Fighting against Web Spam: A Novel Propagation Method based on Click-through Data

Chao Wei, Yiqun Liu, Min Zhang, Shaoping Ma, Liyun Ru, Kuo Zhang
State Key Laboratory of Intelligent Technology and Systems,
Tsinghua National Laboratory for Information Science and Technology,
Department of Computer Science and Technology, Tsinghua University, Beijing, China, 100084
weichao053825@gmail.com, {yiqunliu,z-m,msp}@tsinghua.edu.cn,
lyru@vip.sohu.com, zhangkuo@sogou-inc.com

ABSTRACT

Combating Web spam is one of the greatest challenges for Web search engines. State-of-the-art anti-spam techniques focus mainly on detecting varieties of spam strategies, such as content spamming and link-based spamming. Although these anti-spam approaches have had much success, they encounter problems when fighting against a continuous barrage of new types of spamming techniques. We attempt to solve the problem from a new perspective, by noticing that queries that are more likely to lead to spam pages/sites have the following characteristics: 1) they are popular or reflect heavy demands for search engine users and 2) there are usually few key resources or authoritative results for them. From these observations, we propose a novel method that is based on click-through data analysis by propagating the spamicity score iteratively between queries and URLs from a few seed pages/sites. Once we obtain the seed pages/sites, we use the link structure of the click-through bipartite graph to discover other pages/sites that are likely to be spam. Experiments show that our algorithm is both efficient and effective in detecting Web spam. Moreover, combining our method with some popular anti-spam techniques such as TrustRank achieves improvement compared with each technique taken individually.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval; H.m [**Information SYSTEMS APPLICATIONS**]: Miscellaneous

Keywords

Spam detection, Web search engine, click-through data/bipartite graph, semi-supervised algorithm, label propagation

1. INTRODUCTION

Spamming refers to the malicious attempt to influence the outcome of ranking algorithms, and is usually aimed at obtaining an undeservedly high ranking for one or more Web pages [9]. Castillo and Davison [3] defined Web spam pages as those that benefit from spamming actions, including pages containing inappropriate material, pages with malware or viruses and pages that acquire undeserved traffic by spamming.

There are many reasons for Web spamming. The most important one is that some Web pages try to attract more user visits through search engines without improving content quality or search advertising. According to previous research [19], most users only look through the top results returned by search engines, which means that visitor traffic for a given page or site is highly correlated with its ranking in the results list. The incentive to drive traffic to Web sites as well as the dominant role of search engines is the reason for the ever-increasing amount of Web spam. Most Web spam is created for the purpose of making a profit. In 2005, Singhal [20] estimated that spammers expected to receive a few US dollars per sale for affiliate programs on Amazon, approximately $6 per sale of Viagra, and approximately $20-40 per new member of pornographic sites.

Different techniques are designed to fight against Web spam. State-of-the-art anti-spam techniques use Web page features [17], including both content-based [11] and hyper-link structure-based features [4], to construct Web spam classifiers. Other features, such as features that are extracted from search logs, and browsing logs [4] [5] [12], are also helpful for detecting spam. In fact, we can find spam more accurately by combining all of the signals of a given Web page/site from its content, links, search log features and browsing features [12][17]. In this framework, anti-spam engineers must perform additional work to design specific features and strategies that identify new types of spam by carefully examining characteristics of the spam pages/sites. The biggest limitation of this type of approach is that spammers will develop new spamming techniques immediately after the old tricks are identified. Web spam has evolved from term spamming and link spamming to more sophisticated techniques, such as JavaScript spamming techniques [6].

Existing studies focus mainly on identifying a variety of spam strategies. However, to fight against spam, it is useful and interesting to see how these spam pages attract traffic and make profits. We should notice that spam visiting mainly comes from search engines and its purpose is to draw as much traffic as possible. In other words, they use different spam techniques to cheat search engines, making their pages more “relevant” to specific queries, thus gaining traffic from those queries. These queries will be selected carefully by spammers. Usually, queries with two characteristics are selected. First, they are usually

SIGIR’12, August 12–16, 2012, Portland, Oregon, USA.
Copyright 2012 ACM 978-1-4503-1472-5/12/08 …$15.00.

popular queries or reflect a high demand of users because spammers can gain much traffic if their strategies work. Popular queries are those queries that have high frequencies in search logs. Other queries may not be so popular, but they represent a common information need and thus have the potential to be popular, e.g. “How to lose weight?”, “Are weight-loss pills effective?”. Although different queries may be used, they reflect the same need for weight-loss strategies, which is interesting for many search users. Second, there are usually few key resources or authoritative results for these “spam oriented” queries because spam pages are unlikely to be ranked high in a search results list if there are many relevant and key resources for a given query. For example, few spammers will select the query “yahoo” or “nokia” as spamming contents because there exist authoritative pages for these contents as queries and it is almost impossible for spam pages to be ranked high in such result lists. With the second characteristic, spam pages may be ranked high in the result list, while the first characteristic makes sure the keyword draws much attention. Spamming contents with both characteristics help spam pages to gain user traffic.

Therefore, keywords that 1)are hot or reflect a heavy demand and 2) lack key recourses or authoritative results (we call these keywords spam-oriented queries in this paper) are more likely to be used by spammers and thus lead to spam pages. Web pages/sites that draw traffic mainly from these “spam” queries are more likely to be spam. These two observations indicate that we could use click-through logs to find Web spam. We construct a click-through bipartite graph with query nodes and URL nodes and employ an iterative method to propagate the “spamicity” scores based on the two characteristics of possible spamming contents. We found that a large number of spam pages can be identified by exploiting only the click-through data. This result shows that we could improve anti-spam performance by taking advantage of search logs.

The contributions of this paper are:

1, A label propagation algorithm on click-through bipartite graph is proposed to identify possible spam and its convergence is proven.

2, A thorough analysis of the click-through data is conducted to demonstrate that much can be accomplished with it to identify Web spam. To the best of our knowledge, few publications address the issue of finding spam by using only click-through data.

3, An experimental study on large-scale click-through log data is performed. Results show that the proposed algorithm can identify a variety of spam pages both effectively and efficiently.

The rest of this paper is organized as follows: Section 2 provides a brief review of related work. Section 3 presents our motivation in Web spam detection and formulates the label propagation problem. Section 4 discusses our label propagation problem in detail and proves convergence of our algorithm. In section 5, we perform an experimental validation of our techniques and demonstrate that we can detect spam pages in click-through data both effectively and efficiently. Section 6 gives some conclusion and future work.

2. RELATED WORK

2.1 Web Spamming Techniques and Detection Algorithms

Many spam techniques are emerging along with the development of search engines. Castillo and Davison [3] grouped spamming techniques into two categories: content and link spamming.

