Content Trust Model for Detecting Web Spam

Wei Wang, Guosun Zeng

Department of Computer Science and Technology, Tongji University, Shanghai 201804, China
Tongji Branch, National Engineering & Technology Center of High Performance Computer, Shanghai 201804, China

willtongji@gmail.com

Abstract. As it gets easier to add information to the web via html pages, wikis, blogs, and other documents, it gets tougher to distinguish accurate or trustworthy information from inaccurate or untrustworthy information. Moreover, apart from inaccurate or untrustworthy information, we also need to anticipate web spam – where spammers publish false facts and scams to deliberately mislead users. Creating an effective spam detection method is a challenge. In this paper, we use the notion of content trust for spam detection, and regard it as a ranking problem. Evidence is utilized to define the feature of spam web pages, and machine learning techniques are employed to combine the evidence to create a highly efficient and reasonably-accurate spam detection algorithm. Experiments on real web data are carried out, which show the proposed method performs very well in practice.

Key words: web spam; content trust; ranking; SVM; machine learning

1 Introduction

Information retrieval (IR) is the study of helping users to find information that matches their information needs. Technically, information retrieval studies the acquisition, organization, storage, retrieval, and distribution of information [1]. However, as it gets easier to add information to the web via html pages, wikis, blogs, and other documents, it gets tougher to distinguish accurate or trustworthy information from inaccurate or untrustworthy information. A search engine query usually results in several hits that are outdated and/or from unreliable sources and the user is forced to go through the results and pick what he/she trust requirements.

Please use the following format when citing this chapter:

Wang, W. and Zeng, G., 2007, in IFIP International Federation for Information Processing, Volume 238, Trust Management, eds. Etalle, S., Marsh, S., (Boston: Springer), pp. 139–152.

Moreover, apart from inaccurate or untrustworthy information, we also need to anticipate web spam – where spammers publish false facts and scams to deliberately mislead users. Creating an effective spam detection method is a challenge.

In the context of search engines, a spam web page is a page that is used for spamming or receives a substantial amount of its score from other spam pages. Spam can be great harmful for several reasons. First, spamming is annoying for users because it makes it harder to find truly and trustworthy information and leads to frustrating search experiences. Second, if a user searches for information that is relevant to your pages but your pages are ranked low by search engines, then the user may not see the pages because one seldom clicks a large number of returned pages. Finally, a search engine may waste significant resources on spam pages because spam pages consume crawling bandwidth, pollute the web, and distort search ranking [2].

In this paper, we explore a novel content trust model based on evidence for detecting spam. The notion of content trust was first introduced by Gil et al. to solve the problem of reliability of the web resource [3]. But they only proposed the preliminary notion of content trust, and did not take the information content into account actually. In our opinion, spam web pages are a salient kind of distrusted web resource which can utilize content trust to model it. So, we developed a content trust model with ranking algorithms to detect web spam. Experiments show that our method performs very well in finding spam web pages.

The main contributions of this paper are follows:

- A novel content trust model is proposed for web spam detection
- A ranking algorithm is adapted to the model for spam detection
- Experiments of real web data are carried out to evaluate the proposed method

The rest of this paper is organized as follows. We introduce some background and review some related work in Section 2. Section 3 introduces the proposed content trust model for detecting web spam. We first describe the key evidence for the model, and then a rank learning algorithm is proposed to detect web spam. We evaluate our approach and analyze the experiments results in Section 4. Section 5 concludes the paper.

2 Background and Related Work

2.1 Web Spam and Ranking Problem

Web search has become very important in the information age. Increased exposure of pages on the Web can result in significant financial gains and/or fames for organizations and individuals. Unfortunately, this also results in spamming, which refers to human activities that deliberately mislead search engines to rank some pages higher than they deserve. The following description of web spam taxonomy is based on [1], [2], [5] and [12].

Content-based spamming methods basically tailor the contents of the text fields in HTML pages to make spam pages more relevant to some queries. This kind of spamming can also be called *term spamming*, and there are two main term spam techniques: repeating some important terms and dumping of many unrelated terms [1].

