Master thesis

A Study of Methods for Extracting the Boundaries of Web Spam

ウェブスパムの境界抽出手法に関する研究

February 4th, 2008

Supervisor
Professor Masaru Kitsuregawa

Department of Information and Communication Engineering
Graduate School of Information Science and Technology
The University of Tokyo

48-66450 Young joo Chung

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 higher ranking. Since link-based ranking algorithms are important tools for current search engines, web spammers are making a significant effort to manipulate the link structure of the Web, namely, link spamming.

Link hijacking is one technique of link spamming. By hijacking links from normal sites to target spam sites, spammers can make search engines believe that normal sites endorse spam sites.

In this research, we propose link analysis techniques for finding out link-hijacked sites using modified PageRank algorithms. We tested our methods on a large scale Japanese web archive and evaluated the accuracy. Our contributions are as follows:

- We proposed link hijacking detection methods.
- We evaluated our methods with a large Japanese Web data set.
- We examined and classified link hijacking techniques.

Keywords:
Link analysis, Web spam, Link hijacking, Information retrieval

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Today, the Internet and the World Wide Web are the essential means to seek, find, distribute, and process information. Now, ever growing human knowledge of politics, economics, medicine, science, commerce and more is being gathered on the Web. According to Yahoo! Search, Over 20 billion items have been indexed in 2005.

As information on the Web has grown exponentially, the means of finding timely and accurate information from the Web are getting important. Web scale search engines are tools that help users find information by providing the ordered lists of web pages.

Usually, web search engines provide the ordered list of pages when users ask information with a specific query. A survey carried out in 2007 showed that approximately 50% of search engine users look at no more than the first 5 results in the search list [1]. It implies that sites should be ranked in top to attract visitors and yield financial profit. Hence, obtaining high rankings in search results becomes crucial for the success of sites.

1.2 Web spamming

Web spamming is defined as the behavior of manipulating the web page features with an intention of getting a higher ranking than it deserves without manipulation. Spam pages have pervaded whole web and now, at least 13% of English-language page was estimated as spam [20]. In addition, Amit Singhal, the principal scientist of Google Inc. indicated that search engine spam industry could yield $4.5 billion in 2004 if they have deceived all search engines on all commercially viable queries [21]. The number of spam pages has increased to around 15 to 18 percent from 2003 to 2004 [6]. This increase tendency makes researchers believe that web spam is serious problem.

Web spamming techniques can be categorized into term spamming and link spamming [2]. *Term spamming* is the behavior to manipulate textual contents of pages. Spammers can repeat specific keywords and add irrelevant meta-keywords or anchor texts that are not related with page contents. 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 link structure of the Web to mislead link-based ranking algorithms such as PageRank [3]. Since link-based ranking algorithms take account the number of incoming links to decide the importance of pages, spammers create many links and deceive search engines. For example, spammers can construct a spam farm, an artificially interlinked link structure, to centralize link-based importance scores [4]. After constructing spam farms, spammers should create links from external reputable pages to target spam pages in order to attract an attention of the search engines and get importance score. This behavior is called *link hijacking*.

1.3 Link hijacking

As we can see in Figure 1, posting comments including URLs to spam pages on public bulletin boards is a well-known hijacking method. Hijacked links do not endorse any relevance or quality of pages, so they mislead link-based ranking algorithms which consider the link as human judgment about web pages. Hijacked links could make significant impact on ranking algorithms, since they are usually connected to a large amount 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 research has 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 importance of a large amount 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 countermeasures are not devised. 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 sites by stopping crawling at hijacked link.

In order to find out hijacked sites, we consider the characteristics of the link structure around hijacked sites which is illustrated in Figure 2. While a hijacked site has links pointing to spam

Figure 1 The example of link hijacking: blog and bbs

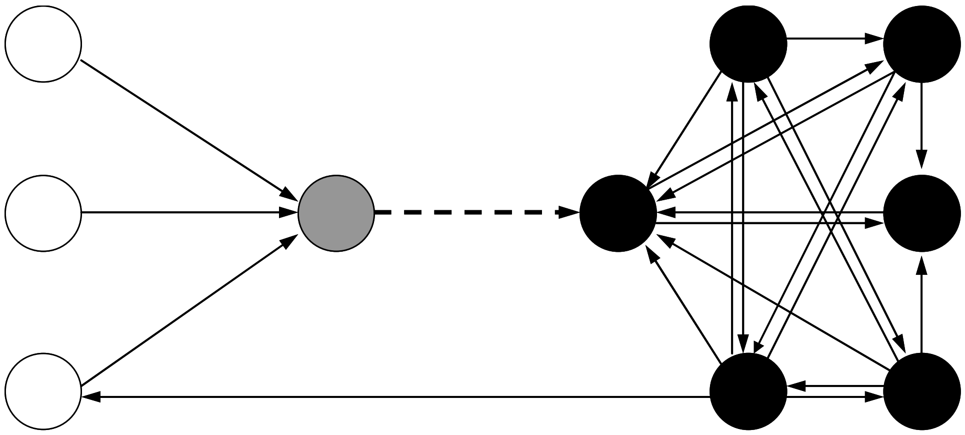


Figure 2 Link structure around a hijacked site. White, gray and black nodes represent normal, spam and hijacked sites, respectively. A dashed link from the hijacked site to a spam site is a hijacked link.

sites, it is rarely pointed to by the spam sites because they have few incentives to share link-based ranking score with hijacked sites. Consequently, we can see a significant change in the link structure between the 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 modified versions of PageRank. For each site, we calculate trustworthiness and spamicity using two different modified PageRank. Intuitively, the spamicity of a site in spam farms might overwhelm its trustworthiness, and the trustworthiness of a hijacked site might overwhelm its spamicity. With this observation, we consider the inverse search of the Web graph from sample spam sites. During the walk, we would find out hijacked sites where the order of the spam value and trust value is reversed. We will mark this site as hijacked.

We also thought the difference of trustworthiness and spamicity between one node and its out neighbors. If a node is hijacked, it will have higher trustworthiness and lower spamicity while a hijacking spam node in its out neighbors has lower trustworthiness and higher spamicity. Based on this change in trustworthiness and spamicity, we introduce a *hijacked score* which is defined as the difference of trustworthiness between a hijacked site and hijacking sites. We would find out hijacked site by computing total hijacked score of normal sites and ordering normal sites by decreasing order of the hijacked score.

We tested our methods and evaluated the precision of them on large-scale graph of the Japanese Web archive including 5.8 million sites and 283 million links.

1.4 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 white scores of the entire Web, TrustRank assigns initial white 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 distrust 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 incoming links 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 gets 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 the fair ranking decision. Carvalho et al. proposed the idea of noisy links, a 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 considered 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 new approaches to discovering hijacked link and pages. With our approaches, we hope to contribute to a new spam detection technique and improve the performance of link-based ranking algorithms.

The rest of this paper proceeds as follows. In Section 2, we review background knowledge for PageRank and link spamming and also introduce several approaches to detecting or demoting link spamming. Section 3 presents our methods to detect hijacked sites. In Section 4, we report experimental result of our algorithms. Finally, we discuss the result of our approaches.

Chapter 2

Preliminaries

2.1 Web graph

The entire Web can be considered as a directed graph. We can denote the Web as $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E})$, where $\boldsymbol{V}$ is the set of nodes and $\boldsymbol{E}$ 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 $\| \boldsymbol{V} \|$, 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:

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

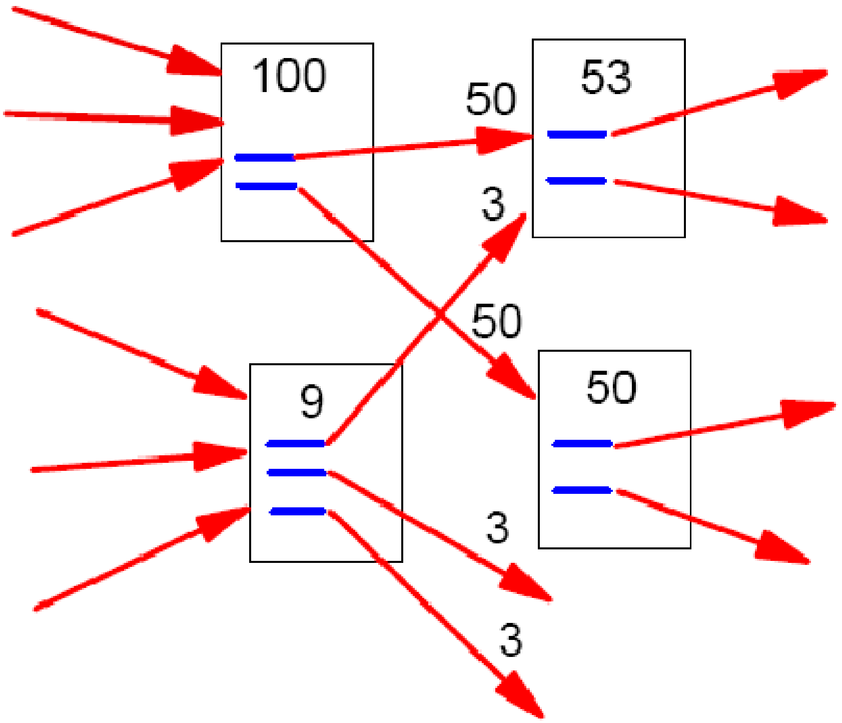


Figure 3 Simplified PageRank calculation

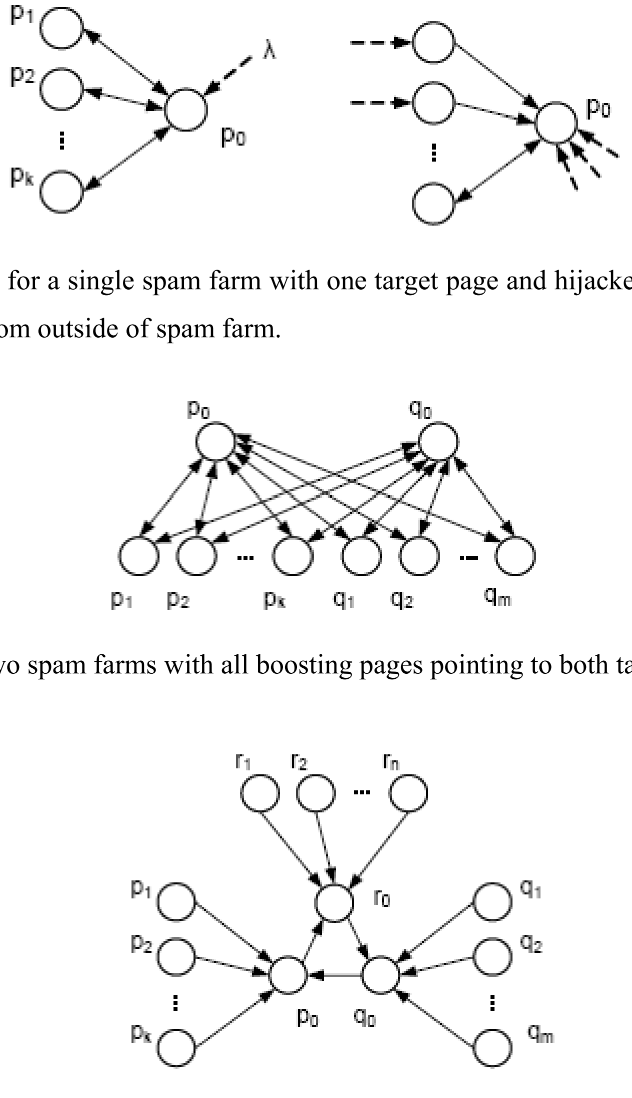
where **p** is PageRank score vector, **T** is transition matrix. $T(p, q)$ is $1/Out(q)$ if there is a link from node q to node p, and 0 otherwise. The decay factor $\alpha < 1$ (usually 0.85) is necessary to guarantee convergence and to limit the effect of rank sink. **d** is a uniformly random distribution vector.

