OSD: An Online Web Spam Detection System[*]

Bin Zhou
School of Computing Science
Simon Fraser University
Burnaby, BC, Canada
bzhou@cs.sfu.ca

Jian Pei
School of Computing Science
Simon Fraser University
Burnaby, BC, Canada
jpei@cs.sfu.ca

ABSTRACT

Web spam, which refers to any deliberate actions bringing to selected web pages an unjustifiable favorable relevance or importance, is one of the major obstacles for high quality information retrieval on the web. Most of the existing web spam detection methods are supervised that require a large and representative training set of web pages. Moreover, they often assume some global information such as a large web graph and snapshots of a large collection of web pages. However, in many situations such assumptions may not hold. Recently, we studied the problem of online web spam detection, and proposed the notion of spamicity to measure how likely a page is a spam web page [9, 7]. Spamicity is a more flexible and user-controllable measure than the traditional supervised classification methods. We developed efficient online link spam and term spam detection methods using spamicity. In this paper, we present a demonstration of *OSD*, an Online Spam Detection system which can efficiently calculate a spamicity score online for any page on the web.

1. INTRODUCTION

Ranking web pages is an essential task in web search and search engines. Due to huge business opportunities brought by high-ranked popular web pages, many tricks have been attempted to affect the rankings of search results in search engines. Conceptually, web spam refers to any deliberate actions that bring to selected web pages an unjustifiable favorable relevance or importance. Web spam seriously hurts the quality of information retrieval on the web. Combating web spam has become more and more important in web search.

Detecting web spam is a challenging web mining task. The first generation of web spam detection methods often adopt a supervised learning model. A training set consisting of pages labeled spam or normal is used to train a classifier, and the classifier is used to categorize other web pages. However, there is some subtlety about modeling web spam detection as a traditional classification problem. The critical difference between a popular web page and a spam page is whether the popularity is justifiable. However, the measurement of justifiability is often subtle and subjective. In other words, it is often hard to label a web page absolutely spam or non-spam. Obtaining a reliable training data set for effective web spam detection is difficult, if possible at all. Moreover, the scale of the web is huge, and keeps increasing with a fastened pace. It is hard to collect and maintain a training set sufficient for accurate spam detection. New tricks emerge from time to time. New training samples have to be captured in order to combat new tricks.

In addition, the applicability of spam detection methods is an important concern. Most of the existing methods assume explicitly or implicitly that the spam detection methods are run at search engine sites where the detectors have good knowledge about the web including the web graph and the snapshots of the web. However, this assumption may not always hold. Spam detection is often required in practice, too, for off-search engine applications, where spam detection has to be conducted on sites where both the knowledge about the web and the computational resources are limited. Particularly, such a spam detector does not have the global knowledge about the web such as the web graph and the snapshots of many web pages. Moreover, the computational resources are constrained so that constructing a large web graph or crawling many web pages are not feasible, partly due to the online response requirement.

Intelligent web browsers on personal computers are an example of such off-search engine applications. A web page currently shown in an intelligent web browser may have multiple links pointing to other web pages. It is desirable that such an intelligent browser can online detect whether those pages are spam or not. The detector can annotate the current web page and give users more information and more control in the course of browsing. In such a case, the spam detection has to be conducted on the browser site which may not have the information about the web graph and the snapshots of the web. Moreover, the spam detection has to be online since a user may not want to wait long for a detection result. To the best of our knowledge, all existing spam detection methods are based on large training data sets and/or global knowledge about the web. They may not

[*]This research is supported in part by an NSERC Discovery Grant and a research contract with Microsoft adCenter. All opinions, findings, conclusions and recommendations in this paper are those of the authors and do not necessarily reflect the views of the funding agencies.

KDD'09, June 28–July 1, 2009, Paris, France.

be applicable to off-search-engine applications.

