neighbors. Then, we categorize hijacked sites into several types and track outgoing links of hijacked sites to check if we can find new spam sites. We also compare two different pairs of the white and spam scores.

The rest of this paper is organized as follows. In Section 2, we review the background knowledge of PageRank and link spamming. Section 3 introduces modified PageRank algorithms and several approaches to detecting or demoting link spamming. Section 4 presents our method for detecting hijacked sites. In Section 5, we report the experimental results. Finally, we conclude and summarize our work in Section 6.

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 [5] 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 shown 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 a node q to a 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 the 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 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.

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.

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. 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 are assigned zero values. As a result, good pages will get a higher trust score, and spam pages get a lower trust score.

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

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

Krishnan et al. proposed Anti-TrustRank to find spam pages [11]. Anti-TrustRank starts score propagation from spam pages instead of good ones. Each spam seed 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]. Core-based PageRank score vector $\mathbf{p}'$ is :

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

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

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

Core-based PageRank is different from TrustRank by the 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 a 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 the 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 in 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 *spam mass*, a measure of how many PageRank scores a page gets through links from spam pages in [10]. Saito et al. employed a graph algorithm to detect web spam [13]. They extracted spam hosts by strongly connected component decomposition and used them as a seed set to separate spam hosts from non-spam hosts.

Du. et al. discussed an effect of hijacked links on the spam farm in [7]. They introduced an extended version of the optimal spam farm. They mentioned the assumption of [6] that leakage by link hijacking is constant might be dropped. Although Du. et al. considered link hijacking, they did not study 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 the change in the trustworthiness of a hijacked site and its out-neighboring sites, we define a hijacked score.

To measure the trustworthiness of a site, we use the white and spam scores of the site. As the white score, we can use TrustRank, and core-based PageRank calculated with a white seed set. As the spam score, we can use Anti-TrustRank, and core-based PageRank calculated with spam seed sites.

Based on the white and spam scores, we define the trustworthiness of a site as *relative trust* $\mathbf{RT}$ that is given by:

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

where $\mathbf{RT}(p)$, $\mathbf{White}(p)$, $\mathbf{Spam}(p)$ represent a relative trust of p, a white score, and a spam score, respectively. If $\mathbf{RT}(p)$ is higher than zero, p is more likely to be normal. In contrast, if $\mathbf{RT}(p)$ is lower than zero, p is more likely to be spam.

Log values of white and spam scores are used because PageRank scores obey the power law distribution. A threshold δ is introduced to reduce the impact caused by the different sizes of seed sets for the white and spam score computation. Modified PageRank algorithms assign the initial score only to seed sites so that the total amount of scores for propagation differs by the number of seed sites. As a result, a normal site s could have

a lower $\mathbf{White}(s)$ than $\mathbf{Spam}(s)$, when the number of white seed sites is much smaller than that of spam seed sites. To solve this problem, we adjust the δ value. If we use a positive δ value, we consider $\mathbf{White}(s)$ of a normal site s is higher than its $\mathbf{Spam}(s)$. On the other hand, when we use a negative δ value, we consider a normal site could have a lower $\mathbf{White}(s)$ than its $\mathbf{Spam}(s)$. In practice, the δ value will be adjusted around zero to obtain the best performance.

Using $\mathbf{RT}$, the out-neighbors of a hijacked site p can be divided into a set of normal-like out-neighbors $nOut(p)$ and a set of spam-like out-neighbors $sOut(p)$.

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

Then, we can create a set H of hijacked candidates. A hijacked site h would be a trustworthy site and have at least one out-neighboring site that has a negative $\mathbf{RT}$ value, and has a lower white score and a higher spam score than h.

$$H = \left\{ h \;\middle|\; \mathbf{RT}(h) \geq 0 \ \wedge \ R(h) \ \neq \phi \right\},$$

where $R(h)$ is:

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

For each hijacked candidate h, we calculate the hijacked score. Two different hijacked scores are designed.