The PageRank algorithm performs a random walk on the web graph G that simulates the behavior of a random surfer. The surfer starts from some page chosen according to some distribution. At each step, the surfer proceeds as follow: with probability α, an outgoing link is selected and the surfer moves to a new page, and with probability 1-α, the surfer jumps to a random page chosen according to distribution **d**. The PageRank score of page p is the fraction of time that the surfer spends at page p, that is proportional to the number of visits to page p during the random walk. Simplified PageRank computation is shown in Figure 3.

2.3 Link spamming

After the success of Google that adopted PageRank as the main ranking algorithm, PageRank became the main target of link spammers. Z. Gyöngyi et al. studied 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, all members of a spam farm can boost their PageRank scores. Various types of spam farms are displayed in Figure 4. 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 structure, spammers should create external links from outside of spam farms in order to provide PageRank score to the target page. To make links from non-spam sites to their own spam site, spammers send trackbacks that lead to spam sites or, post comments including links pointing to target spam sites. A large number of spam trackbacks and comments are created easily in a short period, so it could result in considerable score leakage. 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]. Hijacked pages are hard to detect because their contents and domains are irregular [5].



(a) Optimal structures for a single spam farm with one target page and hijacked links to make boost pages reachable from outside of spam farm.

(b) Two spam farms with all boosting pages pointing to both targets.

(c) Three spam farms with target pages forming a ring.

Figure 4 Various types of a spam farm

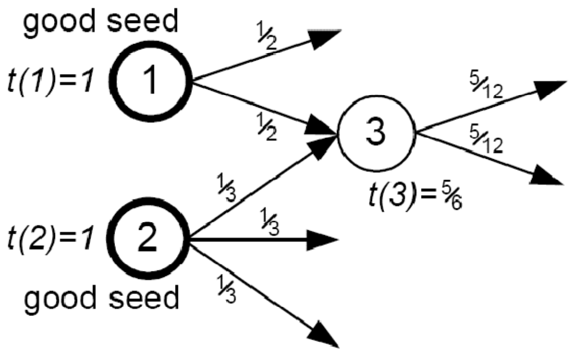


Figure 5 The example of trust propagation

2.4 TrustRank and Anti-TrustRank

To improve the PageRank algorithm, Gyöngyi et al. presented the TrustRank algorithm [6]. The basic intuition of TrustRank is that good sites seldom link spam sites. People trust good sites, and can trust sites pointed to by good sites. Like this, trust can be propagated through the link structure on the Web. So, in TrustRank, a list of highly trustworthy sites is selected as the seed set and each of these sites is assigned a non-zero initial trust score, while all the other sites on the Web have initial values of 0. A biased PageRank algorithm is used to propagate these initial trust scores to their outgoing sites. After convergence, good site will get a decent TrustRank score, while spam sites are likely to get lower scores. The matrix notation of TrustRank is following:

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

Where $\mathbf{t}$ is TrustRank score vector, α is decay factor (0.85), and $\boldsymbol{d}^{\boldsymbol{T}}$ is trust score vector of seed set where

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

V. Krishnan et al. modified TrustRank [7]. Instead of selecting good sites as a seed set, Anti TrustRank starts from spam sites. Each spam site is assigned Anti-Trust score and this score is propagated in the reverse direction along incoming links.

Chapter 3

Detecting Link Hijacking

3.1 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 sets [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 **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 vector $\mathbf{d}^{\boldsymbol{S}}$ is:

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

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

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

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.

3.2 Link hijacking detection algorithm with a backward traversal

Based on the characteristics of links structure around hijacked pages, we observe the changes in $\mathbf{PR}^{+}(p)$ and $\mathbf{PR}^{-}(p)$ during an inverse graph traversal starting from spam seed sites. As long as we are in a spam farm, the visited 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. By detecting this change in scores, we would find the hijacked sites. The algorithm is shown in Figure 6.

First, we compute $\mathbf{PR}^{+}(p)$ and $\mathbf{PR}^{-}(p)$ for each site p. Then start a inversed depth-first search from spam seed sites s- whose core-based PageRank scores are $\mathbf{PR}^{+}(s^{-}) < \mathbf{PR}^{-}(s^{-})$. The search from a site p is performed by selecting a site t whose $\mathbf{PR}^{+}(t)$ is greater than $\mathbf{P}\boldsymbol{R}^{+}(p)$. When it reached at a site q where $\mathbf{PR}^{+}(q) > \mathbf{PR}^{-}(q)$, we output this site as a hijacked page, and stop the further search from this site. We can change the stopping condition as follows:

$$\delta \ < \ \log PR^{+}(s) \text{ - } \log PR^{-}(s)$$

δ means the proposition of PR+ and PR- of a hijacked site. When we use a higher δ value, we consider PR+ of a hijacked site is much higher than PR-. Therefore, we need a further search toward a normal side. When we use a lower δ value, we regard PR+ of a hijacked site is lower, so we can stop the search earlier. Like this, we can adjust when we stop the search, by modifying δ from $-\infty$ to ∞.

3.3 Link hijacking detection algorithm with hijacked score

There exists the limitation when we use a backward traversal method to extract hijacking sites. Since prior method performs an inverse walk from a spam seed, it is difficult to discover a link hijacking by spam sites that do not connected with a spam seed set. In order to overcome this limitation, we propose a different approach that discovers hijacked sites with the difference of trustworthiness between normal sites and hijacking spam sites.

When a normal site p is hijacked by a spam site q, $\mathbf{PR}^+(p)$ would be higher than $\mathbf{PR}^+(q)$ and $\mathbf{PR}^-(p)$ would be lower than $\mathbf{PR}^-(q)$. We investigate all outlinks of a normal site p where $\mathbf{PR}^+(p) > \mathbf{PR}^-(p)$, and compute hijacked score that $\mathbf{PR}^+(p) - \mathbf{PR}^+(q)$ for each q where $\mathbf{PR}^-(q) > \mathbf{PR}^+(q)$, namely, potential spam sites. Figure 7 shows this algorithm. We adopted δ for hijacked score computation, too.

```
Link hijacking detection with Backward traversal

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

$H \leftarrow \phi$
Compute core-based PageRank score $PR+$ and $PR-$

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  $\log PR^+(s) - \log PR^-(s) > \delta$  then
        $H \leftarrow H \cup \{s\}$
        return
end if
for each site $t$ where $\{t|t \in \mathrm{In}(s) \wedge PR^+(s) < PR^+(t)\}$
        dfs($t$, $H$)
end for
end procedure
```

Figure 6 Link hijacking detection with a backward traversal

```
Link hijacking detection with hijacked score

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

$\boldsymbol{H} \leftarrow \emptyset$
compute white and spam score

for each site $p$ where $\delta < \log PR^{+}(p)$ - $\log PR^{+}(p)$
    $\boldsymbol{R} \leftarrow \emptyset$
    $\boldsymbol{R} = \{q|q \in \mathrm{Out}(p)$ and $\log PR^{+}(\mathrm{q}) - \log PR^{-}(\mathrm{q}) < \delta$ and $PR^{+}(q) < PR^{+}(p)$ and
          $PR^{-}(q) > PR^{-}(p)\}$
    if $\boldsymbol{R}$ is not empty then
      $\boldsymbol{H} \leftarrow \boldsymbol{H} \cup \{p\}$
      Hijacked score($p$)=$\sum_{\mathrm{q} \in \boldsymbol{R}} \log(\mathbf{White}(p)) - \log(\mathbf{White}(q))$
    end if
end for
return $\boldsymbol{H}$
```

Figure 7 Link hijacking detection with hijacked score

Chapter 4

Experiment

Our experiments consist of next variants.

- Algorithms
  - Backward traversal from seed set
  - Hijacked score
- Scores for trustworthiness and spamicity
  - TrustRank and Anti-TrustRank
  - core-based PR+ and core-based PR-
- Score reversal conditions
  - log(**PR+**(*p*)) - log(**PR-**(*p*)) > δ. Five different δ were tested.

4.1 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 used a site level graph of the Web, in which nodes represent web sites and edges for 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 incoming links 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 this paper. Certain properties and its statistics of domains of our web graph are shown in Table 1 and 2.

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 1 Properties of the web graph

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

Table 2 Domains in the web data

4.2 Seed Set

To compute a core-based PageRank, we constructed trust seed set and spam seed set. 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 as 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]. Saito et al. obtained this large spam seed set by following steps. First, they extracted strongly connected components (SCC) from web graph. SCC is maximal strongly connected subgraph where each pair of nodes has a directed path between them. Since spam sites tend to construct densely connected link structure, it could be assumed that spam sites form SCC. About 0.6 million spam sites in SCCs around the core (the largest strongly connected component) were obtained. To detect spam in the core, Saito et al. enumerated maximal cliques in the core. A clique is a subset of total nodes where every pair of nodes has an arc between them. They extracted clique whose size is less than 40 from the core and gained 8,000 spam sites. This spam extracting methods showed high precision, so we can use their spam sites as seed sites. Finally, Total 580,325 sites are used for a spam seed set.