Recently, we systematically studied the problem of online web spam detection and made the following progress [9, 7]. First, we proposed to use spamicity to measure how likely a web page is spam. Spamicity is a more flexible and user-controllable measure than the traditional classification methods. Second, we proposed efficient link spam and term spam detection methods based on spamicity. Our spamicity-based methods can return the spamicity score of a web page without any threshold. The score gives the degree of spamicity of the web page. This procedure does not need any training and is cost effective. Last, we evaluated our approach using a large real data set. The experimental results show that our method is effective and efficient.

Based on the above techniques, we developed *OSD* (for Online Spam Detection), a system which can efficiently calculate a spamicity score online for any page on the web. In Section 2, we describe the online web spam detection problem and discuss the critical techniques in our *OSD* system. In Section 3, we outline our demonstration plan.

2. SYSTEM OVERVIEW

In this section, we first describe two major categories of web spam tricks adopted by web spammers. Then, we present the techniques we developed for online web spam detection. We also propose a framework for online web spam detection, and analyze the computational challenges in real applications.

2.1 Web Spam

The main purpose of web spam tricks is to mislead search engines such that the web pages owned by those spammers can obtain a high ranking in search engine ranking results.

Most of the popular web search engines currently adopt some link-based ranking algorithms, such as PageRank [2]. Driven by the huge potential benefit of promoting rankings of pages, many attempts have been conducted to boost page rankings by making up some linkage structures, which are known as link spam. Gyöngyi *et al.* [4, 3] referred link spam to the cases where spammers set up structures of interconnected pages, called link spam farms, with the only purpose to boost the link-based ranking.

Term spam is the other type of web spam which disguises the content of a page so that it appears relevant to popular searches. Most of the term spam detection methods proposed so far adopted statistical analysis approaches.

All the existing methods assume explicitly or implicitly that the detector has some global knowledge about the web, such as the web graph and an extensive collection of web page snapshots so that statistics can be derived. Moreover, most of the existing methods require a training data set containing a good number of representative spam pages and/or normal pages. As analyzed in Section 1, those assumptions may not hold all the time.

2.2 Link Spamicity and Link Spam Detection

Web links can be modeled as a directed web graph $G = (V, E)$, where V is the set of web pages, and E is the set of hyperlinks. A link from page p to page q is denoted by an edge $p \rightarrow q$. An edge $p \rightarrow q$ can also be written as a tuple (p, q). A page p may have multiple hyperlinks pointing to page q, however, in the web graph, only one edge $p \rightarrow q$ is formed. Hereafter, by default our discussion is about a directed web graph $G = (V, E)$.

The link-based ranking methods such as PageRank are popularly used by major search engines in ranking web pages. PageRank measures the importance of a page p by considering how collectively other web pages point to p directly or indirectly. Formally, for a web page p, the PageRank score is defined as

$$PR(p, G) = d \sum_{p_i \in M(p)} \frac{PR(p_i, G)}{OutDeg(p_i)} + \frac{1-d}{N},$$

where $M(p)$ is the set of pages having a hyperlink pointing to p, $OutDeg(p_i)$ is the out-degree of p_i (i.e., the number of hyperlinks from p_i pointing to some pages other than p_i), d is a damping factor which models the random transitions on the web, and $N = |V|$ is the total number of pages in the web graph. The second additive term on the right hand side, $\frac{1-d}{N}$, is traditionally referred to as the random jump probability and corresponds to a minimal amount of PageRank score that every page gets by default.

Link spam is typically a local activity. In order to boost the ranking of one page, a small number of pages and links (comparing to the whole web) are created deliberately. Driven by this critical insight, in order to determine whether one target page is link spam or not, we can extract some important neighbor pages to capture the local link structures of the target page. We can calculate the utility-based link spamicity score, that is, how the PageRank score of the target page is boosted, to evaluate the likelihood that the target page is link spam.

Simply selecting those k-neighbor pages is not a precise way to capture the local link structures of the target page, since different neighbor pages may have different importance to the ranking score of the target page. To address the challenge, we proposed a page farm model [8].