Content spamming, including term spamming and content-hidden techniques, refers to techniques that deliberately manipulate page contents and URL keywords to improve their rankings. Gyöngyi et al. [11] provided a list of different types of content spam, including term spam techniques such as repetition, dumping, weaving and stitching and content-hidden techniques such as cloaking [24], redirection spam [6] and visual cloaking [15]. Ntoulas et al. [17] introduced several content-based features to build spam classifiers, and their work is considered to be one of the most influential studies on detecting content spam. They found that spam pages contains more popular terms than non-spam pages. Other studies exploit additional text features to detect spam pages. Linguistic features [18] such as part-of-speech (POS) n-grams, textual features [2], language model features [14] and HTML patterns [22, 23] have been fully studied and were proven to be useful in spam detection.

Link spammers create certain hyper-link structures to boost their scores in typical link analysis, such as PageRank and HITS. Link farms [25], honey pots [11] and spam link exchange belong to this spamming technique. A variety of trust and distrust propagation algorithms such as Trustrank [10] and BadRank [21] and their variants [16] [26] are utilized and proven to be effective in terms of demoting spam. It has also been observed that spam sites often form dense sub-graphs and many works [4] [25] use link-based features, including the degree and spamicity of neighbors to detect them. Recently, Cheng et al. [7] used information from SEO forums to find spam site candidates and thus link farms.

2.2 Usage Data

To detect spam pages more efficiently and effectively, researchers usually combine different spam signals from different usage data, including browsing logs and search logs, to build classifiers.

Liu et al. [13] used browsing logs to estimate the importance of Web pages by defining a continuous-time Markov process on user browsing graph. They showed that their algorithm of BrowseRank is effective in demoting spam sites. Liu et al. [12] proposed a user behavior-oriented Web spam detection framework that was based on browsing logs (captured by a toolbar). They showed that spam sites’ traffic relies almost completely on search engine-originated visits. Other features include the probability that a given page is a source page (i.e. people follow hyper-links on it) and the short-time navigation time (based on the assumption that most Web users would not visit many pages inside a spam Web site).

Search logs contain valuable information about queries and their corresponding URLs. However, there is no detailed analysis on how to utilize these data. Previous studies only created additional features from them [5] [12] [17] for content-based or link-based Web spam detection.

To sum up, state-of-the-art anti-spam techniques use content-based, link-based and user behavior features to construct Web spam classifiers. Anti-spam engineers must perform additional work to design specific features and strategies that can identify new types of spam by carefully examining spamming techniques. The biggest limitation of these approaches is that spammers will develop new Web spam techniques immediately after the old tricks are identified. Due to the fact that spammers often develop new Web spam techniques immediately after the old tricks are identified, the identification of spam and the resultant anti-spam techniques becomes a vicious cycle.

3. MOTIVATION AND PROBLEM FORMULATION

Traditional anti-spam techniques focus on identifying different spam techniques. These techniques fail to exploit how spam pages manage to gain traffic, in other words, how the spammers carefully select the keywords and boost the ranking of their pages/sites in corresponding results lists. The more traffic that spam sites receive, the more profit they will make and the more frustrated search engine users get.

Although spammers want to gain as much traffic as possible from search engines, it is not likely, as discussed in Section 1, for their spam pages/sites to rank high in all of the keywords for which they optimize.

Based on these observations, we design a label propagation algorithm on click-through data. Firstly, a small number of seed pages are selected and labeled as spam or non-spam. Then their labels are propagated on the click-through bipartite graph and other possible spam/non-spam pages are identified. The input consists of a) a set of labeled URLs (spam or non-spam), b) a set of unlabeled URLs and c) a set of constraints between URLs and the queries in the log. The goal is to find spam pages/sites from the unlabeled data.

We first give some definitions before formulating our problem.

Ⅰ. Search engine click-through data C and bipartite graph G.

The click log consists of triples <q, u, f_{qu}>, where q is a query, u is an URL representing a document on the Internet and f_{qu} is the number of times that URL u is clicked when query q is issued. Define Q = {q | q appears in C} and U = {u | u appears in C}. Click-through data C has an equivalent form – a click-through bipartite graph G = (Q, U, E). There are two different types of nodes, queries and URLs in G. For every record <q, u, f_{qu}> in C, there is an edge (q, u) ∈E with weight f_{qu}.

Each q/u is assigned with a probability p_q/p_u, which denotes how likely this q/u is to be a spam query/page or in other words, the spamicity of q/u.

Note that the click-through bipartite graph can be constructed either on page-level or site-level. In the latter form, u is replaced by its site but not the URL of itself. For example, <"Nokia", http://product.pcpop.com/Mobile/00283_1.html, 100> is replaced by <"Nokia", http://product.pcpop.com/, 100>.

Ⅱ. Labeled Seed URL set L.

L contains all of the pages/sites in C(G) that are manually labeled as spam or non-spam. More formally,

L = {u | u is labeled as a spam page/site or non-spam page/site}.

We will discuss the construction details of L in Section 5.2.

Ⅲ. URL result set RU and query result set QU.

RU and QU contain all the <u, p_u> and <q, p_q> pairs, respectively. After our algorithm ends, each URL u or query q in C (or G) will be assigned with a probability p_u/p_q, which denotes the probability that this URL or query is a spam page/site or query. More formally,

RU = {<u, p_u> | p_u is the spamicity score for u}.

RQ = {<q, p_q> | p_q is the spamicity score for q}.

Given G = (Q, U, E) and L⊂U, the goal of the spam pages/sites mining problem is to obtain the results set RU and RQ, which contain all of the possible spam pages/sites and queries in G, respectively.

4. A LABEL PROPAGATION ALGORITHM

4.1 Algorithm design

In this paper, we propose a label propagation (LP) algorithm to solve the problem that is defined in the previous section. More specifically, for every query q, we could calculate the probability p_q that q is a spam query by incorporating all of the label information of its neighbors. Similarly, we could calculate p_u for every URL u. We describe this procedure more formally as follows.

For ∀q/u, we use l_q/l_u to denote its label, which is S for spam and N for non-spam. Note that $P(l_u=N) = 1-P(l_u=S)$. Thus every URL u in labeled set L would have $P(l_u=S)=1$ or $P(l_u=S)=0$ initially and every URL u in the set U-L would have $P(l_u=S)=0$. Then we have

$$P(l_q=S)=\sum_{u:(q,u)\in E}\omega_{qu}P(l_u=S) \quad (1)$$

where

$$\omega_{qu}=\frac{f_{qu}}{\sum_{u:(q,u)\in E}f_{qu}}. \quad (2)$$

ω_{qu} can be interpreted as the transition probability from query q to URL u. It can be drawn from equation (1) and (2) that q's label is determined by all of its neighbors' labels. The bigger ω_{qu} is, the more influence its corresponding node has on determining the label of q.