4.3 Preparatory work

4.3.1 PageRank and TrustRank

To verify whether link hijacking affects on TrustRank, we computed PageRank and TrustRank. We chose the same way as used in TrustRank. First, we computed the PageRank score for each site. Then, we generated the list of site in decreasing order of their PageRank score, and distribute them into 20 buckets. Each bucket includes a different number of sites but total scores of sites in each bucket are equivalent. Sample set for evaluation was constructed by selecting 50 sites at random from each bucket. Then, we classified these sites manually. As a result, we obtained 233 trust sites and 206 spam site. We also computed TrustRank and made 20 buckets to include the same number of sites as PageRank bucket.

Figure 8 shows the number of white sites in each approach’s bucket, and Figure 9 shows the number of spam sites in each approach’s bucket. The horizontal axis corresponds to the bucket numbers, and the vertical axis means the number of sites in each bucket. Sites in top buckets have a higher probability to be shown in top rankings of the search result.

These figures show that TrustRank succeeded to demote spam sites up to a certain degree, but both failed to find good sites. Some white sites had lower rankings. Also, it has been found that there exist spam sites in upper buckets after manual check. These means spam sites can get high white scores, or there does exist trust leak.

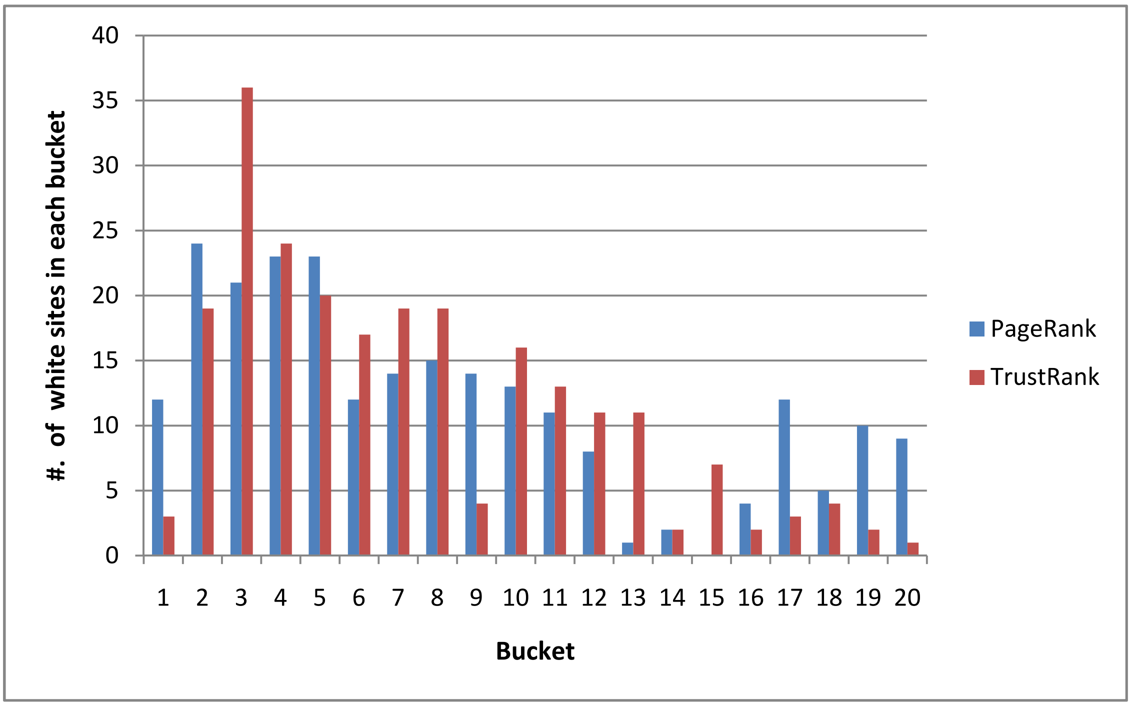


Figure 8 Number of white sites in PageRank and TrustRank

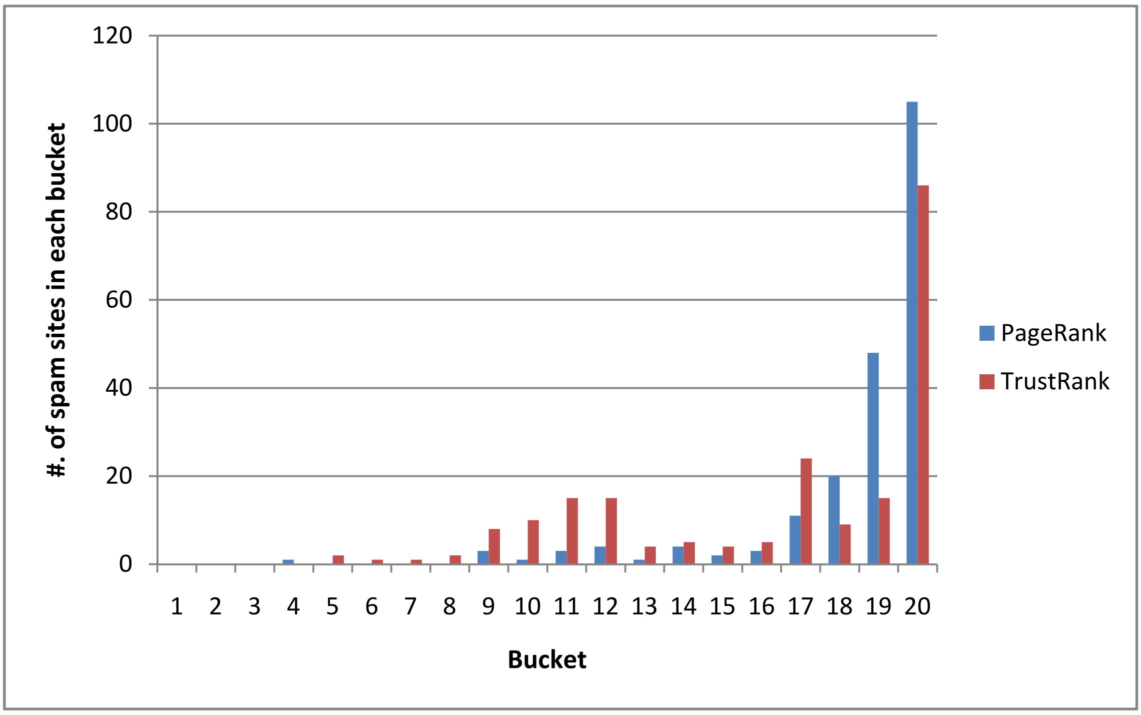


Figure 9 Number of spam sites in PageRank and TrustRank

4.3.2 TrustRank and Anti-TrustRank

As a preparatory work to extract hijacked sites, we computed TrustRank and Anti-TrustRank for whole web sites. We assumed:

- Normal sites have a high TrustRank score and a low Anti-TrustRank score.
- Spam sites have a low Anti-TrustRank score and a high Anti-TrustRank score.
- Hijacked sites will have abnormal combination of TrustRank and Anti-TrustRank scores.

We computed TrustRank and Anti-TrustRank scores and looked into score distribution. The result showed that there exist some sites with abnormal TrustRank score and Anti-TrustRank score. Figure 12 shows this.

We counted sites above score 1e-10 (non-zero) to examine how many sites with abnormal scores exist. The detail is shown in Figure 13. Sites with non-zero TrustRank score and nearly zero Anti-TrustRank scores were 684,107 sites (27%). Sites with non-zero Anti-TrustRank and nearly zero TrustRank scores were 682,462 sites (27%). Finally, the number of sites with TrustRank scores and Anti-TrustRank scores above 1e-10 was 1,005,293(40%).

We picked up top sites from the target space and discovered that most of them are portal sites which contain many subpages that are very likely to be hijacked.