Consider a web graph $G = (V, E)$. For a subset of vertices $H \subset V$, the induced subgraph on H with respect to PageRank calculation is $G(H) = (V, E')$, where $E' = E - \{p \rightarrow q | p \notin H\}$. A page farm model is used to model the surrounding link structures of a target page. For given parameters θ $(0 < \theta < 1)$ and $k > 0$, a (θ, k)-farm is a minimal set of pages whose distances to p are no more than k, and whose induced subgraph contributes to at least a θ portion of the PageRank score of a target page. The page farm of target page p as an induced subgraph is denoted by $Farm(p)$. Using real data sets, we showed when $\theta \geq 0.8$ and $k \geq 3$, the (θ, k) farms capture local link structures of web pages accurately [8, 7].

Using page farms, we can develop a utility-based link spam detection method. Intuitively, if p is link spam, then $Farm(p)$ should try to achieve the PageRank of p as high as possible. We can calculate the maximum PageRank score using the same number of pages and the same number of links as $Farm(p)$ has. We presented a systematic analysis on the optimal link structure [9, 7]. A page farm is called an optimal spam farm if it achieves the maximum PageRank score of the target page. We denote the PageRank score of an optimal spam farm having n pages and l links by $PR_{max}(n, l)$. We define the utility-based link spamicity of p whose page farm contains n pages and l links as $ULSpam(p) = \frac{PR(p)}{PR_{max}(n,l)}$.

The utility can be used as a measure on the likelihood that p is link spam. The utility-based link spamicity of a web page is between 0 and 1. The higher the utility-based link spamicity, the more the page farm is utilized to boost the PageRank of the target page. Spammers (i.e., builders of spam web pages) build up the link spam farms with the only purpose to boost the rankings of target pages as much as possible.

The optimal spam farms do not commonly happen on the web, because they are quite different from normal page farms. Moreover, since optimal spam farms are highly regular [9, 7], a search engine may easily detect the optimal spam farms. To disguise, a spammer may modify the optimal spam farm. However, the utility of the page farm of a spam page has to be high in order to obtain a high PageRank score using a relative small number of supporting pages. Using the utility-based link spamicity, we can still capture the disguised link spam.

The problem of utility-based link spam detection can be defined as follows. For a target page p and a utility threshold α, the problem is to determine whether the utility-based link spamicity of p is greater than or equal to α. If so, p is a suspect of link spam. Otherwise, p is reported as a normal page.

2.3 Term Spamicity and Term Spam Detection

The term-based ranking methods such as TFIDF [1] adopted by web search engines are victims of term spam. The previous studies (e.g., [6]) showed that the algorithms used by search engines to rank web pages based on their content information use various forms of the fundamental TFIDF metric. A web page p and a search query Q can be regarded as a set of keywords. The TFIDF score of a web page p with respect to a search query Q is defined as

$$TFIDF(p, Q) = \sum_{t \in P \cap Q} TF(t) \times IDF(t),$$

where $TF(t)$ is the term frequency of keyword t in p, and $IDF(t)$ is the inverse document frequency of keyword t which is the total number of documents in the collection divided by the number of documents that contain t.

A web page p can be divided into several parts, such as the page body, the page title, the meta tags in the HTML header, the URL address, and the anchor fields. The appearances of keywords in different fields carry different significance. When ranking web pages, in order to evaluate the content relevance, search engines may assign different weights to different fields. For example, the keywords in the text fields are used to determine the global relevance of the page with respect to a specific query.

Term spam refers to tricks that tailor the contents of text fields to make spam pages relevant for some queries. Spammers want to increase the TFIDF scores of term spam pages as much as possible. Since the IDF score for a specific keyword is often hard to be affected substantially by a spammer, the primary way to increase the TFIDF score is to increase the frequencies of keywords within some specific text fields of the term spam pages.