First, we focus on spam-like out-neighbors of a hijacked site. This is based on the assumption that a hijacked site would have many spam out-neighbors by the attack from many different spammers. Therefore, we make the hijacked score grow as the average of $|\mathbf{RT}|$ of sites in $sOut(h)$ grows. Hijacked score $\mathbf{H}_s$ can be described as following:

$$\mathbf{H}_s(h) = \frac{\sum_{s \in sOut(h)} |\mathbf{RT}(s)|}{\|sOut(h)\| + \lambda},$$

where λ is a penalty parameter that penalizes the effect caused by the small number of out-neighbors. Without λ, a site that has small spam out-neighbors is more likely to obtain a higher hijacked score. This is not desirable because we try to find a site that is hijacked by many spam sites.

Second, we consider both normal-like and spam-like out-neighbors of a hijacked site. It can be assumed that a hijacked site points to normal sites as well as spam sites, since it is originally normal. Based on this, the average $\mathbf{RT}$ of both normal-like and spam-like out-neighbors is used for the hijacked score calculation. A weight parameter γ is introduced so that we can adjust the influence of normal and spam out-neighbors. The following is the second hijacked score $\mathbf{H}_{ns}(h)$.

$$\mathbf{H}_{ns}(h) = \left(\frac{\sum_{n \in nOut(h)} |\mathbf{RT}(n)|}{\|nOut(h)\| + \lambda} \right)^{\gamma} \cdot \left(\frac{\sum_{s \in sOut(h)} |\mathbf{RT}(s)|}{\|sOut(h)\| + \lambda} \right)^{1-\gamma}$$

$\mathbf{H}_{ns}(h)$ increases as the average of the $|\mathbf{RT}|$ values of both normal-like and spam-like out-neighbors grows. When the average of the $|\mathbf{RT}|$ values of either normal out-neighbors or spam out-neighbors becomes lower, $\mathbf{H}_{ns}(h)$ decreases since a site h seems to be a spam or normal site. If we use a bigger γ value, we strengthen $|\mathbf{RT}|$ of normal-like out-neighbors than that of spam-like ones. If we use 0 for γ, $\mathbf{H}_{ns}(h)$ will be $\mathbf{H}_s(h)$.

5. Experiments

To evaluate our method, we perform experiments using the large-scale snapshot of our Japanese Web archive crawled in 2004. Core-based PageRank scores $\mathbf{PR}^+$ and $\mathbf{PR}^-$ are used for the white and spam scores, respectively. After the $\mathbf{RT}$ value of each site are obtained based on the white and spam scores, we compute two types of hijacked scores and compare the detection precision of them. In addition, we examine whether observing hijacked sites can help to discover newly emerging spam sites.

5.1 Data Set and Seed Set

To evaluate our algorithm, we perform experiments on the large-scale snapshot of our Japanese Web archive. We have been crawling the Web from 1999, and our archive contains over 10 billion pages. For the experiments, we use pages crawled in May 2004. 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 when they are written in Japanese. We use a site as a unit when filtering non-Japanese pages. The crawler stops collecting pages from a site, if it cannot find any Japanese pages on the site within the first few pages. Hence, our data set contains fairly large 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. To build a site graph, we choose the representative page of each site that has 3 or more incoming links from other sites, and whose URL is within 3 tiers (e.g. http://A/B/C/). Pages below each representative page are contracted to one site. 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 data set a web graph in our experiments.

To compute the white and spam scores, we construct white and spam seed set. Seed sites are selected by manual and automated methods.

To generate the white seed set, we refer the method in [8] and [10]. We compute PageRank scores of whole sites and perform 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. In addition to this, sites with specific URL including .gov (US governmental sites) and .go.jp (Japanese governmental sites). In the end, we have 40,396 trustworthy 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 use spam sites obtained by [13]. Saito et al. obtained this large spam seed set by following steps. First, they decomposed the Web into strongly connected components (SCC) based on the assumption that spam sites form SCC. Large SCCs except the largest one were regarded as spam. To detect spam sites in the largest SCC, or a core, Saito et al. considered maximal cliques. Cliques whose sizes were less than 40 were extracted from the core, and about 8,000 spam sites were obtained from them. Finally, they used these spam sites as a reliable spam seed set and expanded it by a minimum cut technique to separate links between spam and non-spam sites. Since this spam detection method showed a high precision, we use their spam sites as seed sites. Finally, a total of 580,325 sites is used as a spam seed set.