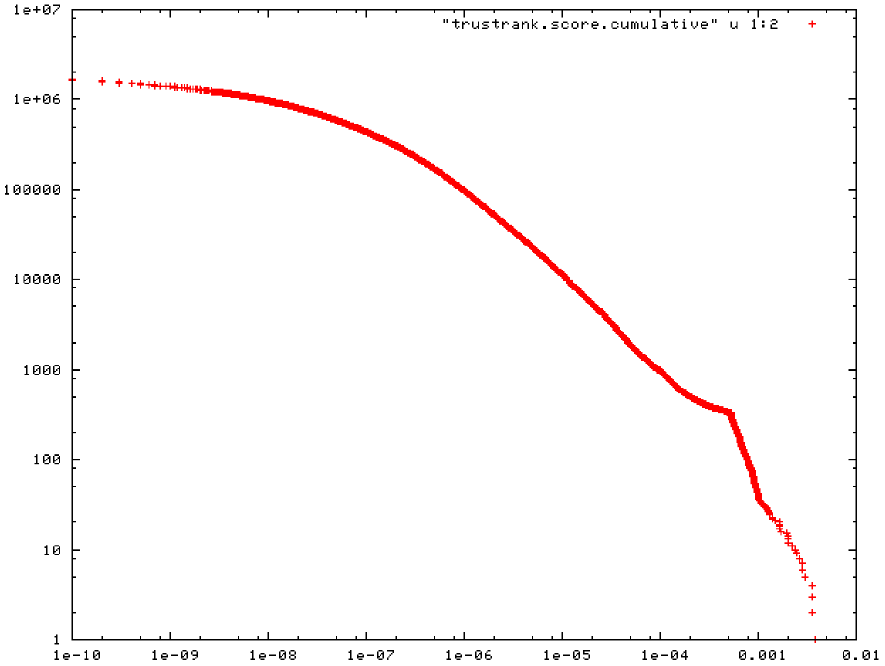


Figure 10 The cumulative distribution of TrustRank score

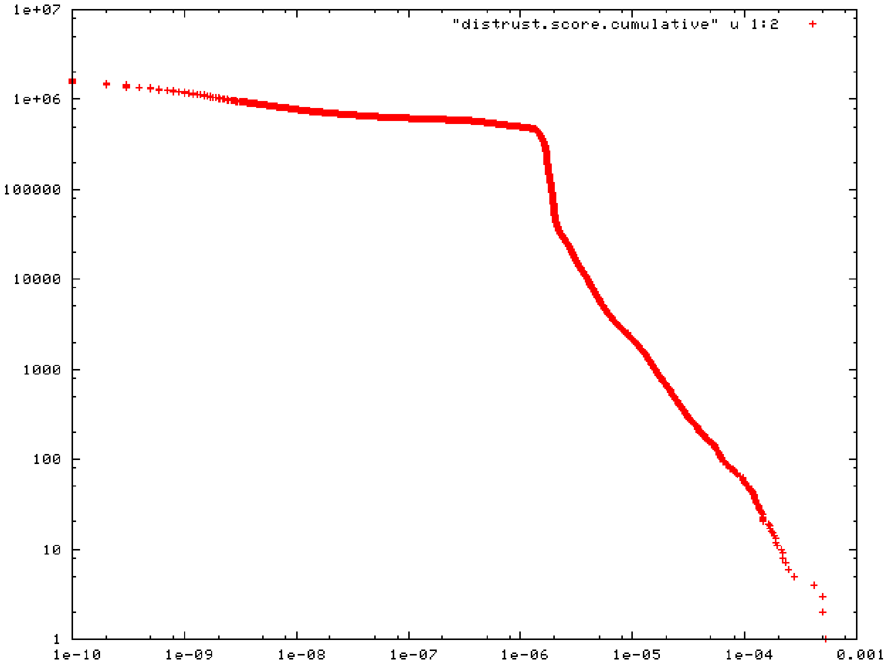


Figure 11 The cumulative distribution of Anti-Trust score

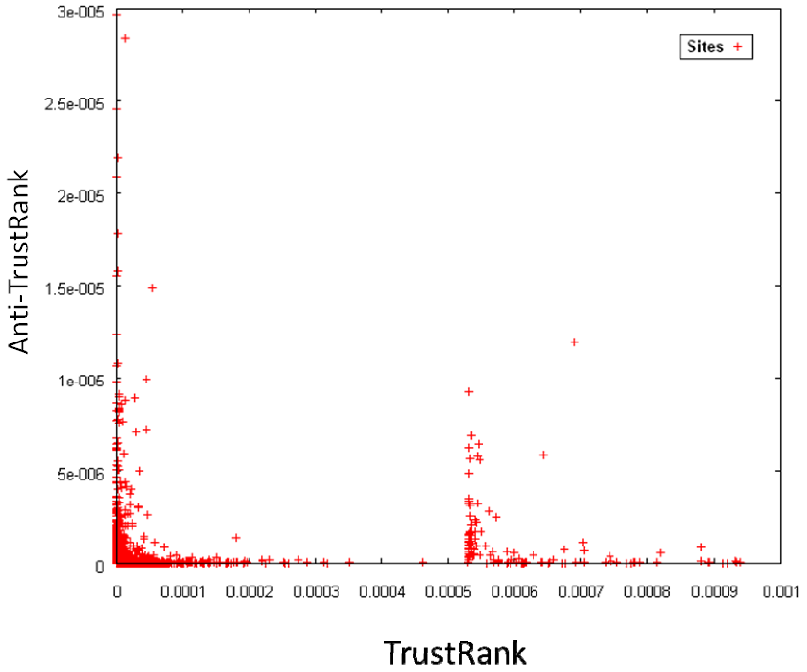


Figure 12 Score distribution of TrustRank and Anti-TrustRank

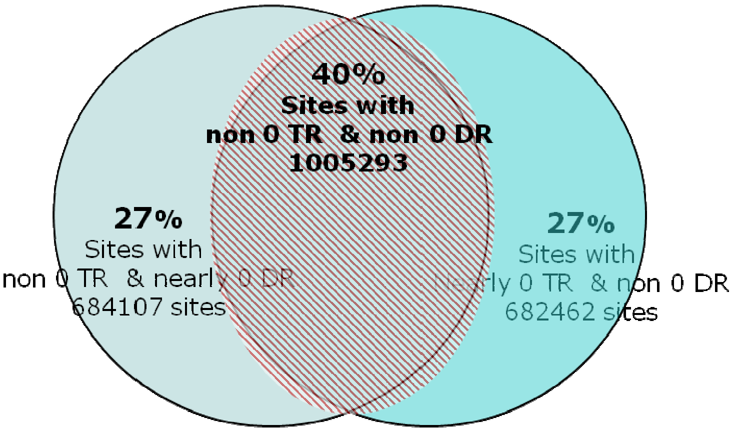


Figure 13 Target space detected by TrustRank and Anti-TrustRank

4.4 Types of Hijacking

We checked about 1000 sites during our experiments and obtained 284 hijacked sites. We divided them into7 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 their spam sites. Since web sites tend to maintain links pointing to expired domains for a while, spammers are able to get links from normal sites.
- Free link registration sites which allow spammers to register links on their sites.
- Hosting sites that include spam sites of some customers.
- Sites with public access log statistics showing links to referrers. Spammers access such sites frequently, and then their sites are appeared in the referrer list. Normal sites that contain advertisements to spam sites. Spammers can create links on normal sites by sponsoring them.
- Normal sites that point to expired hijacking sites.

Table 3 and Figure 14 show the number and percentage of sites of each hijacking type.

We can see that the most frequently used technique is Blog and bbs hijacking. This seems that since blog and bbs hijacking is relatively easy to do, many spam sites use this technique. Also, expired hijacking is a quite popular technique among spammers. Especially, official sites for movies are likely to be hijacked because their domains are used for a while, not permanently.

Figure 15, 16, 17 show the real examples of hijacking techniques.

Figure 15 shows the server statistics hijacking. This page displays the list of referrer blogs. We can see many links to spam blogs containing the word ‘*nude*’.

Figure 16 shows the advertisements to spam techniques. We can see the links to spam sites on the top of page. Although this page is about *java-linux news*, it contains many links to sports sites that construct link spam farm.

Figure 17 illustrates the hijacking technique by expired domains. The left site, *Satellite Image*, links to the sites of *spacetech,* which is a software company. However, the domain *www.spacetech.co.uk* had been expired and spammer bought it and used it for a spam site. Then, spammer can get link from normal sites, which was originally created to refer a software company.

Hijacking type	**the number of sites**
Blog and BBS	84
Expired sites	60
Link register sites	23
Hosting sites	40
Server statistics	7
Normal sites having ads to spam sites	29
Normal sites pointing to expired sites	41
Total	**284**

Table 3 Types of hijacking

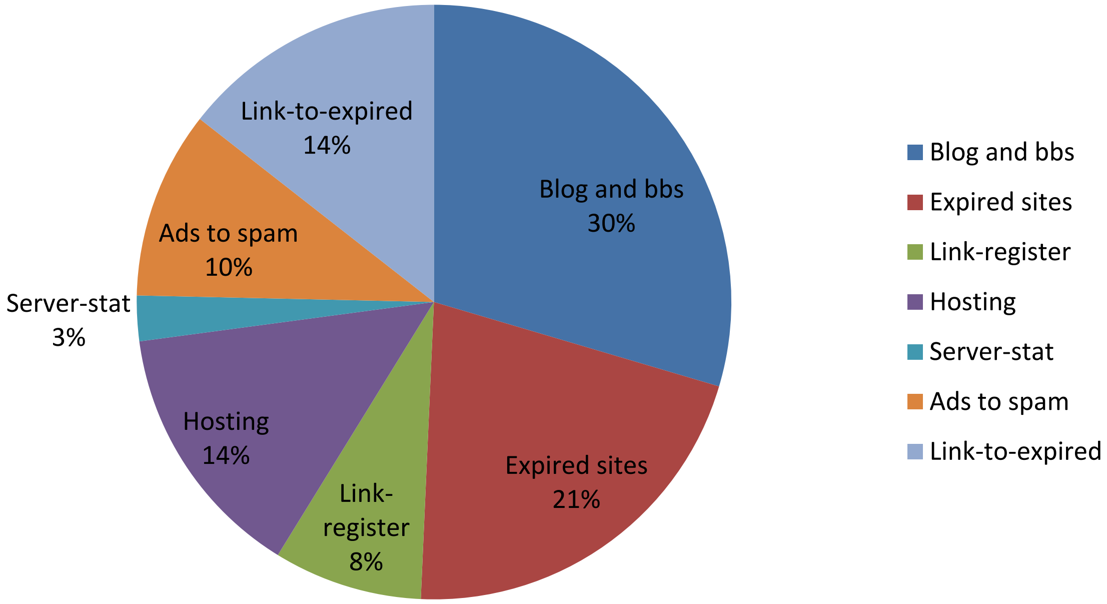
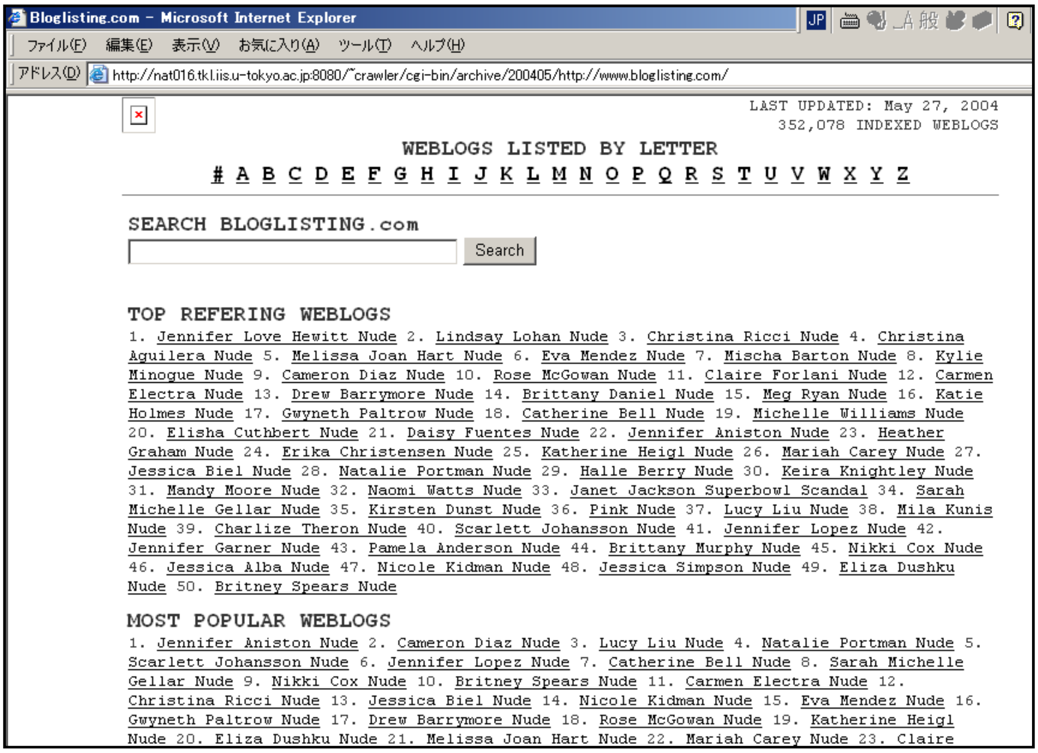