Similarly, for each URL u in U\L, the probability $P(l_u=S)$ is computed as

$$P(l_u=S)=\sum_{q:(q,u)\in E}\omega_{uq}P(l_q=S) \quad (3)$$

where

$$\omega_{uq}=\frac{f_{qu}}{\sum_{q:(q,u)\in E}f_{qu}} \quad (4)$$

is the transition probability from URL u to query q.

Note that both ω_{qu} and ω_{uq} are not limited to the above form but arbitrary. The only requirement for them is they should have a probability interpretation, which means $\sum_{q:(q,u)\in E}\omega_{uq}=1$ and

$\sum_{q:(q,u)\in E} \omega_{qu} = 1$. We can also interpret ω_{qu} and ω_{uq} as functions of features of queries and URLs and thus incorporate these features to extend our algorithm. We leave it as our future work.

Using Equation (1) and (3), we can obtain $P(l_q=S)$ and $P(l_u=S)$ recursively for all of the queries and URLs in the click-through bipartite graph. We can have a concise representation of this iterative process. Suppose that there are |Q| queries: q_1, $q_2 \ldots q_{|Q|}$ and |U| URLs: u_1, $u_{2 \ldots} u_{|U|}$. Define vectors:

$$\mathbf{P_Q}=(P(l_{q1}=S), P(l_{q2}=S)\ldots P(l_{q|Q|}=S))^T,$$

$$\mathbf{P_U}=(P(l_{u1}=S), P(l_{u2}=S)\ldots P(l_{u|U|}=S))^T,$$

and the transition probability matrixes:

$$\mathbf{M_{qu}}=(\omega_{qu})_{|Q|\times|U|}, \text{ and } \mathbf{M_{uq}}=(\omega'_{uq})_{|U|\times Q}.$$

Then in the ith iteration, we have

$$\mathbf{P^i_Q=M_{qu}\, P^{i-1}_U}$$

$$\mathbf{P^i_U= M_{uq}P^i_Q}$$

It should be noted that in each round of iteration, all of the URLs in seed set L should be re-assigned their initial labels. In this way, the algorithm converges. We will prove the convergence in section 4.4.

4.2 An Example

Consider a sample portion of a bipartite graph from a search engine click log, as shown in Figure 1.

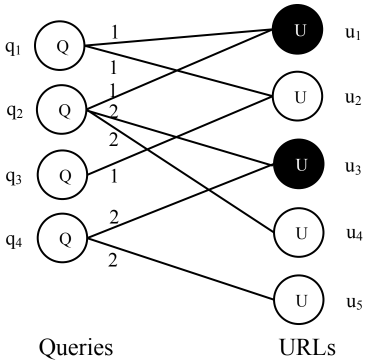

Figure 1. An example to demonstrate how the label propagation algorithm works

Assume that L={u_1, u_3} and both of the URLs are spam pages, as denoted in the shaded circle in Figure 1. Other nodes, including all of the query nodes and the remaining URL nodes, initially have P(l=S) as 0. Then we have

$$\mathbf{P^0_U}=(1,0,1,0,0)^T,$$

$$M_{qu} = \begin{pmatrix} 0.5, & 0.5, & 0 & ,0 & ,0 \\ 0.2, & 0 & ,0.4, & 0.4, & 0 \\ 0 & ,1 & ,0 & ,0 & ,0 \\ 0 & ,0 & ,0.5, & 0 & ,0.5 \end{pmatrix}, \text{ and}$$

$$M_{uq} = \begin{pmatrix} 0.5, & 0.5, & 0 & ,0 \\ 0.5, & 0 & ,0.5, & 0 \\ 0 & ,0.5, & 0 & ,0.5 \\ 0 & ,1 & ,0 & ,0 \\ 0 & ,0 & ,0 & ,1 \end{pmatrix}$$

After the first iteration, we get

$$\mathbf{P^1_Q=M_{qu}\, P^0_U}=\begin{pmatrix} 0.5, & 0.5, & 0 & ,0 & ,0 \\ 0.2, & 0 & ,0.4, & 0.4, & 0 \\ 0 & ,1 & ,0 & ,0 & ,0 \\ 0 & ,0 & ,0.5, & 0 & ,0.5 \end{pmatrix}\begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

$$=(0.5,0.6,0,0.5)^T$$

$$\mathbf{P^1_U= M_{uq}P^1_Q}=\begin{pmatrix} 0.5, & 0.5, & 0 & ,0 \\ 0.5, & 0 & ,0.5, & 0 \\ 0 & ,0.5, & 0 & ,0.5 \\ 0 & ,1 & ,0 & ,0 \\ 0 & ,0 & ,0 & ,1 \end{pmatrix}\begin{pmatrix} 0.5 \\ 0.6 \\ 0 \\ 0.5 \end{pmatrix}$$

$$=(0.55,0.25,0.55,0.6,0.5)^T$$

We notice that here both $P(l_{u1}=S)$ and $P(l_{u3}=S)$ are 0.55, which should be re-set to 1 before the we apply the next iteration.

4.3 The Positive Feedback Problem

Label propagation algorithms or random walks on click-through bipartite graphs have the positive feedback problems. Take u_5 in Figure 1 for example. $P(l_{u5}=S)$ is 0.5 after the first iteration. Because u_3 is a seed URL which are manually labeled as spam, we will set $P(l_{u3}=S)$ to be 1 before the second iteration begins. Therefore, it is easy to see that $P(l_{u5}=S)$ is 0.75 after the second iteration and converges to 1 if the algorithm continues. The reason is that edge e = <q_4, u_5> is undirected and u_5 is a 1-degree node, which means that score of u_5 will flow back to q_4; from this process, it obtains its original spam score. We call this effect the positive feedback problem which would magnify the noise in the click-through bipartite graph and distort the final results.

Consider an extreme case given in Figure 2.

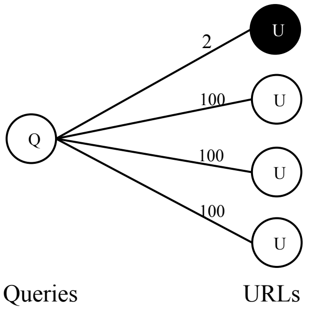

Figure 2. An extreme case for the positive feedback problem

Suppose that a non-typical user issues a query q to the search engine and then clicks a spam page while most of the other users use this query to navigate to high quality sites. All of the spam scores of the URLs will converge to 1 after our algorithms is applied on the graph, which contradicts the real explanation of the pages and this query.