According to the TFIDF-based ranking methods, spammers may have two different strategies to deliberately influence the ranking results. First, a spammer can make a term spam page relevant to a large number of search queries. In other words, the term spam page receives a positive TFIDF score for a large set of queries. This can be done by including a large number of distinct keywords in the page. Second, a spammer can make a term spam page highly relevant to a specific search query. That is, the page receives a high TFIDF score for the given query. This can be done by repeating some targeted keywords in the page.

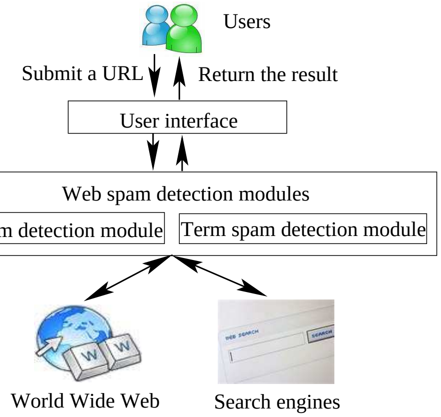
Users
Submit a URL
Return the result
User interface
Web spam detection modules
Link spam detection module
Term spam detection module
World Wide Web
Search engines

Figure 1: The general framework of *OSD*.

For a page, we can calculate a term spamicity score to measure the likelihood of the page being term spam. In order to determine whether a page p is term spam, we only need to parse page p and those pages having a link pointing to p (to collect the anchor fields). We propose the utility-based term spamicity. If page p is term spam, to be relevant to a search query Q, p should try to achieve the TFIDF score as high as possible. We can calculate the maximum TFIDF score for a query Q using the same number of keywords that p has. The utility of page p with respect to term spam is the ratio of the TFIDF score of p against the maximum TFIDF score of p that can be achieved. The utility can be used as a measure of the likelihood that p is term spam. Intuitively, if the utility is close to 1, the page is likely to be term spam.

We presented the results of the largest TFIDF score that a page p with n keywords of l occurrences can achieve [9]. We denote by $TFIDF_{max}(p)$ the maximum TFIDF score that a page p can achieve. For a target page p, the utility-based term spamicity of p is $UTSpam(p) = \frac{TFIDF(p,Q)}{TFIDF_{max}(p)}$, where Q is the set of keywords in p. The utility-based term spamicity of a web page p is between 0 and 1. The higher the utility-based term spamicity, the more page p is utilized to boost the TFIDF score.

The problem of online term spam detection can be defined as, given a utility-based term spamicity threshold β and a web page, determine whether the term spamicity of the page is greater than or equal to β. If so, then the page is a suspect of term spam. Otherwise, the page is not term spam.

2.4 A General Framework

Based on the above analysis, we build our *OSD* system by incorporating two spam detection modules, the link spam detection module and the term spam detection module. The general framework is shown in Figure 1.

The user communicates with the *OSD* system. Here users not only refers to web user individuals, but also refers to some other units that may need to access the spamicity information of some target web pages (e.g., intelligent web browsers). A user submits the URL of a target web page to the *OSD* system through a user interface. The two modules, the link spam detection module and the term spam detection module, collect related information of the target URL and then calculate the corresponding link spamicity score and term spamicity score, respectively. The calculation needs to access the web and the search query results returned by some search engines. After all, the *OSD* system combines the two spamicity scores and returns the final result to the user.

2.5 Critical Techniques

There are some critical technical challenges when we are developing the *OSD* system.

First of all, the link spam detection module needs to extract the (θ, k)-farm of a target page. A major operation to extract page farms is to calculate the PageRank score of a page in various induced subgraphs. Online calculating the PageRank score directly using the popular power method [5] is infeasible in practice since one round of PageRank calculation is time consuming. Moveover, extracting page farms involves finding those most important pages supporting the target page. How to efficiently identify those pages becomes a big challenge.