Figure 14 Percentages of each type of hijacking


Bloglisting.com - Microsoft Internet Explorer
ファイル(F) 編集(E) 表示(V) お気に入り(A) ツール(T) ヘルプ(H)
アドレス(D) http://nat016.tkl.iis.u-tokyo.ac.jp:8080/~crawler/cgi-bin/archive/200405/http://www.bloglisting.com/

LAST UPDATED: May 27, 2004
352,078 INDEXED WEBLOGS

WEBLOGS LISTED BY LETTER

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

SEARCH BLOGLISTING.com

Search

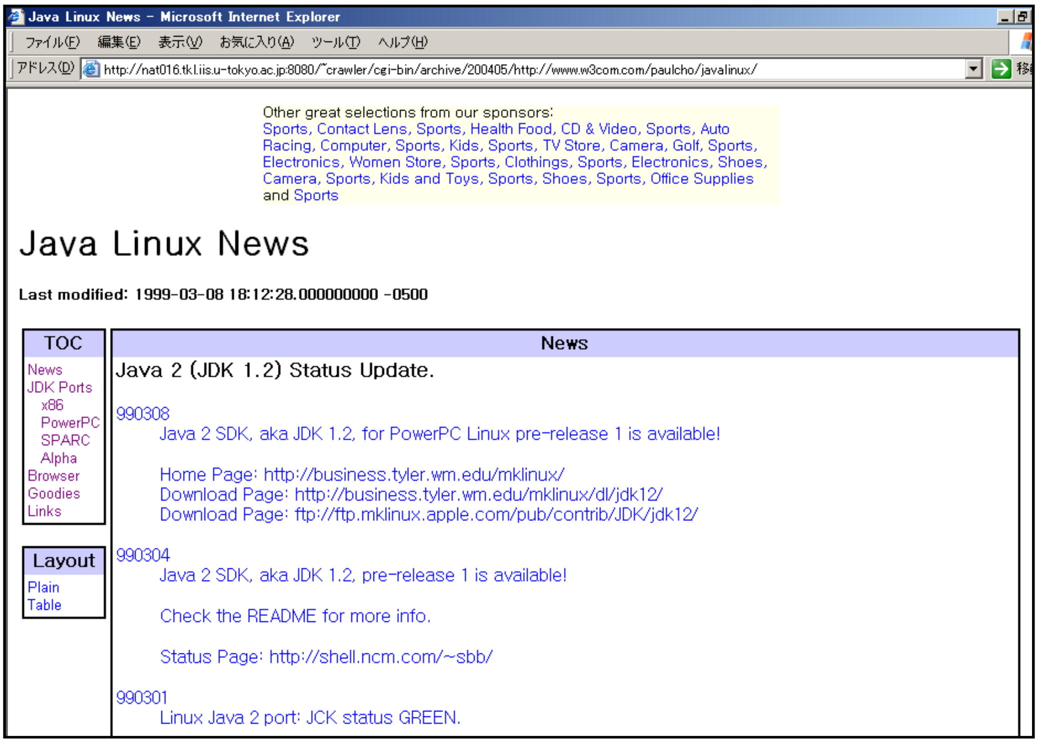
TOP REFERING WEBLOGS

1. Jennifer Love Hewitt Nude 2. Lindsay Lohan Nude 3. Christina Ricci Nude 4. Christina Aguilera Nude 5. Melissa Joan Hart Nude 6. Eva Mendez Nude 7. Mischa Barton Nude 8. Kylie Minogue Nude 9. Cameron Diaz Nude 10. Rose McGowan Nude 11. Claire Forlani Nude 12. Carmen Electra Nude 13. Drew Barrymore Nude 14. Brittany Daniel Nude 15. Meg Ryan Nude 16. Katie Holmes Nude 17. Gwyneth Paltrow Nude 18. Catherine Bell Nude 19. Michelle Williams Nude 20. Elisha Cuthbert Nude 21. Daisy Fuentes Nude 22. Jennifer Aniston Nude 23. Heather Graham Nude 24. Erika Christensen Nude 25. Katherine Heigl Nude 26. Mariah Carey Nude 27. Jessica Biel Nude 28. Natalie Portman Nude 29. Halle Berry Nude 30. Keira Knightley Nude 31. Mandy Moore Nude 32. Naomi Watts Nude 33. Janet Jackson Superbowl Scandal 34. Sarah Michelle Gellar Nude 35. Kirsten Dunst Nude 36. Pink Nude 37. Lucy Liu Nude 38. Mila Kunis Nude 39. Charlize Theron Nude 40. Scarlett Johansson Nude 41. Jennifer Lopez Nude 42. Jennifer Garner Nude 43. Pamela Anderson Nude 44. Brittany Murphy Nude 45. Nikki Cox Nude 46. Jessica Alba Nude 47. Nicole Kidman Nude 48. Jessica Simpson Nude 49. Eliza Dushku Nude 50. Britney Spears Nude

MOST POPULAR WEBLOGS

1. Jennifer Aniston Nude 2. Cameron Diaz Nude 3. Lucy Liu Nude 4. Natalie Portman Nude 5. Scarlett Johansson Nude 6. Jennifer Lopez Nude 7. Catherine Bell Nude 8. Sarah Michelle Gellar Nude 9. Nikki Cox Nude 10. Britney Spears Nude 11. Carmen Electra Nude 12. Christina Ricci Nude 13. Jessica Biel Nude 14. Nicole Kidman Nude 15. Eva Mendez Nude 16. Gwyneth Paltrow Nude 17. Drew Barrymore Nude 18. Rose McGowan Nude 19. Katherine Heigl Nude 20. Eliza Dushku Nude 21. Melissa Joan Hart Nude 22. Mariah Carey Nude 23. Claire

Figure 15 The example of server-statistics


Java Linux News - Microsoft Internet Explorer
ファイル(F) 編集(E) 表示(V) お気に入り(A) ツール(T) ヘルプ(H)
アドレス(D) http://nat016.tkl.iis.u-tokyo.ac.jp:8080/~crawler/cgi-bin/archive/200405/http://www.w3com.com/paulcho/javalinux/
移

Other great selections from our sponsors: Sports, Contact Lens, Sports, Health Food, CD & Video, Sports, Auto Racing, Computer, Sports, Kids, Sports, TV Store, Camera, Golf, Sports, Electronics, Women Store, Sports, Clothings, Sports, Electronics, Shoes, Camera, Sports, Kids and Toys, Sports, Shoes, Sports, Office Supplies and Sports

Java Linux News

Last modified: 1999-03-08 18:12:28.000000000 -0500

TOC

News
JDK Ports
x86
PowerPC
SPARC
Alpha
Browser
Goodies
Links

Layout

Plain
Table

News

Java 2 (JDK 1.2) Status Update.

990308
Java 2 SDK, aka JDK 1.2, for PowerPC Linux pre-release 1 is available!

Home Page: http://business.tyler.wm.edu/mklinux/
Download Page: http://business.tyler.wm.edu/mklinux/dl/jdk12/
Download Page: ftp://ftp.mklinux.apple.com/pub/contrib/JDK/jdk12/

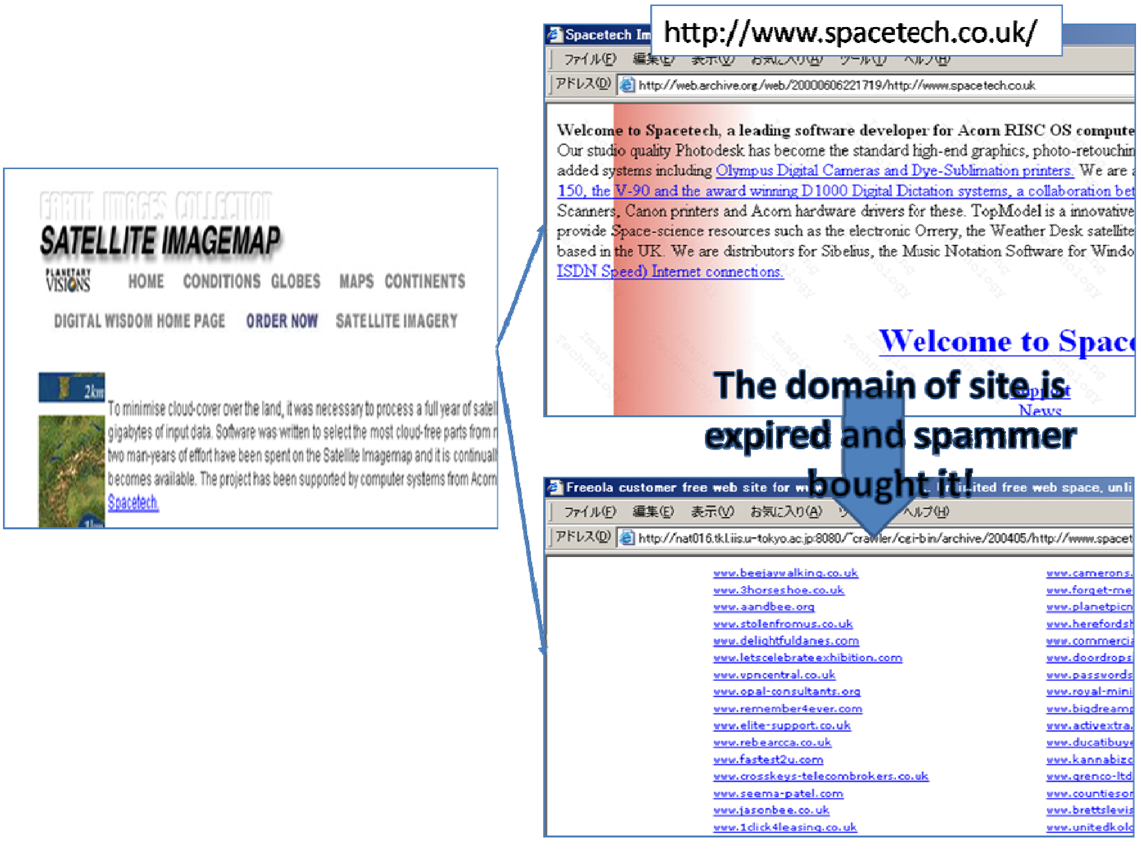
990304
Java 2 SDK, aka JDK 1.2, pre-release 1 is available!