It can be concluded from Figure 1 that this problem mainly affects those 1-degree nodes, such as u_4, u_5 and q_3. To solve this problem, we introduce the concept of confidence. We define the confidence of a node in the bipartite graph as a function of its degree d. More formally,

$$c(q)=f(d_q),\ c(u)=f(d_u),$$

where d_q and d_u are the degrees of node q and u, respectively. Then Equations 1 and 3 can be revised as follows.

$$P(l_q=S)=\sum_{u:(q,u)\in E}\omega_{qu}c(u)P(l_u=S)=\sum_{u:(q,u)\in E}\omega_{qu}P'(l_u=S)$$

$$P(l_u=S)=\sum_{q:(q,u)\in E}\omega_{uq}c(q)P(l_q=S)=\sum_{q:(q,u)\in E}\omega_{uq}P'(l_q=S)\cdot$$

It should be noted that function f here is arbitrary and we have many choices for function f. For simplicity, we use the indicator function in this paper.

Define

$$f(d)=\begin{cases}0\ ,\ if\ d=1\\ 1\ \ ,else\end{cases}.$$

An intuitive interpretation for the indicator function is that before we obtain sufficient information to judge a 1-degree node if it is a spam page, we could treat it in the bipartite graph as pseudo normal page, forcing its spam score to be 0. Note that this constraint only applies to the unlabeled nodes since we need the seed-nodes to propagate their spamicity.

Now, applying the revised algorithm to Figure 2, we obtain $P(l_{q3}=S)$ = 2/302 and $P(l_{u3}=S) = P(l_{u4}=S) = P(l_{u5}=S)$ = 2/302, which are more reliable than the original algorithm.

The outline of the Label Propagation algorithm is shown in Figure 3.

```
The Label Propagation Algorithm:
Input：labeled seed set L，click-through data C(G)
Output：$P(l_u=S)$ and $P(l_q=S)$ for all URLs and queries in G
Begin
    Do
    for u∈L, set $P(l_u=S)=1$ or 0 according to their label by human assertors.
        for all q∈Q do
            $P(l_q=S)=\sum_{u:(q,u)\in E}\omega_{qu}P'(l_u=S)$
        end for
        for all u∈U\L do
            $P(l_u=S)=\sum_{q:(q,u)\in E}\omega_{uq}P'(l_q=S)$
        end for
    until convergence

    Output $P(l_u=S)$ for every URL u in U and $P(l_q=S)$ for every query q in Q
End
```

Figure 3. The label propagation algorithm

4.4 Convergence of the LP Algorithm

It is evident that $\mathbf{M_{qu}}$ and $\mathbf{M_{uq}}$ are right stochastic matrixes, each of whose rows consists of nonnegative real numbers, with each row summing to 1. Then consider $\mathbf{M_{uu}=M_{uq}M_{qu}}$. For each element m_{ij} in $\mathbf{M_{uu}}$, we have $m_{ij}=\sum_k\omega_{ik}\omega'_{kj}$ in $\mathbf{M_{uu}}$, where ω_{ik} and ω'_{kj} are elements $\mathbf{M_{uq}}$ and $\mathbf{M_{qu}}$, respectively. Thus we have

$$\begin{aligned}\sum_j m_{ij}&=\sum_j\sum_k\omega_{ik}\omega'_{kj}\\&=\sum_k\sum_j\omega_{ik}\omega'_{kj}\\&=\sum_k\omega_{ik}\sum_j\omega'_{kj}\\&=\sum_k\omega_{ik}\\&=1\end{aligned},$$

which means that $\mathbf{M_{uu}}$ is also a right stochastic matrix.

Now, if we are only interested in $\mathbf{P_U}$, the iteration process can be rewritten as

$$\mathbf{P^i_U= M_{uu}P^{i-1}_U= M_{uq}M_{qu}\ P^{i-1}_U,}$$

where i denotes the iterations.

Suppose that there are |L| seed URLs in L, |C| 1-degree nodes and thus r = |U|-|L|-|C| remaining URLs in C. More specifically, let the probability vector $\mathbf{P_U}$= （$\mathbf{P_T}$ $\mathbf{P_L}$） where $\mathbf{P_T}$ are the top |L|+|C| rows of $\mathbf{P_U}$(the labeled data and the pseudo labeled data) and $\mathbf{P_L}$ are the remaining r rows of $\mathbf{P_U}$(the unlabeled data). We split $\mathbf{M_{uu}}$ after the (|L|+|C|)th row and the (|L|+|C|)th column into 4 sub-matrixes

$$M_{uu}=\begin{pmatrix}M_{(|L|+|C|)(|L|+|C|)} & M_{(|L|+|C|)r}\\ M_{r(|L|+|C|)} & M_{rr}\end{pmatrix}.$$

Note that $\mathbf{P_T}$ never really changes. It can be shown that in our algorithm, $P_L=M_{rr}P_L+M_{r(|L|+|C|)}P_T$, which leads to $P_L=lim_{n\to\infty}M^n_{rr}P^0_L+[\sum_{i=1}^n M^{i-1}_{rr}]M_{r(|L|+|C|)}P_T$. Zhu and Ghahramani [27] proved that $\mathbf{P_L}$ converges to $(I-M_{rr})^{-1}M_{r(|L|+|C|)}P_T$ if $\mathbf{M_{uu}}$ is a right stochastic matrix. Thus the initial value of $\mathbf{P_L}$ is inconsequential.

Using the same approach, we could prove that $\mathbf{P_Q}$ also converges.

5. EXPERIMENT EVALUATION

The goal of the experiments is to evaluate how effective our algorithm is in detecting spam Web sites. Given a seed set L, the LP algorithm returns a list of pages/sites that are sorted according to their probability of being spam. Seed page/sites are not included in the list. We also obtain a list of queries that are sorted according to their probability of being used as a spam-oriented query. A detailed discussion of the queries and sites is in Section 5.6.

5.1 Bipartite Graph Construction

We collected query logs from March 1st, 2011 to March 9th, 2011 with the help of a famous commercial search engine company in

China. No private information was included in these logs. We pruned all of the query-URL pairs with just one click on any day in the log since they may contain noise and possible privacy information. After that, this click-through log consisted of 8,443,963 unique queries, with 12,470,865 unique URLs in 1,055,001 sites. Altogether, 17,660,907 query-URL pairs were collected and they were used in constructing the bipartite graph. The maximal component of the graph contains 2,111,135 (25.0%) unique queries, 3,614,514 (29.0%) URLs and 7,805,300 (44.2%) query-URL pairs. An interesting observation is that the second largest connected component contains only 326 queries, which is a much smaller number compared with the first component. It does not make sense to run the propagation process on such small connected graph. Therefore, we focus on the maximal component of the graph.