Second, to make our method extensively applicable, we assume that the whole web graph is unavailable. Consequently, we have to obtain the local link structure of a page by parsing the content of the page and querying web search engines. The major search engines including Google, Yahoo, and Microsoft Live Search can answer "link search" queries. A user can submit one URL to a search engine. The engine returns a list of pages having a hyperlink pointing to the submitted URL. Moreover, given a page p, we can parse the content of the page so as to know the pages that p points to. The major cost in link spam detection can be divided into two parts. The search engine querying load (in-link search) is the cost of sending a link search query to a search engine and obtaining the list of pages that have a hyperlink pointing to the query page. The web page out-link parsing load (out-link search) is the cost of retrieving a page and extracting the outgoing links in the page. The implementation needs to take into account the different types of cost.

Third, in term spam detection, there are three different major types of cost. The web page keyword parsing load (keyword search) is to extract the keywords from a web page by retrieving the page and parsing it. The search engine querying load (in-link search) is to extract the 1-neighbor pages having links pointed to a target page. This operation is to extract the anchor texts. This cost is the same as that in online link spam detection. If the IDF scores of keywords are not available, then we also need to derive them by querying search engines. Similarly to the case of link spam detection, the implementation needs to take into account the different types of cost.

For each challenge, we developed efficient and effective methods [9, 7].

3. DEMONSTRATION PLAN

The design and development of the *OSD* system involve a few challenging data mining issues. We will present our prototype system thoroughly in our demonstration. Particularly, we will focus on the following aspects.

First, *we will illustrate some examples of link spam pages and term spam pages in data analysis*. We will show the audience various examples of web spam pages, and outline the reasons that the existing methods cannot handle online web spam detection well. The audience will gain more understanding about the essence of online web spam detection, and raise awareness of online web spam detection in their research work.

Second, *we will present the technical details used in* OSD, including the efficient implementation. We will analyze the rationale behind the design, as well as the consideration regarding the scalability issues. This will show the audience how our system can efficiently and effectively online identify those web spam pages.

Third, *we will demonstrate a set of real case studies using our prototype system, and report the experimental results conducted on a large real data set*. We will share with the audience our interesting findings in the real cases.

Finally, *we will showcase our prototype system*, including a link spam detection module and a term spam detection module. Our system works as a plug-in tool to web browsers. Each page to be viewed in the browser will be assigned a spamicity score. The related statistics results will also be showed. The audience is encouraged to test our prototype system on various web pages to further understand online web spam detection and experience the *OSD* prototype system.

4. REFERENCES

[1] R. A. Baeza-Yates and B. A. Ribeiro-Neto. *Modern Information Retrieval*. ACM Press/Addison-Wesley, 1999.

[2] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In *Proceedings of the seventh international conference on World Wide Web (WWW'98)*, pages 107–117, Amsterdam, The Netherlands, The Netherlands, 1998. Elsevier Science Publishers B. V.

[3] Z. Gyöngyi and H. Garcia-Molina. Link spam alliances. In *Proceedings of the 31st International Conference on Very Large Databases (VLDB'05)*, pages 517–528. ACM, 2005.

[4] Z. Gyöngyi and H. Garcia-Molina. Web spam taxonomy. In *Proceedings of the 1st International Workshop on Adversarial Information Retrieval on the Web (AIRWeb'05)*, 2005.

[5] A. Langville and C. Meyer. Deeper inside pagerank. *Internet Mathematics*, 1(3):335–380, 2004.

[6] A. Ntoulas, M. Najork, M. Manasse, and D. Fetterly. Detecting spam web pages through content analysis. In *Proceedings of the 15th International World Wide Web Conference (WWW'06)*, pages 83–92, New York, NY, USA, 2006. ACM Press.

[7] B. Zhou and J. Pei. Link spam detection using page farms. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, accepted.

[8] B. Zhou and J. Pei. Sketching landscapes of page farms. In *Proceedings of the 2007 SIAM International Conference on Data Mining (SDM'07)*, Minneapolis, MN, USA, 2007. SIAM.

[9] B. Zhou, J. Pei, and Z. Tang. A spamicity approach to web spam detection. In *Proceedings of the 2008 SIAM International Conference on Data Mining (SDM'08)*, pages 277–288, Atlanta, GA, USA, 2008. SIAM.