5.2 Types of Hijacking

In order to understand a layout of sites at the boundary of spam, we collect in-neighbors of spam seeds within three hops. From those sites, we randomly select 1,392 samples and manually classify them into 4 categories; hijacked, normal, spam and unknown. Unknown sites are written in unrecognizable languages such as Chinese, Dutch, German and so on. Table 1 shows the result of the classification. The 33% of total sites is 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 hijacked expired sites. Hijacked expired 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 voluntarily make links pointing to spam sites, because they believe those spam sites are normal and 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 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 Parameter Selection

To select the penalty parameter λ and the weight parameter γ(See Section 4), hijacked scores of 1,392 samples described in Section 5.2 are obtained. Types of

Table 1 The number of sample sites in each type

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

Table 2 Types of hijacked sites

Hijacked site type	Number of sites
Blog and BBS	117
Expired sites	78
Hosting sites	64
Link to expired site	60
Link register sites	55
Link to spam by mistake	51
Advertisement to spam	30
Server statistics	10
Total	465

†Sites of reputable companies such as `adobe.com`, `microsoft.com` are included in the white seed set. For other sites, we check them manually with yearly web snapshots from 2004 to the present. If a site remains without spam contents and controlled by the same authority, we select it as a white seed.

Table 3 The number of hijacked sites in top 300 sample sites with high $\mathbf{H}_{ns}$ score obtained with different δ and γ. λ is fixed to 60.

γ / δ	-5	-4	-3	-2	-1	0	1	2	3	4
0.0($\mathbf{H_s}$)	100	99	100	109	121	144	166	171	161	144
0.3	110	114	129	144	167	179	170	159	141	138
0.4	112	120	140	165	177	189	163	151	139	133
0.5	114	125	159	177	189	187	159	146	140	133
0.6	139	161	181	196	189	183	151	144	136	133
0.7	168	188	**205**	200	182	171	152	148	136	132
0.8	185	198	193	179	169	165	150	146	135	130
0.9	189	187	177	159	154	150	142	143	135	134

top 300 sites are examined and parameter values that showed the best precision are selected for the hijacked score computation of whole sites. For the white score and spam score, we used core-based PageRank scores.

In both $\mathbf{H}_s$ and $\mathbf{H}_{ns}$, the best precision is achieved when λ is 60. We find that if the value of λ exceeds 60, the number of spam sites in the top result hardly changes. The fraction of normal sites with a high hijacked score remains stable regardless of λ.

To select weight parameter γ of $\mathbf{H}_{ns}$, we examine the number of hijacked sites in top 300 sites with high $\mathbf{H}_{ns}$ calculated with different γ and δ values. As shown in Table 3, the precision is getting higher as the value of δ decreases and the value of γ increases. This means if we select a site s as a hijacked candidate even if $\mathbf{White}(s)$ is lower than $\mathbf{Spam}(s)$, we should intensify the influence of trustworthiness of normal-like out-neighbors in $\mathbf{H}_{ns}$. However, this tendency does not continue if δ is smaller than -3. The best result is achieved when δ is -3 and γ is 0.7.

5.4 Evaluation

With core-based PageRank scores and parameters determined in Section 5.3, $\mathbf{H}_s$ and $\mathbf{H}_{ns}$ of whole sites are calculated.

The result of $\mathbf{H}_s$ For δ from $+1$ to $+4$, we choose top 200 sites with high $\mathbf{H}_s$ scores and categorize them into hijacked, normal, spam, and unknown by hand. † The detail is shown in Table 4. The best precision 44.5% is obtained when δ is $+3$. The penalty parameter λ is fixed to 60.

The result of $\mathbf{H}_{ns}$ With different δ values from -4 to -1, we compute $\mathbf{H}_{ns}$ score and evaluate top 200 sites. As described in Table 5, we detect hijacked sites with the best precision of 70% when δ is -3. This result is better than that of $\mathbf{H}_s$ by 25.5%. The penalty parameter λ is 60 and the weight parameter γ is 0.7.

We can notice that δ increases, the number of normal sites increases in both Table 4 and 5. This is because with a higher δ, a site should have a higher white score to be a hijacked candidate. Likewise, as δ decreases, the proportion of spam sites increases. This means our algorithm adds sites with a relatively high spam score into the hijacked candidate set.