Although our algorithm can be applied on both the site-level and the page-level with click-through data, we applied it on the former mainly because of the sparsity problem. A single page usually has fewer queries or even only one query that is pointing to it, which makes it vulnerable to noise.

5.2 Seed Set Selection

5.2.1 Spam Seeds Selection

Seed set contains labeled sites for our LP algorithm. With the help of a famous commercial search engine in China, we obtained a spam site list. A total of 2,100 of these sites appear in our click-through data, and we use them as the spam section of seeds for our algorithms.

5.2.2 Non-spam Seeds Selection

We manually select several ordinary web sites as non-spam seeds according to two criteria. Firstly, these web sites should be famous. There are two reasons for this requirement: a) the more famous the site is, the less likely it will contain spam and b) these sites contain numerous key-resources and we need them to judge the spamicity of the queries. Secondly, sites that mainly consist of user-generated contents are removed because the quality of these contents cannot be guaranteed. For example, http://news.sina.com/ is a non-spam site because most of the pages in it are news articles. The contents will be posted after human examination. In contrast, http://bbs.sina.com.cn/ is not included in the seed set because it is a forum and its contents are not highly reliable. We manually selected 1,153 sites as non-spam section of the seeds.

At last, we constructed the seeds for the algorithm, including 2,100 spam sites and 1,153 non-spam sites. These two sites lists are available at http://www.thuir.cn/weichao/sigir2012/.

5.3 Performance Comparison

We have implemented the LP algorithm on the click-through bipartite graph constructed from search logs mentioned above, and we name it as LP.

Although the algorithm converges, it is not easy to decide when to end the algorithm. We observed that the spam probability changed little after 20 iterations. Specifically, in our experiment, there was little difference between the results after 20 iterations and 40 iterations. Thus, we ran the iteration process 20 times and then output the results.

We use two baselines in the experiment, namely PageRank and TrustRank. PageRank is widely used in ranking search engine results and TrustRank proves to be effective at detecting Web spam. With the help of the same commercial search engine, we constructed a Web link graph that contained 258,326,221 sites. For each hyperlink in page A that points to page B, we added a directed edge from A's site to B's site. Thus the link graph contains 4,883,760,072 edges. We implement the two methods that are completely based on this Web link graph, and they can be regarded as a representative of conventional content-independent anti-spam methods in a real Web environment. Also, we used the common α value of 0.85 in the implementations. We denote the above two baselines as PR and TRUST respectively.

Content-based anti-spam methods were not used in this paper because most of them work only for specific spam techniques. We can expect that they will not achieve a good performance because a variety of spam techniques are used by real spam sites.

Because our label propagation algorithm adopts a different type of information from state-of-the-art anti-spam techniques, it is interesting to combine these algorithms together, and we expect that this combination will achieve better performance than either algorithm. We use the method that is proposed in [1] to combine the results of different algorithms. Suppose L_u is the rank of site u in test set sorted by spamicity score obtained from the LP method, and O_u is the rank obtained from another anti-spam algorithm. Both the result lists are sorted in descending order by spamicity. Note that PageRank/TrustRank result is sorted by scores in ascending order since a lower PageRank/TrustRank score implies this site is more likely to be spam. The merged score S is calculated as follows:

$$S(\mathrm{u},\omega)=\omega\cdot\frac{1}{\mathrm{L}_{\mathrm{u}}+1}+\frac{1}{O_{\mathrm{u}}+1},$$

where ω represents the importance of the LP algorithm's score. We set ω =1, indicating that both algorithms are equally important in the final rank calculation. Then the result list of LP is combined with PR and TRUST and we name them LP-PR and LP-TRUST respectively.

We ran the five algorithms and compared their performances on the test set with respect to the precision, recall, and AUC value, which are all common-used evaluation metrics for spam detection in previous studies [5][12].

5.4 Test Set and Labeling Criterion

Human annotators were recruited to label the URLs list that was returned by the algorithms. However, such labeling is not a trivial task. As discussed in Section 5.1, there are more than 1 million sites in the bipartite graph and it is therefore impossible to label all of them. Since we concern performance of the LP algorithm the most, we selected 3,000 sites uniformly from top half part of the LP results list and used them as our test set. The reason to focus on the top half of the LP results list is that the cost of mislabeling a reputable site as spam is much higher that the opposite. Two experienced human experts were asked to make spam judgments for them, according to the labeling criterion of Web Spam Challenge [28]. Based on the instructions, assessors will have four options for each site they have to tag:

- NONSPAM - The site does not contain spamming aspects.
- BORDERLINE - The site contains some aspects that

are suspicious of being spam.

- SPAM - The site contains spamming aspects.
- CAN'T CLASSIFY - The assessor could not classify the site.

It should be noted that we adopt a stricter judgment on spam sites when “BORDERLINE” and “CAN'T CLASSIFY” sites are labeled as NONSPAM. The reason is that the cost of mislabeling a normal site as spam is much higher than the opposite.

Another problem is that some of the sites may disappear from the Web for different reasons and could no longer be accessed. For these sites we send the URLs to a commercial search engine (http://www.sogou.com/) and obtain their snapshots. Then, they are labeled as “Spam”, “Non-spam” or “Can’t Access” based on these snapshots. Those whose snapshots did not exist were labeled as “Can’t Access”.

Each annotator was asked to label 1500 sites. Moreover, they were asked to label another 150 sites to evaluate their agreement. The Cohen’s Kappa value is 0.856, which suggests a perfect agreement between the annotators.

Of all of the 3,000 sites, 1,490 are labeled as NONSPAM, 870 are labeled as SPAM and the remaining 640 sites can’t be accessed now. We removed all of the inaccessible sites and used the 2,360 labeled sites as our test set (further information about these sites is available at http://www.thuir.cn/weichao/sigir2012/).

5.5 Experiment results

We performed the five algorithms and compared their performances on the test set by precision, recall, and AUC value. The experimental results are shown in Figure 4 and 5.