Check the README for more info.

Status Page: http://shell.ncm.com/~sbb/

990301
Linux Java 2 port: JCK status GREEN.

Figure 16 The example of spamming by advertisements



To minimise cloud-cover over the land, it was necessary to process a full year of satell
gigabytes of input data. Software was written to select the most cloud-free parts from n
two man-years of effort have been spent on the Satellite Imagemap and it is continuall
becomes available. The project has been supported by computer systems from Acorn
Spacetech.

http://www.spacetech.co.uk/

http://web.archive.org/web/20000606221719/http://www.spacetech.co.uk

Welcome to Spacetech, a leading software developer for Acorn RISC OS compute
Our studio quality Photodesk has become the standard high-end graphics, photo-retouchin
added systems including Olympus Digital Cameras and Dye-Sublimation printers. We are a
150, the V-90 and the award winning D1000 Digital Dictation systems, a collaboration bet
Scanners, Canon printers and Acorn hardware drivers for these. TopModel is a innovative
provide Space-science resources such as the electronic Orrery, the Weather Desk satellite
based in the UK. We are distributors for Sibelius, the Music Notation Software for Windo
ISDN Speed) Internet connections.

Welcome to Spac

The domain of site is expired and spammer bought it!

http://nat016.tkl.iis.u-tokyo.ac.jp:8080/~crawler/cgi-bin/archive/200405/http://www.spacet

www.beejaywalking.co.uk
www.3horseshoe.co.uk
www.aandbee.org
www.stolenfromus.co.uk
www.delightfuldanes.com
www.letscelebrateexhibition.com
www.vpncentral.co.uk
www.opal-consultants.org
www.remember4ever.com
www.elite-support.co.uk
www.rebearcca.co.uk
www.fastest2u.com
www.crosskeys-telecombrokers.co.uk
www.seema-patel.com
www.jasonbee.co.uk
www.1click4leasing.co.uk

www.camerons.
www.forget-me
www.planetpicn
www.herefordsh
www.commercia
www.doordrops
www.passwords
www.royal-mini
www.bigdreamp
www.activextra.
www.ducatibuye
www.kannabizc
www.grenco-ltd
www.countiesor
www.brettslewis
www.unitedkolc

Figure 17 The example of hijacking by expired domain

4.5. Evaluation

4.5.1 Backward Traversal

Using the white and spam seed sets, we extracted lists of candidate hijacked sites with different δ values from -2.0 to 2.0. (See the algorithm in Section 3.2). After we had the lists, we sorted them in the descending order of Anti-TrustRank scores. We chose Anti-TrustRank score since sites with high Anti-TrustRank scores tend to have many links to spam sites, and such sites can be considered to be influential. We first looked through top 100 sites with high Anti-TrustRank scores and checked them manually whether they are hijacked or not.

Figure 18 shows the number of each hijacking type in the top 100 results using different δ values. We categorized detected samples into hijacked, normal site with direct link to expired-hijacked sites, spam, normal, grey, and finally, unknown. Normal sites contain useful contents and do not employ any web spam techniques. Some sites are judged as grey if they exchange links with other site with relevant contents. However, if the number of sites participating link exchange exceeds 20, we judged these sites as spam. Unknown sites are written in unrecognizable languages like Chinese, German and so on.

We can find 17-30% of hijacked sites. When we consider normal sites pointing to expired hijacked sites as hijacked, our method shows 20-32% of precision. The highest precision was obtained when δ is -2.0

Normal sites increase as δ increases. This is because of that with a higher δ, hijacked sites should have higher white scores. In the same context, as δ decrease, the proportion of spam sites increases. This means we consider sites with relatively higher spam scores as hijacked.

Figure 19 shows the percentage of sites with various hijacking techniques. We can see the most popular technique is blog and bbs hijacking, which is consistent with the result of our classification in section 4.4.

4.5.2 Hijacked Score

A different list of suspicious sites is also generated with algorithm that we described in Figure 5. We checked top 100 sites with high hijacked scores by hand and classified them. The detail is shown in Figure 20 and 21. We categorized detected samples into the same categories as we used for the result of the backward traversal algorithm.

We found out hijacked sites with precision of 34-45%. The precision of algorithm increase 35-47%, when we include normal-link-to-expired. This is much better than a backward traversal. Best precision was obtained when δ is 0.

Figure 21 shows the percentage of sites with various hijacking techniques. We can also see the most frequently used technique is blog and bbs hijacking.

4.5.3 Comparison of Two Methods

Figure 22 and 23 shows the results of a backward walk and hijacked score for top 1000 sites.

We counted how many hijacked samples are in top 1000 sites of each method. These figures confirm that we can discover hijacked sites with the highest precision when we use zero δ and hijacked score. We expect that the precision might increase if we check total sites by hands.

Two methods extract slightly dissimilar hijacked sites. In both methods, the most used technique in top 100 sites is blog and bbs hijacking. However, while a backward walk method extracted more expired sites, a hijacked score method detected more hosting hijacked sites.

We could improve the detection precision if we analysis the reason of the different extractions between two methods.