140 hijacked sites obtained by the best performance of $\mathbf{H}_{ns}$ are categorized into different hijacked types. Table 6 shows the detail. The most dominant hijacked type is blog and BBS which is followed by hosting. Note that we successfully find several expired sites which seems most useful to discover emerg-

Table 4 Top 200 precision of $\mathbf{H}_s$

δ	1	2	3	4
Hijacked	55	75	89	65
Normal	3	4	25	78
Spam	132	109	79	50
Unknown	10	15	7	7
Total	200	200	200	200
Precision	22.5%	37.5%	44.5%	32.5%

Table 5 Top 200 precision of $\mathbf{H}_{ns}$

δ	-4	-3	-2	-1	0
Hijacked	138	140	139	128	110
Normal	25	25	36	47	72
Spam	37	33	23	22	16
Unknown	0	2	2	3	2
Total	200	200	200	200	200
Precision	69%	70%	69.5%	64%	55%

Table 6 Breakdown of detected hijacked sites by $\mathbf{H}_{ns}$ when $\delta = -2, \lambda = 60$ and $\gamma = 0.7$.

Hijacked site type	Number of sites
Blog and BBS	48
Expired sites	19
Hosting sites	30
Link to expired site	13
Link register sites	8
Link to spam by mistake	18
Advertisement to spam	0
Server statistics	3
Total	140

†Labeling sites is expensive and time consuming. To determine whether a site s is a hijacked or not, first we check s is normal or spam. If it is normal, then we check its out-neighbors whether there are spam sites. If we find a spam out-neighbor, we examine if a link to such a out-neighbor is created by a spammer or by a site author. To judge a site to be an expired site, we have to check past snapshots. Only when the site was normal in the past, and is spam in the present and linked by normal sites, we determine a site as an expired site.

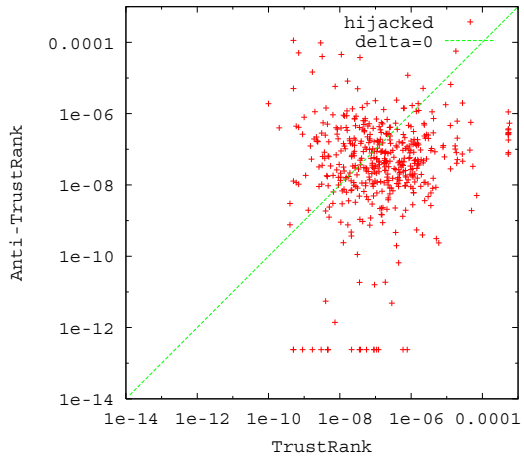


Fig. 1 TrustRank and Anti-TrustRank score pair

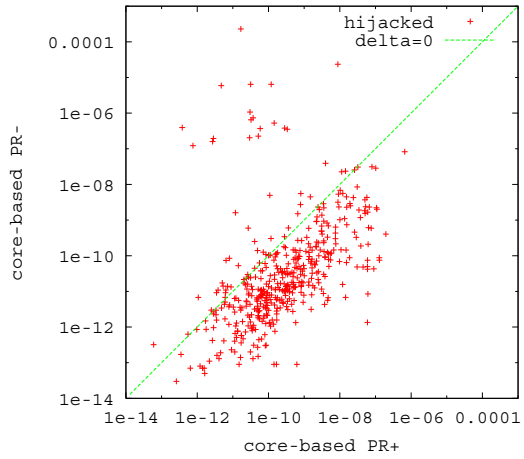


Fig. 2 Core-based PR+ and Core-based PR- pair

ing spam sites.(See Section 5.6)

5.5 Comparision of Different Score Pairs

We computed the hijacked scores using 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. To clarify the reason of this, we examine each score pair of hijacked sites described in Section 5.2. Figure 1 and 2 demonstrate the result. Log scale is used for x and y axis. It is shown that the core-based PageRank score pair of hijacked sites have some linear relationship compared to TrustRank and Anti-TrustRank pair. Since hijacked sites with a high $\mathbf{PR}^-$ score appear in Figure 2, we check them manually and find that all of such sites are hijacked expired sites that have turned into spam. Pearson correlation coefficient of the core-based PageRank pair is 0.73 if we exclude scores of expired sites. However, correlation coefficient of the TrustRank and Anti-TrustRank pair shows 0.1, which is quite low.