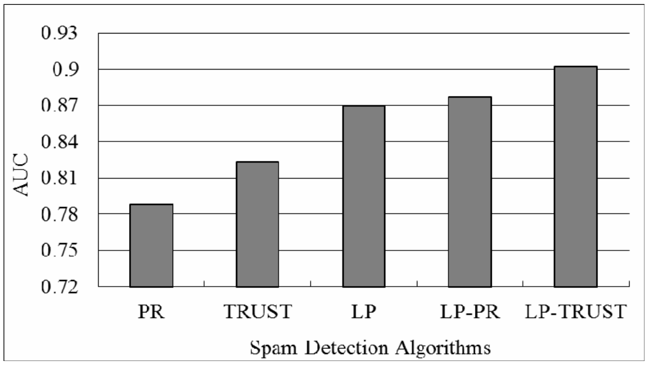

Figure 4. AUC value for different spam detection strategies

From the figures, we can see that all of the AUC values of the five algorithms are greater than 0.78, which suggests that they are effective in detecting Web spam. It is not surprising that PageRank performs the worst because spam sites can boost their PageRank scores using tricks such as the link-farm. TRUST works better than PR, which is consistent with previous research [10]. The AUC value of LP is 0.870, which is much better than both PR (0.788) and TRUST (0.823). This demonstrates our algorithm is effective in detecting Web spam sites. It is encouraging to see our LP algorithm performs better than TrustRank while the former is also less time-consuming. In our experiment, the LP algorithm based on our collected click-through data converged in less than 20 minutes. However, since Trustrank and Pagerank were performed on a complete Web graph that usually contains much more nodes than a click-through bipartite graph, it took more time for both of them to converge, e.g. 10 hours in our experiment.

LP-PR receives a higher AUC value (0.877) while the LP-TRUST approach is impressive in boosting the performance of Spam detection, which obtains a much higher AUC value (0.902) than all of the other algorithms. The precision of this approach can achieve as much as 90% with a recall of 50% and as much as 84.5% with a recall of 70%.

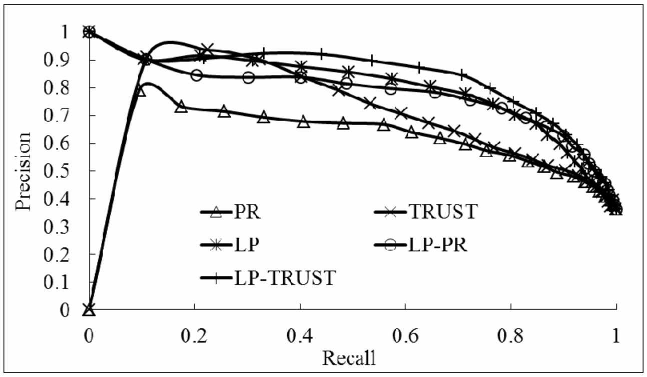

Figure 5. Precision-Recall Curve for different spam detection algorithms

We also conduct another experiment to see how robust our algorithm is. It is known that seed selection is truly important in semi-supervised algorithms such as TrustRank. We randomly split our Spam sites into 21 subsets (each with 100 seed sites) and then add them gradually into the seed set. The experiment results are summarized in Figure 6.

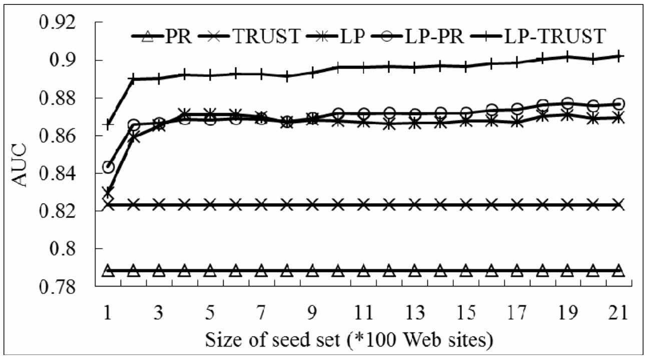

Figure 6. Algorithm performance with different seeds set

It can be seen that all of our algorithms are very robust. They can achieve a relatively high AUC value after only 400 sites are added into the seed sets. We also notice that LP performs consistently better than PR and TRUST. The AUC value of LP-PR is slightly smaller than LP when the size of spam seed sites is between 400 and 700. However, it consistently outperforms LP when more sites are added. LP-TRUST is always the best of all the algorithms.

One-side paired Wilcoxon-tests shows that LP, LP-PR and LP-TRUST are significantly better than TRUST and PR. LP-TRUST is significantly better than other methods. The detailed result is listed in Table 1.

Table 1 One-side paired Wilcoxon-test result (p-value indicates whether algorithm in the row is significantly better than algorithm in each column.)

Wilcoxon-test (p-value)	PR	TRUST	LP	LP-PR
TRUST	2.525e-06	NA	NA	NA
LP	3.206e-05	3.710e-05	NA	NA
LP-PR	3.206e-05	3.206e-05	0.000543	NA
LP-TRUST	3.206e-05	3.206e-05	3.206e-05	3.206e-05

The experiment shows that our novel algorithm is successful in detecting spam sites. Moreover, because we derive this algorithm from a totally different perspective from current anti-spam techniques, combining it with state-of-the-art techniques will result in a more powerful approach for detecting Web spam.

5.6 Discussions

5.6.1 Query analysis

We then conducted several experiments to see how our algorithm detects Web spam sites and why combining LP and TrustRank results in a significantly better performance. We selected the top 1,000 spamming queries and classified them into 7 categories manually: Porn, Game, Health, Entertainment, Software, Lottery and Others. The detailed composition is shown in Figure 7.

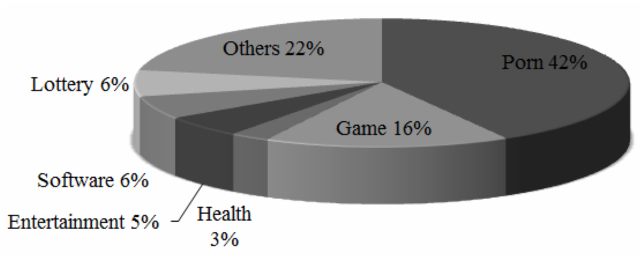

Figure 7. Composition for top spammy queries

We can see from Figure 7 that 42% of these queries are porn terms. Liu et al. [12] find that many of the queries that lead to spam site are porn terms. Spammers use these terms for optimization usually because they are always popular in search engine logs. Moreover, these queries often lack key-resources, which are "reputable" porn sites, because porn contents are illegal in many countries. This contradiction makes porn terms the first choice for spammers.

We notice that game related queries come at the second place, accounting for 16% of all of the 1,000 queries. This can be explained by the fact that quite a lot Internet users are relatively young and are prone to accessing online games. It is not surprising that they use the search engines to find the games. However, in contrast to queries that lead to reputable online games servers, most of these queries have illegal intents. For example, 64.2% of these game-related queries are about "private servers" for online games. These servers are illegal because they emulate online games without warranty. We find that sites that are targeted by these illegal queries usually adopt spam techniques. In contrast, ordinary game sites are not willing to take the risk using spam techniques because they will be put on the blacklist and thus cannot draw any traffic from search engines. Usually, they will adopt other strategies, such as sponsor searches, to draw the attention of potential users. However, things are different for illegal sites because search engines seldom provide services for them. For this reason, they turn to spam techniques. For example, we issued "私服", which means "private server" in Chinese to Google (http://www.google.com/) on October 28th, 2011 and found that almost all of the top 10 sites except for the 5th and the 10th are employing some spamming techniques. In fact, the 5th result is an entity of "私服" in an online encyclopedia site, similar to wikipedia.com and the 10th result is a news page of a famous Web site. This result suggests that illegal web sites are likely to use spam techniques. We list their URLs and spamming techniques in Table 2.