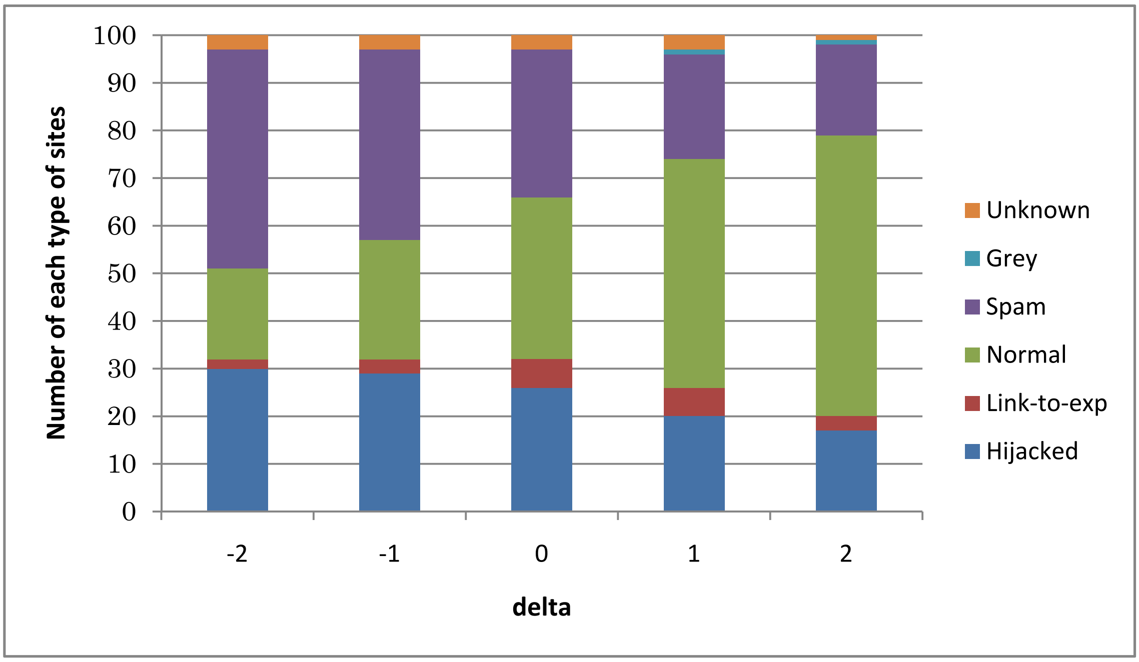


Figure 18 Top 100 precisions of backward traversal

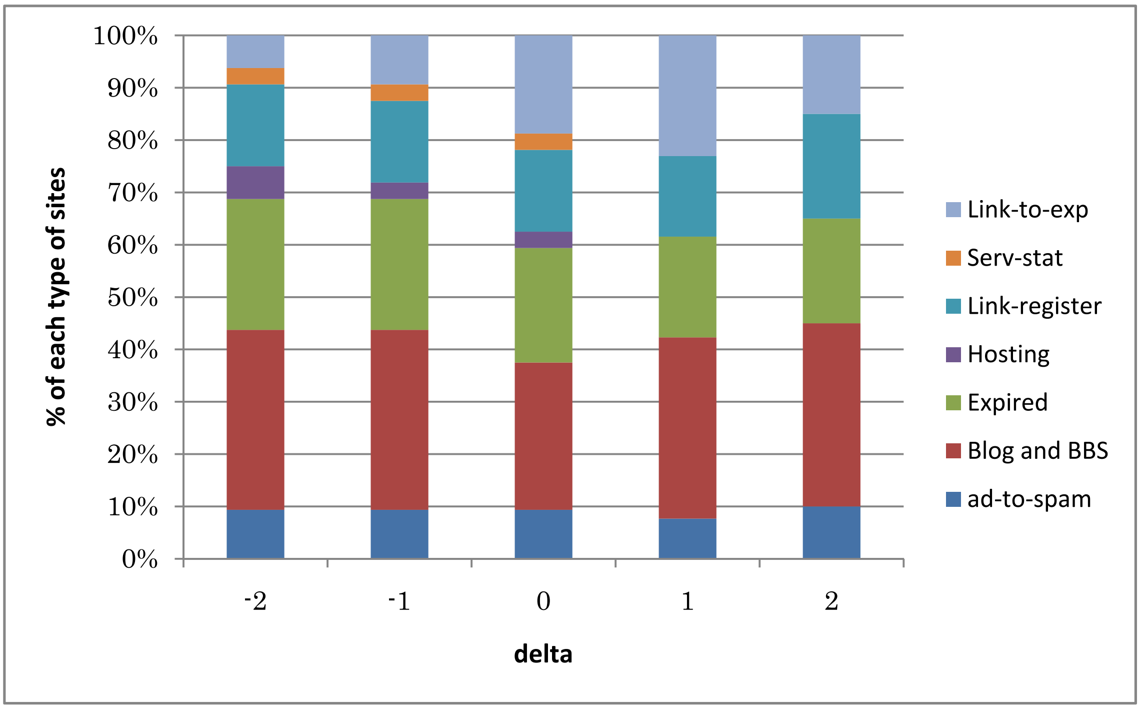


Figure 19 Percentage of each hijacking type

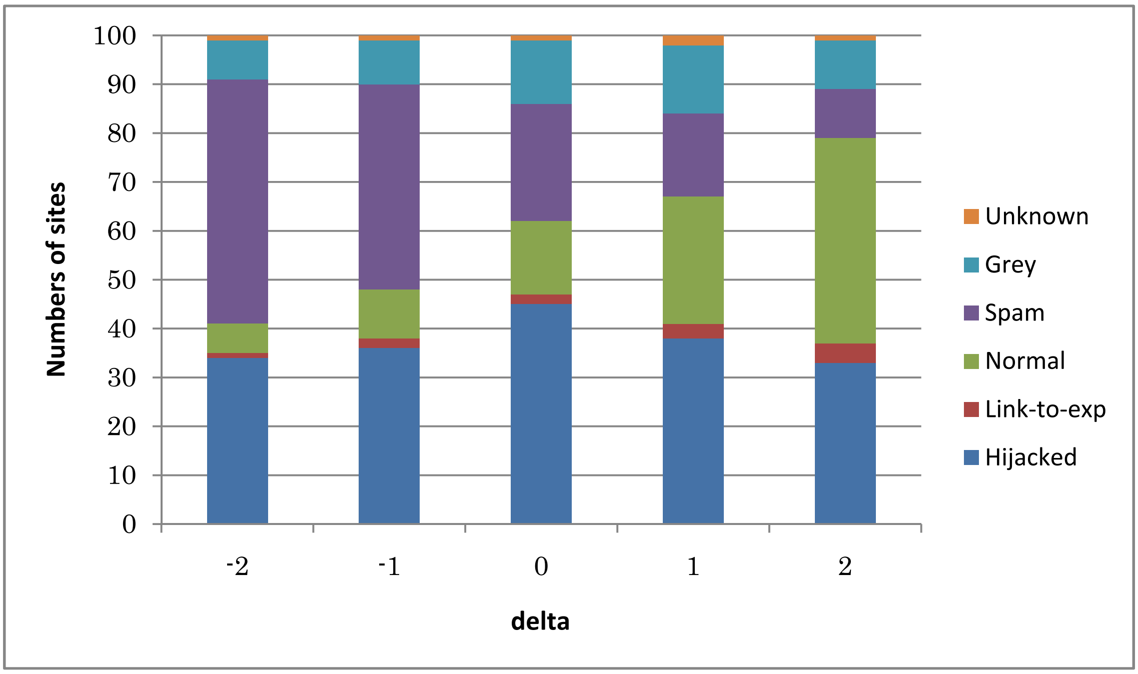


Figure 20 Top 100 precisions of hijacked score

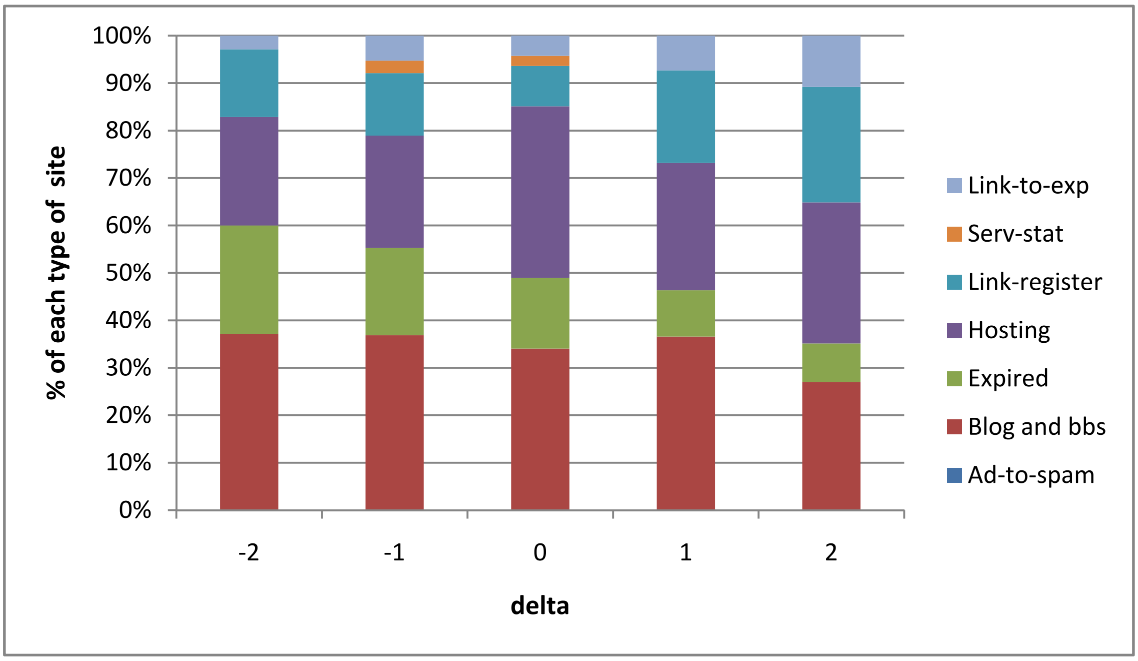


Figure 21 Percentage of each hijacking type

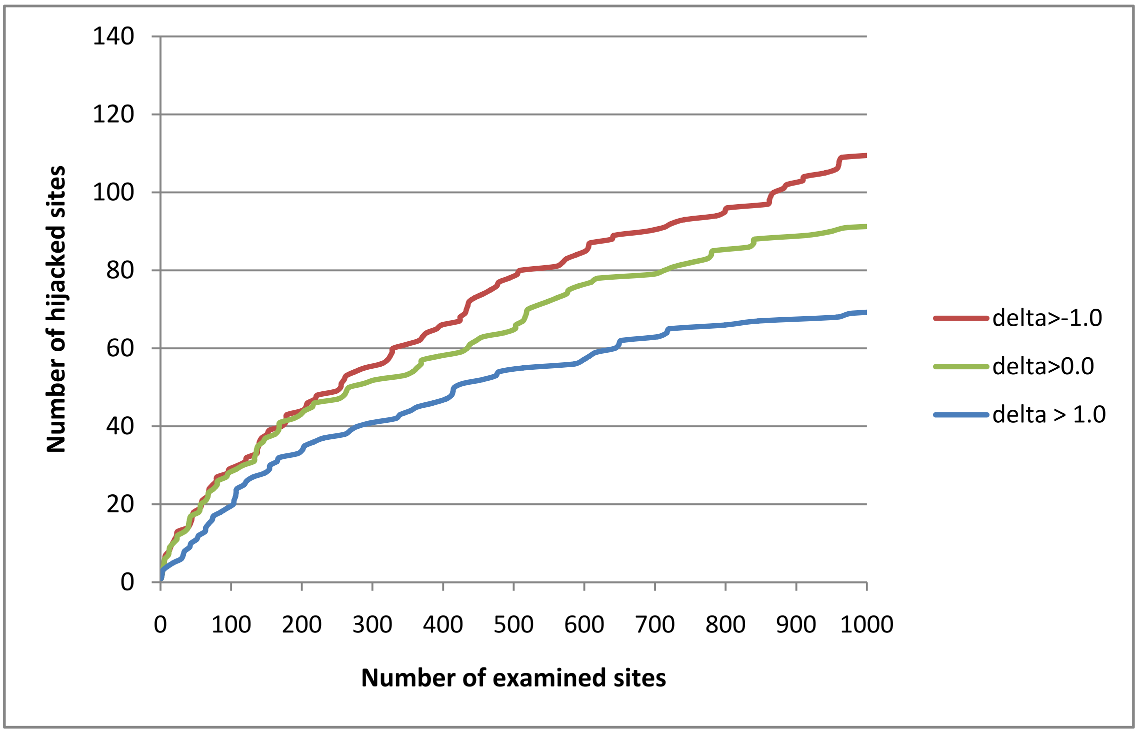


Figure 22 Top 1000 result for backward traversal;

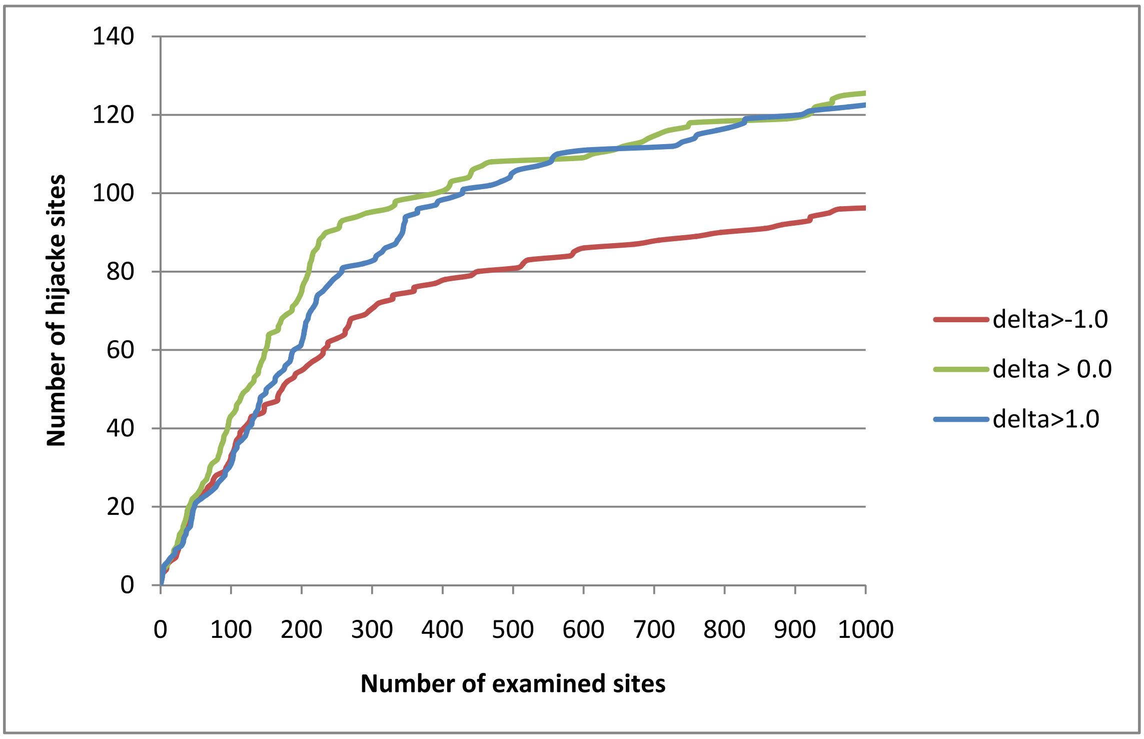


Figure 23 Top 1000 result for hijacked score

4.6 Various Score Distributions of Sites

We also tested our extracting methods with TrustRank and Anti-TrustRank scores instead of core-based PageRank scores. However, it did not show a good performance. In order to clear up the reason of different performances between two score propagation strategies, we investigated score distributions of sites. The results are shown in from Figure 24 to Figure 27.