Link spam is the practice of adding extraneous and misleading links to web pages, or adding extraneous pages just to contain links. An early paper investigating link spam is Davison [6], which considered nepotistic links. Baeza-Yates et al. [7] present a study of collusion topologies designed tot boost PageRank [8] while Adali et al. [9] show that generating pages with links targeting a single page is the most effective means of link spam. Gyongyi et al. [10] introduce TrustRank which finds non-spam pages by following links from an initial seed set of trusted pages. In [4] Fetterly et al. showed ways of identifying link spam based on divergence of sites from power laws. Finally, Mishne et al. [11] present a probabilistic method operating on word frequencies, which identifies the special case of link spam within blog comments.

Hiding techniques is also used by spammers who want to conceal or to hide the spamming sentences, terms and links so that Web users do not see them [1]. *Content hiding* is used to make spam items invisible. One simple method is to make the spam terms the same color as the background color. In *cloaking*, Spam Web servers return a HTML document to the user and a different document to a Web crawler. In this way, the spammer can present the Web user with the intended content and send a spam pages to the search engine for indexing.

There are pages on the Web that do not try to deceive search engines at all and provide useful and reliably contents to Web users; there are pages on the Web that include many artificial aspects that can only be interpreted as attempts to deceive search engines, while not providing useful information at all and of course can be regarded as distrusted information; finally, there are pages that do not clearly belong to any of these two categories [12]. So, in our opinion, web spam detection can not be simply considered as a problem of classification which most of the traditional work do [2, 4]. In fact, it can be regarded as a ranking problem which arises recently in the social science and in information retrieval where human preferences play a major role [13, 14]. The detail of ranking problem will be introduced in section 3.2.

2.2 Trust, Content Trust and Spam Detection

On the other hand, trust is an integral component in many kinds of human interaction, allowing people to act under uncertainty and with the risk of negative consequences. Human users, software agents, and increasingly, the machines that provide services all need to be trusted in various applications or situations. Trust can be used to protect data, to find accurate information, to get the best quality service, and even to bootstrap other trust evaluations [3]. In order to evaluate the reliability of the web resource, content trust was proposed as a promising way to solve the problem. So, it is promising to use content trust to model the reliability of the information, and solve the problem of web spam detection. Content trust was first

introduced by Gil et al. on the International World Wide Web Conference in 2006. They discussed content trust as an aggregate of other trust measure, such as reputation, in the context of Semantic Web, and introduced several factors that users consider in detecting whether to trust the content provided by a web resource. The authors also described a simulation environment to study the models of content trust. In fact, the real value of their work is to provide a starting point for further exploration of how to acquire and use content trust on the web.

Trust has been utilized as a promising mechanism to solve the problem of spam detection, and this kind of work including [10], [15], [16], [17] and [18]. TrustRank [10] proposed by Gyongyi et al. maybe the first mechanisms to calculate a measure of trust for Web pages. It is based on the idea that good sites seldom point to spam sites and people trust these good sites, and in their more recent paper [16], the concept of "spam mass" is introduced to estimate a page's likehood to be spam. B. Wu et al. [15] expand on this approach to form a better performing Topic TrustRank. It combines topical information with the notion of trust on the Web based on link analysis techniques. Metaxas et al. [17] also describe an effective method to detect link spam using trust, which propagate from a seed set of spam pages along incoming links. Further more, L. Nie et al. [18] describe and compare various trust propagation methods to estimate the trustworthiness of each Web pages. They propose how to incorporate a given trust estimate into the process of calculating authority for a cautious surfer.

In fact, before trust was introduced into the effort of fighting web spam, it has been used in other system, such as reputation systems and peer-to-peer systems. Kamvar et al. [19] proposed a trust-based method to determine reputation in peer-to-peer systems. Guha et al. [20] study how to propagate trust scores among a connected network of people. Moreover, varieties of trust metrics have been studied, as well as algorithms for transmission of trust across individual webs of trust, including ours previous research [21, 22].

Compared to the research summarized above, we utilize trust mechanism based on actual content of the web pages, and explore a set of evidence to denote the content trust of the web pages, and propose a novel content trust model with ranking algorithm for detecting spam.

3 Content Trust Model for Spam Detection

In human society, evidence for trust plays a critical role in people's everyday life, and historians, juries and others rely on evidence to make judgments about the past and trust what will happen in the future. In a legal setting, evidence is defined as follows:

(Oxford English Dictionary) "*Evidence is information, whether in the form of personal testimony, the language of documents, or the production of material objects that is given in a legal investigation, to establish the fact or point in question*"

In light of this, based on previous research, we explore a set of salient evidences which can help to tell a web page is a spam or not, and most of them based on the content of web pages. Moreover, some of these evidences are independent of the language a page is written in, others use language-dependent statistical properties.