Table 7 The number of spam sites in 2005 and 2006 discovered by observing outgoing links of hijacked pages.

Year Out sites	2005 spam / total	2006 spam / total	Total spam / total (%)
BBS1	64/68	23/25	87/93(93.5%)
BBS2	12/13	0/0	12/13(92.3%)
Blog1	0/4	0/13	0/17(0%)
Blog2	73/73	0/0	73/73(100%)
Expired1	1964/1981	4/8	1968/1989(98.8%)
Expired2	1/1	21/21	22/22(100%)

Note that the fact that the best detection precision is obtained when we use a negative δ value(See Section 5.4) does not imply hijacked sites generally have a higher spam score than its white score. Table 3 and 5 show that most hijacked sites already have detected when $\delta = 0$, which suggests hijacked sites is likely to have a higher or same white score as its spam score.

5.6 Spam Sites Discovery by Tracking Hijacked Sites

To confirm that observing hijacked sites can help spam detection, we randomly select six sites from sample hijacked sites described in Section 5.2: two blogs, two BBS, and two expired sites. These three hijacked types are chosen because they are assumed to be hijacked easily and continuously by spammers.

From six sample sites, we pick up a page p in each site s which points to more than one site that has a negative **RT** value, and has a lower white score and a higher spam score than s. With selected pages, we extract their out-neighboring pages from the web snapshot of 2005 and 2006 that did not linked by hijacked pages in 2004.

For the evaluation, we manually check newly appeared out-neighboring pages whether they are spam or not. If a page is spam, a site containing that page is judged spam. If multiple pages are appeared in one site and one of them are spam, that site is classified as spam. †

As shown in Table 7, almost all newly appeared sites in out-neighbors are spam. We find that by observing an expired site, many spam sites can be detected if an expired site belongs to a spam farm that continuously grows. There is no newly created links to spam pages on Blog2. It seems that the author failed to delete hijacked links in old postings of 2004.

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 can affect link-based ranking algorithms. Thus, detecting hijacked sites and penalizing hijacked links is an important problem to be

†Note that pages that cannot be opened and pages written in unrecognizable languages are discarded.

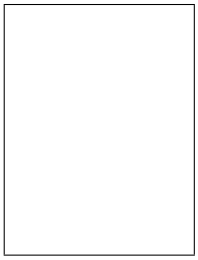
solved.

To find hijacked sites, we focused on the trustworthiness of a hijacked site and its out-neighboring sites. Based on that a hijacked site is the trustworthy site pointing to untrustworthy sites, we defined two different types of a hijacked score that evaluates how likely a site is hijacked by spammers.

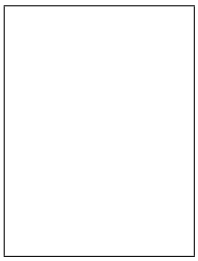
Experimental results showed that our approach is quite effective. The best precision in the hijacked site detection was 70%. We also compared two types of the hijacked scores. Hijacked scores that consider the distribution of the trustworthiness in both normal and spam out-neighbors outperformed 25.5% compared to scores that consider only spam out-neighbors. We also showed that by observing hijacked pages in detected sites, we can discover newly appearing spam sites with a high probability.