Figure 24 and 25 shows the score distribution of whole sites. Red, green, blue points correspond to all sites, spam seeds and white seeds, respectively. Purple squares are for sample hijacked sites. With TrustRank and Anti-TrustRank scores, hijacked sites are apt to be judged as spam. Their score distribution is similar to that of spam sites. However, when we look the result of core-based PageRank, hijacked pages are near the boundary (on the diagonal of the graph) between white and spam sites.

Figure 26 and 27 shows the detailed score distribution of hijacked sites. The green points represent hijacked sites, and red points for normal sites which have direct links to expired hijacked site. We can see clearly that there exists a score correlation between core-based PageRank PR+ and PR-. By these figures we can conclude that core-based PageRank is more suitable for our methods.

Note that some sites have relatively high core-based PageRank PR- scores. We looked into those sites and found out that all of them are sites with expired domains. Since spammers employ these sites as a member of a spam farm, their score distributions become similar to those of spam sites.

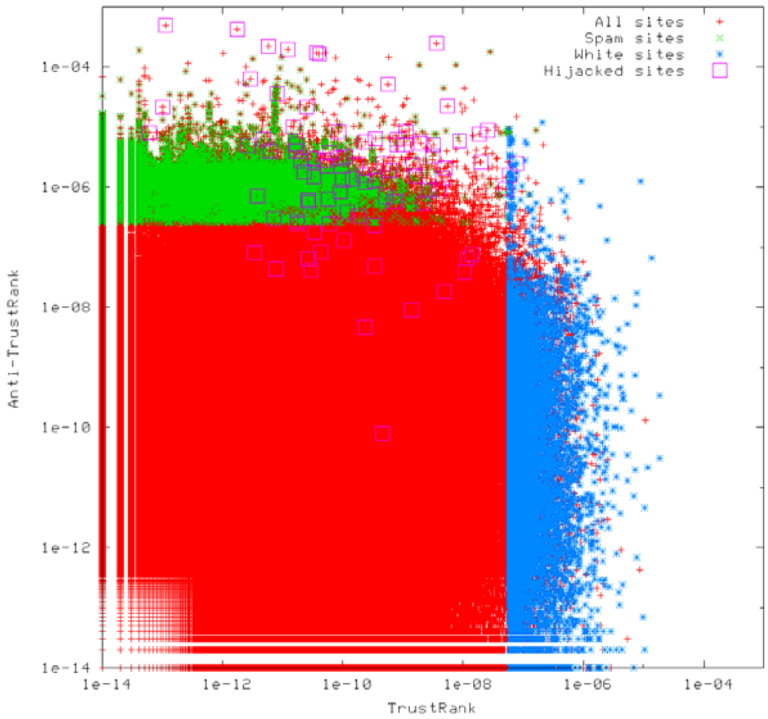


Figure 24 Score distribution of TrustRank and Anti-TrustRank

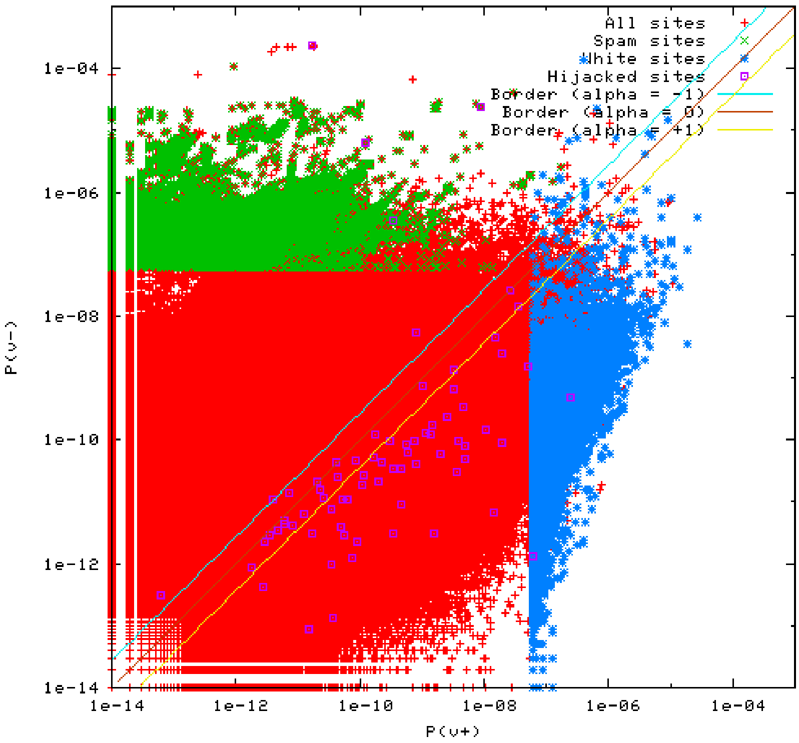


Figure 25 Score distribution of core-based PR+ and PR-

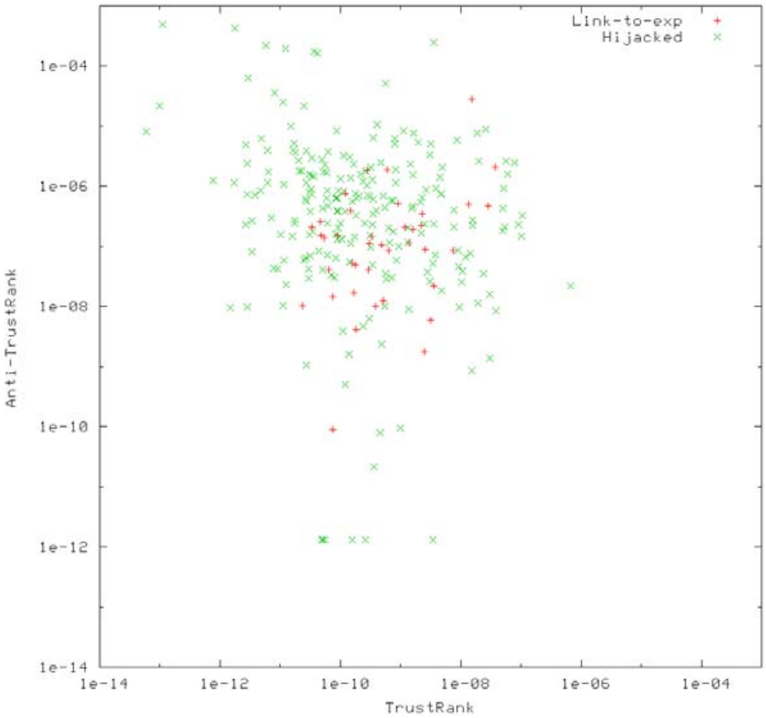


Figure 26 TrustRank and Anti-TrustRank score distribution of hijacked sites

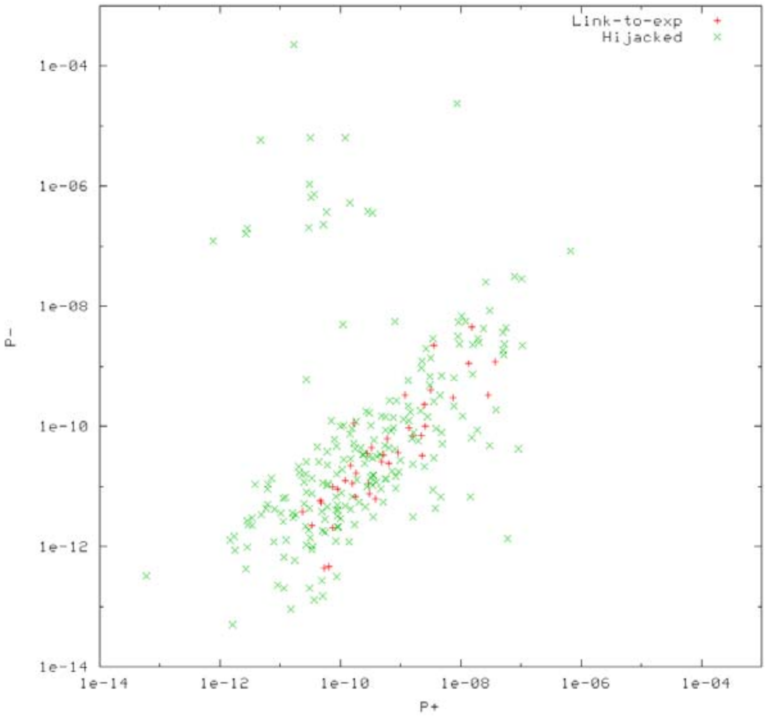


Figure 27 core-based PageRank score distribution of hijacked sites

Chapter 5

Conclusion

In this paper, we proposed new methods for link hijacking detection. Link hijacking is one of the essential methods for link spamming and numerous hijacked links are now being generated by spammers. Since link hijacking has a significant impact on the link-based ranking algorithm and disturbs assigning global importance to, 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 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 a spam seed. Also, we considered the score changes between a normal sites and its out neighbors to detect hijacking.

Experimental result showed that our approach is quite effective. Our best result for finding hijacked sites was 30% with a backward traversal and about 50% with hijacked score.

Chapter 6

Future Work

We can develop our study to detect web spam.

By investigating changes in the structure of spam and hijacked sites with historical snapshots, we could design methods for identifying spam and hijacked sites based on temporal changes of web graph.

Also, we can consider methods to detect newly appeared spam sites using changes around the link structure of hijacked sites.

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] 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.

[9] 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.

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

[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. 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.

[13] 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.

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

[15] 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.

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

[17] J. Caverlee and L. Liu. "Countering web spam with credibility-based link analysis," Proc. the 26th annual ACM symposium on Principles of distributed computing, 2007.

[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

[20] A. Ntoulas, M. Najork, M. Manasse, and D. Fetterly. "Detecting spam web pages through content analysis," Proc. the 15th international conference on World Wide Web, 2006.

[21] A. Singhal, "Challenges in running a commercial web search engine," Proc. the 28th

annual international ACM SIGIR conference on Research and Development in Information Retrieval, 2005.

[22] Dennis Fetterly, Mark Manasse, and Marc Najork. “Spam, damn spam, and statistics: using statistical analysis to locate spam web pages,” Proc. the 7th International Workshop on the Web and Databases, 2004.