The overview of the proposed content trust model can be descried in Figure 1.

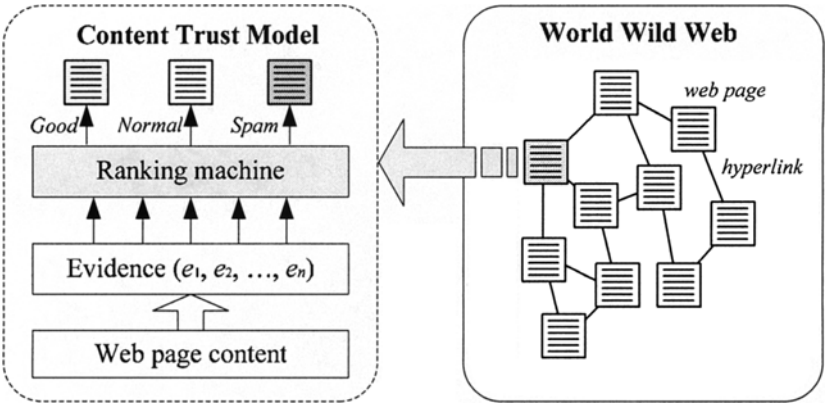

Figure 1 Overview of the content trust model

We first analysis the content of the web page, and extract some salient evidence which can be used to evaluate the reliability of the content. Then, we train a ranking machine using the evidence as the feature to predict the trustworthy of the future web pages. It is obvious that the evidence extraction and the rank machine training is the key. We descript them in more detail in the following section.

3.1 Evidence for detecting web spam

There are many salient factors that affect how users determine trust in content provided by Web pages. So we extract the following evidence for detecting web spam based on previous research [2, 4, 5].

One popular practice when creating spam pages is "keyword stuffing". During keyword stuffing, the content of a web page is stuffed with a number of popular words. So, the first evidence can be number of words in the page. Evidence of an excessive number of words in the title of a page is a better indicator of spam than the number of words in the full page, which can be defined as the second evidence. The third evidence takes keyword stuffing one step further, concatenating a small number (2 to 4) of words to form longer composite words.

Another common practice among search engines is to consider the anchor text of a link in a page as annotation describing the content of the target page of that link. Evidence of higher fractions of anchor text may imply higher prevalence of spam, which can be defined as the fourth evidence. Some search engines use information from certain HTML elements in the pages that are not rendered by browsers. We

define the fifth evidence of fraction of visible content. Some spam pages replicate their content several times in an attempt to rank higher. To locating redundant content within a page, we measure the redundancy of web pages by the compression ratio, which defined as the sixth evidence.

The seventh evidence is to examine where the keywords in spam pages come from. We first identified the 100 most frequent words in our corpus, and then computed, for each page, the fraction of words contained in that page found among the 100 most common words. For the eighth evidence, we examined the prevalence of spam in pages, based on the fraction of stop-words that they contain. To account for this potential pitfall, and we also measure the fraction of the 100 most popular words contained within a particular page.

The ninth and tenth evidence in this paper are Independent *n*-gram likelihoods and Conditional *n*-gram likelihoods, which can be used to analyze the content of the page for grammatical and ultimately semantic correctness. More details can be found in reference [2].

Except the evidence discussed above, we also use the following additional evidence to detect web spam.

- Various features of the host component of a URL
- IP addresses referred to by an excessive number of symbolic host names
- The rate of evolution of web pages on a given site
- Excessive replication of content

Table 1 describes the major evidence used in this paper.