We conducted similar investigations on queries in other categories and find a similar conclusion (we do not list them here because of the space restrictions) that most of these queries meets the two criteria discussed in Section 1. First, they are hot or reflect a heavy demand of search engine users, such as online games or maintaining fitness. Secondly, they lack key resources or authoritative results. Several of these queries are even illegal.

Table 2 Top 10 results for query "私服"

Rank	URL	Spam or not
1	http://www.7774f.com/	Spam (dumping and weaving)
2	http://www.52dayu.com/	Spam (dumping, weaving and redirection)
3	http://www.cnnds.com/	Spam (repetition and stitching)
4	http://www.shiqi.cc/	Spam (link exchange)
5	http://baike.baidu.com/view/6975.htm	Not a spam (an online encyclopedia site)
6	http:// www.20zf.com/	Spam (link exchange)
7	http://www.wnlzj.com/	Spam (weaving and stitching)
8	http://www.luosisa.com/	Spam (weaving)
9	http://www.ipput.com/	Spam (weaving and stitching)
10	Vertical search result (News)	Not a spam

Previous research [5] suggests that monetizable queries are more likely to be spam keywords. However, we find that popular queries, such as porn terms, as well as queries with illegal intents are also more likely to be spam keywords. Two reasons can explain this. First, many spammers try to attract traffic to their sites so that they can increase revenue from advertisers instead of making profit directly from users by selling products. The more traffic or PageRank score their sites have, the more money the advertiser will pay them. Thus they try to gain traffic from popular queries and sometimes succeed in queries with little key-resources. Second, as for the illegal Web sites, they cannot improve their ranks in search engines by regular techniques such as sponsor search, which leads them to spam techniques.

5.6.2 Site analysis

We next examined the characteristics of the spam sites. When looking at the top ranked sites by the spam scores given by LP algorithm, we found that most of them contained illegal contents, such as pornography, private servers and online gambling, and that most used different spam techniques to draw traffic. Our algorithm can find them successfully because queries leading to these sites are more specific, containing porn terms or words such as "private server". Unlike porn sites, which mainly benefit from repetition and keywords stuff, private servers or online gambling sites often use dumping, weaving and stitching to attract traffic. Moreover, these sites often link to each other and hope to benefit from link-exchange. For example, http://www.uiop8.com/ uses almost all of the spam techniques that are listed above.

Other spam sites that we detected are more "general". Most of them would select a topic that they want to optimize and try to create as much spam content as possible, which appear to be relevant to as many queries as possible in this topic. The site http://www.hywww8.com/ is an example.

We also find that spam sites with few queries and clicks are more likely to be spam. Most of the top spam sites in the result list have only 1 query and 2 clicks (notice that 2 is the minimum number of clicks in our pruned click-through logs). This is consistent with our intuition. Reputable sites usually cover more topics and have more user clicks. As a result, when a site is clicked after a spammy query and it seldom appears in the click-through data which means it has few non-spammy queries, we have a bigger confidence that this is a spam site.

5.6.3 Combining label propagation with TrustRank

Although our method can discover a variety of spam sites, they have its own limitations. It is difficult to label ordinary sites with a rare number of queries. The main problem is that there is little information about these sites in the click-through data. However, other anti-spam techniques, such as TrustRank, could provide us with information about how trustworthy a site is, which could be used to boost the performance of our algorithms. In contrast, TrustRank also has its limitation. http://www.17646.com/, a private server for online games, manages to obtain a high TrustRank score, ranking the 86,000th of all the 258,326,221 sites. Therefore, TrustRank fails to identify it as a spam site. However, it ranks the 80th for its spam score in our algorithm. By combining its spam rank and trust rank, we rank it at the 176th place in the final results list, which means that it is likely to be a spam site.

Because we use a completely different method to detect Web spam, we expect to obtain a much better performance by combining it with other state-of-the-art algorithms. We leave this promising task for our future work.

6. CONCLUSION AND FUTURE WORK

In this paper we have proposed a novel label propagation algorithm on click-through bipartite graphs to detect Web spam. Different from current approaches which focus on identifying predefined types of Web spam pages/sites, this algorithm exploits the characteristics of spam queries. The spamisity score propagates between queries and URLs iteratively on the click-through bipartite graph from a seed set that contains both spam and non-spam pages/sites. Experiment results show that our algorithm is both efficient and effective in detecting Web spam pages/sites. This algorithm introduces a novel perspective to fight against Web spam, and combining it with some current anti-spam techniques results in a much better performance.

For future work, we plan to investigate the following aspects. Firstly, we will attempt to solve the sparsity problem that was discussed in Section 5 by collecting more click-through data. We also notice that there are several small connected components except for the maximal one in the click-through log. Are there spam sites? How many spam sites exist in these components? We will try to answer these questions in future work. Secondly, our framework will be extended to embody other features, including content, hyperlink features and click-through features. For example, we found that even for those popular but not spammy queries, sites with few clicks are also suspicious. Query "戴尔官方网站" ("DELL official Website" in Chinese) appeared 6,159 times in the log and 39 different sites were clicked after it. Among these sits, spam site http://www.buydellonline.cn/ appeared only twice. Moreover, it appeared only 4 times in the click-through data. In contrast, http://www.dellenglish.com/ which appeared only 3 times after this query but had 38 clicks in query "English study", turned to be an ordinary site. This indicates such features can be used to boost the LP algorithm performance. Moreover, we notice that while some queries are generally more likely to be spam queries, some other queries are not likely to be spam queries, such as the name of a small company. Query taxonomy information is also a helpful feature. We will incorporate such information in our propagation process in our future work, too.

7. ACKNOWLEDGMENTS

This work was supported by Natural Science Foundation (60903107, 61073071), National High Technology Research and Development (863) Program (2011AA01A205) and Research Fund for the Doctoral Program of Higher Education of China (20090002120005).

8. REFERENCES

[1] Agichtein, E., Brill, E. and Dumais, S. 2006. Improving web search ranking by incorporating user behavior information. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval* (Seattle, Washington, August 6-11, 2006).SIGIR '06. ACM, New York, NY, 19-26.

[2] Attenberg, J. and Suel, T. 2008. Cleaning search results using term distance features. In *Proceedings of the 4th International Workshop on Adversarial Information Retrieval on the Web* (Beijing, China, April 22, 2008). AIRWeb '08. ACM, New York, NY, 21-24.