References

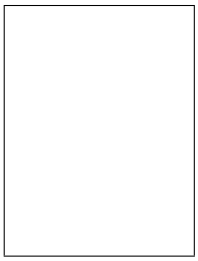
[1] The Official Google Blog, `http://googleblog.blogspot.com/2008/07/we-knew-web-was-big.html`
[2] S. Nakamura, S. Konishi, A. Jatowt, H. Ohshima, H. Kondo, T. Tezuka, S. Oyama, K. Tanaka, "Trustworthiness Analysis of Web Search Results", Proc. 11th European Conference on Research and Advanced Technology for Digital Libraries. Budapest, Hungary, 2007.
[3] A. Ntoulas, M. Najork, M. Manasse, and D. Fetterly, "Detecting Spam Web pages through Content Analysis", Proc. of 15th International Conference on World Wide Web. Edinburgh, Scotland, UK, 2006.
[4] D. Fetterly, M. Manasse and M. Najork, "Spam, Damn Spam, and Statistics: Using Statistical Analysis to Locate Spam Web Pages", Proc. 7th International Workshop on the Web and Databases. Paris, France, 2005.
[5] L. Page, S. Brin, R. Motwani, T. Winograd, The PageRank citation ranking: Bringing Order to the Web. Technical report,Stanford Digital Library Technologies Project, Stanford University, Stanford, CA, USA, 1998.
[6] Z. Gyöngyi and H. Molina, "Link Spam Alliance", In : 31st International Conference on Very large Data Bases, Trondheim, Norway, 2005.
[7] Y. Du, Y. Shi, X. Zhao, "Using Spam Farm to Boost PageRank", In : 3rd International Workshop on Adversarial Information Retrieval on the Web. Banff, Alberta, Canada, 2007.
[8] Z. Gyöngyi, H. Garcia-Molina and J. Pedersen, "Combating Web spam with TrustRank", In : 30th International Conference on Very Large Data Bases. Toronto, Canada, 2004.
[9] B. Wu, B, V. Goel, B. D. Davison, "Topical TrustRank: Using Topicality to Combat Web Spam", In : 15th International Conference on World Wide Web. Edinburgh, Scotland, UK, 2006.
[10] Z. Gyöngyi, P. Berkhin, H. Garcia-Molina and J. Pedersen, "Link Spam Detection Based on Mass Estimation", In : 32nd international conference on Very Large Data Base. Seoul, Korea, 2006.
[11] V. Krishnan, R. Raj, "Web Spam Detection with Anti-TrustRank", In : 2nd International Workshop on Adversarial Information Retrieval on the Web. Edinburgh, Scotland, UK, 2006.
[12] A. Benczur, A. K. Csalogány, T. Sarlós, M. Uher, "SpamRank-fully automatic link spam detection", In : 1st International Workshop on Adversarial Information Retrieval on the Web. Chiba, Japan, 2005.
[13] H. Saito, M. Toyoda, M. Kitsuregawa and K. Aihara, "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] M. Najork and J. L. Wiener. "Breadth-first Crawling Yields High-quality Pages", In : 10th international conference on World Wide Web, Hong Kong, China, 2001.
[15] The Official Google Blog, `http://googleblog.blogspot.com/2005/01/preventing-comment-spam.html`



Young-joo Chung received her B.S in Computer Science and Engineering from Seoul National University, Korea in 2005. In 2008, she received the M.S in Information Engineering from the Department of Information and Communication Engineering of the University of Tokyo where she is currently a Ph.D Candidate. Her research interests include Web mining and analysis.



Masashi Toyoda is an Associate Professor of the Institute of Industrial Science, the University of Tokyo, Japan. He received B.S, M.S and Ph.D degrees in Computer Science from Tokyo Institute of Technology, Japan, in 1994, 1996, 1999, respectively. He worked at the Institute of Industrial Science, the University of Tokyo, as a Specially Appointed Associate Professor from 2004 to 2006. His research interests include web mining, user interfaces, information visualization and visual programming. He is a member of the ACM, IEEE CS, IPSJ, and JSSST.



Masaru Kitsuregawa is currently a Full Professor and a Director of the Center for Information Fusion at the Institute of Industrial Science, the University of Tokyo. He received B.S, M.S in Electronics Engineering from the University of Tokyo, Japan, 1978 and 1980, respectively. From the same university, he received Ph.D degree in Information Engineering in 1983. His current research interests cover database engineering, Web archive/mining, advanced storage system architecture, parallel database processing/data mining, digital earth, and transaction processing. He served as Program Co-chair of the IEEE International Conference on Data Engineering (ICDE) 1999, and served as General Co-chair of ICDE 2005 (Tokyo). He served as a VLDB trustee and an ACM SIGMOD Japan Chapter Chair. Prof. Kitsuregawa is a Fellow of the IPSJ and IEICE, Japan, and he currently serves a director of the DBSJ. He is a member of the IEEE CS.