Table 1 Evidence for spam detection

	Name	How to calculate
1	Number of words in the page	the number of words in the page
2	Number of words in the page title	the number of words in title
3	Average length of words	$\frac{\sum \text{the length (in characters) of each non-markup words}}{\text{the number of the words}}$
4	Amount of anchor text	$\frac{\text{all words (excluding markup) contained in anchor text}}{\text{all words (excluding markup) contained in the page}}$
5	Fraction of visible content	$\frac{\text{the aggregate length of all non-markup words on a page}}{\text{the total size of the page}}$
6	Compressibility	$\frac{\text{the size of the compressed page}}{\text{the size of the uncompressed page}}$
7	Fraction of page drawn from globally popular words	$\frac{\sum \text{the number of each words among the N most common words}}{\text{the number of all the words}}$

8	Fraction of globally popular words	$\frac{\text{the number of the words among the N most common words}}{\text{N}}$
11	Various features of the host component of a URL	
12	IP addresses referred to by an excessive number of symbolic host names	
13	Outliers in the distribution of in-degrees and out-degrees of the graph induced by web pages and the hyperlinks between them	
14	The rate of evolution of web pages on a given site	
15	Excessive replication of content	
...	...	

3.2 Ranking machine for spam detection

As we have discussed above. One way of combining our evidence methods is to view the spam detection as a ranking problem. In this case, we want to create a ranking model which, given a web page, will use the page's features jointly in order to correctly rank it in one of several ordered classes, such as good, normal and spam. We follow a standard machine learning process to build out ranking model. In general, constructing a ranking machine involves a training phase during which the parameters of the classifier are determined, and a testing phase during which the performance of the ranking machine is evaluated. The whole process can be described in Figure 2.

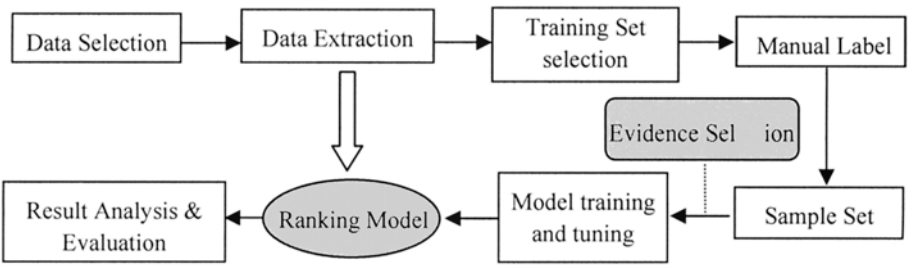

Figure 2 Process of factoid/definition mining from content

The most important process in Figure 2 is evidence selection which forms the features of the proposed ranking model. Besides evidence described above, we also use some normal text features. The total number of the feature is 24 in our implementation of the model. For every web page in the data set, we calculated the value for each of the features, and we subsequently used these values along with the class label for the training of our ranking machine.

In ranking problem, a number of candidates are given and a total order is assumed to exist over the categories. Labeled instances are provided. Each instance is represented by a feature vector, and each label denotes a rank. Ranking SVM [14] is a method which formalizes learning to rank as learning for classification on pairs

of instances and tackles the classification issue by using SVM. The reason why we use ranking SVM is because it performs best compare to the other method, such as Naïve Bayesian [24] and decision tree [23] for ranking problem. The experiments result is described in section 4 lately. Here, we only introduce our method of adapting ranking SVM to the problem of spam detection.

In formally, assume that there exists an input space $X \in R^n$, where n denotes number of features. There exists an output space of ranks (categories) represented by labels $Y = \{r_1, r_2, \cdots, r_q\}$ where q denotes number of ranks. Further assume that there exists a total order between the ranks $r_q \succ r_{q-1} \succ \cdots \succ r_1$, where $\succ$ denotes a preference relationship. A set of ranking functions $f \square F$ exists and each of them can determine the preference relations between instances:

$$\vec{x}_i \succ \vec{x}_j \Leftrightarrow f(\vec{x}_i) \succ f(\vec{x}_j) \tag{1}$$

Suppose that we are given a set of ranked instances $S = \{(\vec{x}_i, y_i)\}_{i=1}^{l}$ from the space $X \times Y$. The task here is to select the best function f' from F that minimizes a given loss function with respect to the given ranked instances.