[3] Castillo, C. and Davison, B.D. 2011. *Adversarial Web Search*. Foundations and trends in Information Retrieval. 4, 5 (2011), 377-488.

[4] Castillo, C., Donato, D., Gionis, A., Murdock, V. and Silvestri, F. 2007. Know your neighbors: Web spam detection using the web topology. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (Amsterdam, The Netherlands, July 23-27, 2007). SIGIR '07. ACM, New York, NY, 423-430.

[5] Chellapilla, K. and Chickering, D.M. 2006. Improving cloaking detection using search query popularity and monetizability. In *Proceedings of the 2nd International*

Workshop on Adversarial Information Retrieval on the Web (Seattle, Washington, August 10, 2006). AIRWeb '06. ACM, New York, NY, 17-24.

[6] Chellapilla, K., and Maykov, A. 2007. A taxonomy of JavaScript redirection spam. In *Proceedings of the 3rd International Workshop on Adversarial Information Retrieval on the Web* (Banff, Alberta, Canada, May 8, 2007). AIRWeb '07. ACM, New York, NY, 81-88.

[7] Cheng, Z., Gao, B., Sun, C., Jiang, Y. and Liu, T. 2011. Let Web Spammers Expose Themselves. In *Proceedings of the fourth ACM international conference on Web search and data mining* (Hong Kong, China, February 9-12, 2011). WSDM '11, ACM, New York, NY, 525-534.

[8] Erdélyi, M., Garzó, A. and Benczúr, A.A. 2011. Web spam classification: a few features worth more. In *Proceedings of the 2011 Joint WICOW/AIRWeb Workshop on Web Quality* (Hyderabad, India, March 28, 2011). WebQuality '11, ACM, New York, NY, 27-34.

[9] Gyöngyi, Z. and Garcia-Molina, H. 2005. *Spam: It's Not Just for Inboxes Anymore.* IEEE Computer Magzine. 38, 10 (2005), 28-34.

[10] Gyöngyi, Z., Garcia-Molina, H. and Pedersen, J. 2004. Combating Web Spam with TrustRank. *In Proceedings of the 30th International Conference on Very Large Data Bases* (Toronto, Canada, August 29 – September 3, 2004). VLDB '04. VLDB Endowment, US, 576-587.

[11] Gyöngyi, Z. and Garcia-Molina, H. 2005. Web spam taxonomy. In *Proceedings of the First International Workshop on Adversarial Information Retrieval on the Web* (Chiba, Japan, May 10, 2005). AIRWeb '05. ACM, New York, NY, 39-47.

[12] Liu, Y., Cen, R., Zhang, M., Ma, S., and Ru, L. 2008. Identifying Web spam with user behavior analysis. In *Proceedings of the 4th International Workshop on Adversarial Information Retrieval on the Web* (Beijing, China, April 22, 2008). AIRWeb '08. ACM, New York, NY, 9-16.

[13] Liu Y., Gao B., Liu TY., Zhang Y., Ma Z., He S. and Li H. 2008. BrowserRank: letting web users vote for page importance. In *Proceedings of the 31th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (Singapore, July 20-24, 2008). SIGIR '08. ACM, New York, NY, 451-458.

[14] Martinez-Romo, J. and Araujo, L. 2009. Web spam identification through language model analysis. In *Proceedings of the 5th International Workshop on Adversarial Information Retrieval on the Web* (Madrid, Spain, April 21, 2009). AIRWeb '09. ACM, New York, NY, 21-28.

[15] Moshchuk, A., Bragin, T., Gribble, D.S. and Levy, M. H. 2006. A crawler-based study of spyware on the web. In *Proceedings of the thirteenth Annual Symposium on Network and Distributed System Security* (San Diego, California, US, February, 2006). NDSS '06.

[16] Nie, L., Wu, B. and Davison, D.B. 2007. Winnowing wheat from the chaff: Propagating trust to sift spam from the Web. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (Amsterdam, The Netherlands, July 23-27, 2007). SIGIR '07. ACM, New York, NY, 869-870.

[17] Ntoulas, A., Najork, M., Manasse, M. and Fetterly, D. 2006. Detecting Spam Web Pages through Content Analysis. *In Proceedings of the 15th International Conference on World Wide Web* (Edinburgh, Scotland, May 23-26, 2006). WWW '06. ACM, New York, NY, 83-92.

[18] Piskorski, J., Sydow, M. and Weiss, D. 2008. Exploring linguistic features for Web spam detection: A preliminary study. In *Proceedings of the 4th International Workshop on Adversarial Information Retrieval on the Web* (Beijing, China, April 22, 2008). AIRWeb '08. ACM, New York, NY, 25-28.

[19] Silverstein, C., Marais H., Henzinger M., and Moricz M. 1999. Analysis of a Very Large Web Search Engine Query Log. *Association for Computer Machinery, SIGIR Forum, 33, 3.*

[20] Singhal, A. *Challenges in running a commercial search engine*. 2005. Keynote presentation at SIGIR 2005, August 2005.

[21] Sobek, M. 2002. PR0 — Google's PageRank 0 penalty, http://pr.efactory.de/e-pr0.shtml, 2002.

[22] Urvoy, T., Chauveau, E., Filoche, P. and Lavergne, T. Tracking Web spam with HTML style similarities. *ACM Transactions on the Web*. 2, 1 (February, 2008).

[23] Urvoy, T., Lavergne, T. and Filoche, P. 2006. Tracking Web spam with hidden style similarity. In *Proceedings of the 2nd International Workshop on Adversarial Information Retrieval on the Web* (Seattle, Washington, August 10, 2006). AIRWeb '06. ACM, New York, NY, 25-32.

[24] Wu, B. and Davison, D.B. 2006. Detecting semantic cloaking on the Web. *In Proceedings of the 15th International Conference on World Wide Web* (Edinburgh, Scotland, May 23-26, 2006). WWW '06. ACM, New York, NY, 819-828.

[25] Wu, B. and Davison, D. B. 2005. Identifying link farm spam pages. In *Special interest tracks and posters of the 14th International Conference on World Wide Web* (Chiba, Japan, May 10-14, 2005). WWW '05. ACM, New York, NY, 820 – 829.

[26] Wu, B., Goel, V. and Davison, D.B. 2006. Propagating trust and distrust to demote Web spam. In *Workshop on Models of Trust for the Web* (Edinburgh, Scotland, May 22, 2006). MTW '06.

[27] Zhu, X. and Ghahramani, Z. 2002. *Learning from Labeled and Unlabeled Data with Label Propagation*. CMU CALD technical report CMU-CALD-02-107.

[28] http://www.yr-bcn.es/webspam/datasets/uk2006-info/