Herbrich et al. [14] propose formalizing the rank learning problem as that of learning for classification on pairs of instances in the field of information retrieval. We can adapt this method to the spam detection problem in a similar way. First, we assume that f is a linear function.

$$f_{\vec{w}}(\vec{w}, \vec{x}) \tag{2}$$

where $\vec{w}$ denotes a vector of weights and $\langle \cdot, \cdot \rangle$ stands for an inner product. Plugging (2) into (1) we obtain

$$\vec{x}_i \succ \vec{x}_j \Leftrightarrow \left\langle \vec{w}, \vec{x}_i - \vec{x}_j \right\rangle > 0 \tag{3}$$

The relation $\vec{x}_i \succ \vec{x}_j$ between instance pairs $\vec{x}_i$ and $\vec{x}_j$ is expressed by a new vector $\vec{x}_i - \vec{x}_j$. Next, we take any instance pair and their relation to create a new vector and a new label. Let $\vec{x}^{(1)}$ and $\vec{x}^{(2)}$ denote the first and second instances, and let $y^{(1)}$ and $y^{(2)}$ denote their ranks, then we have

$$\left(\vec{x}^{(1)} - \vec{x}^{(2)}, z\right), z = \begin{cases} +1, y^{(1)} \succ y^{(2)} \\ -1, y^{(1)} \prec y^{(2)} \end{cases} \tag{4}$$

From the given training data set S, we create a new training data set S' containing m labeled vectors.

$$S' = \{\vec{x}_i^{(1)} - \vec{x}_i^{(2)}, z_i\}_{i=1}^{m} \tag{5}$$

Next, we take S' as classification data and construct a SVM model that can assign either positive label $z = +1$ or negative label $z = -1$ to any vector $\vec{x}^{(1)} - \vec{x}^{(2)}$.

Constructing the SVM model is equivalent to solving the following Quadratic Optimization problem [14]:

$$\min_{\vec{w}} \sum_{i=1}^{m} \left[1 - z_i \left\langle \vec{w}, \vec{x}_i^{(1)} - \vec{x}_i^{(2)} \right\rangle \right] + \lambda \left\| \vec{w} \right\|^2 \tag{6}$$

The first term is the so-called empirical Hinge Loss and the second term is regularizer.

Suppose that $\vec{w}^*$ is the weights in the SVM solution. Geometrically $\vec{w}^*$ forms a vector orthogonal to the hyperplane of Ranking SVM. We utilize $\vec{w}^*$ to form a ranking function $f_{\vec{w}^*}$ for ranking instances.

$$f_{\vec{w}^*}(\vec{x}) = \left\langle \vec{w}^*, \vec{x} \right\rangle \tag{7}$$

When Ranking SVM is applied to spam detection, an instance is created from the evidence we proposed in Section 3.1. Each feature is defined as a function of the document content.

4 Simulation Results and Performance Evaluation

4.1 Data configuration

The data set in the following experiments is collected through Google search engine follow the whole process showed in Figure 3. The process of assembling this collection consists of the following two phases: web crawling and then labeling, which are described in the rest of this section.

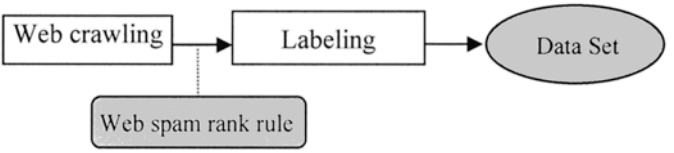

Figure 3 Process of web spam data collection

We follow the whole spam data collection process proposed in [12]. The crawl was done using the *TrustCrawler* which developed for this research. The crawler was limited to the .cn and .com domain and to 8 levels of depth, with no more than 5,000 pages per host. The obtained collection includes 500,000 million pages, and includes pages from 1000 hosts. The collection was stored in the WARC/0.9 format which is a data format in which each page occupies a record, which includes a plain text header with the page URL, length and other meta-information, and a body with the verbatim response from the Web servers, including the HTTP header. A total of ten volunteer students were involved in the task of spam labeling. The volunteers were provided with the rules of spam web pages described in reference [12], and they were asked to rank a minimum of 200 hosts. Further, we divide out data set in two

groups according to the language used in the page. The first data set is composed with English web pages (DS1), and the other is Chinese web pages (DS2).

In order to train our rank machine, we used the pages in the manually ranked data set to serve as our training data set. For our feature set, we used all the metrics described in Section 3. But for Chinese data set, some of the evidence is not suitable, such as "average length of words", and we ignore such features. Here, without loss of generality, every page labeled with three kind of rank: good, normal, and spam. For every web page in the data set, we calculated the value for each of the features, and we subsequently used these values along with the class label for the training of our ranking model.

4.2 Ranking techniques comparison

We experimented with a variety of ranking techniques, and here we only present the following algorithms: decision-tree based ranking techniques (R-DT) [23], Naïve Bayesian based ranker (R-NB) [24] and ranking support vector machine (R-SVM), which modified by us in section 3.2 to suit the problem of spam detection. All algorithms are implemented within the Weka framework [25].

The metric we used to compare the different algorithm here is the ROC (Receiver Operating Characteristics) curve [26], and AUC. An ROC curve is useful for comparing the relative performance among different classifiers, and the area under the ROC (AUC) provides a approach for evaluation which model is better on average. If a ranking is desired and only a dataset with class labels is given, the area under AUC can be used to evaluate the quality of rankings generated by an algorithm. AUC is a good "summary" for comparing two classifiers across the entire range of class distributions and error costs. AUC is actually a measure of the quality of ranking. The AUC of a ranking is 1 (the maximum AUC value) if no positive example precedes any negative example.

Using the metric of AUC, we found that R-SVM based techniques performed best both on DS1 and DS2, but that the other techniques were not far behind. The result is showed in Figure 4. The experiments in the rest of the paper are all carried out with R-SVM.

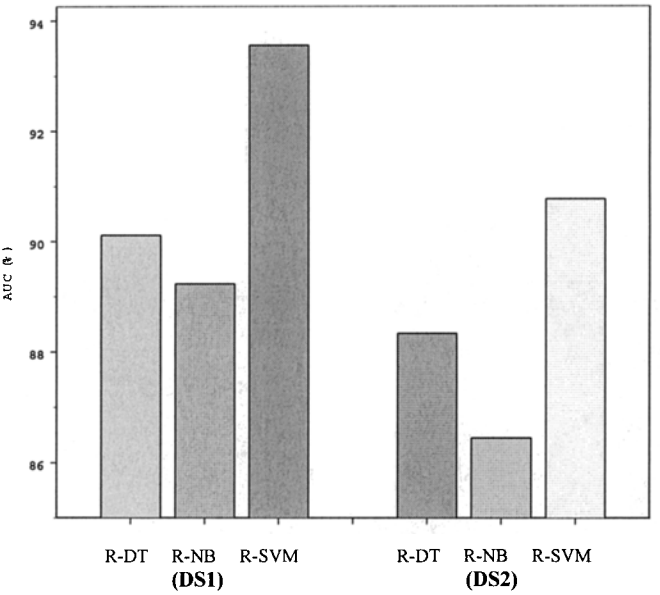

Figure 4 Comparison of varies ranking algorithms on AUC

4.3 Performance of ranking SVM for spam detection

Using all of the aforementioned features, the ranking accuracy after the ten-fold cross validation process is encouraging: 90.13% of our judged pages were ranked correctly, while 9.87% were ranked incorrectly. We can summarize the performance of our ranking machine using a precision- recall matrix (Table 2). More detail about how to calculate recall and precision can be found in reference [1].

The precision-recall matrix shows the recall (the true-positive and true-negative rates), as well as the precision:

Table 2 Recall and precision of our ranking machine

	DS1		DS2	
Rank	Recall (%)	Precision (%)	Recall (%)	Precision (%)
Good	81.34	83.77	83.36	85.95
Normal	95.15	93.84	96.89	91.40
Spam	87.79	88.12	86.04	86.82

Here, the evaluation measure is based on rankings of each web page, which is different from recall and precision measures in traditional classification.

We have also experimented with various techniques for improving the accuracy of our ranking method. Here, we will report on the most popular ones: boosting [13]. This technique essentially creates a set of models, which are then combined to form a

composite model. In most cases, the composite model performs better than any individual one (Table 3). More detail of this method can be found in reference [13]

After applying boosting to the ranking machine described above we obtain the following precision/recall values, which improve the accuracy of the method on all the terms.

Table 3 Recall and precision after boosting

	DS1		DS2	
Rank	Recall (%)	Precision (%)	Recall (%)	Precision (%)
Good	84.78	85.95	84.65	86.07
Normal	96.37	95.67	97.67	92.83
Spam	89.96	90.60	86.98	87.05

5 Conclusions

In this paper, we explore a novel content trust model for spam detection algorithm based on evidence of the pages. This method takes the web spam detection task as a ranking problem. And we present how to employ machine learning techniques that combine our evidence to create a highly efficient and reasonably-accurate spam detection algorithm. Experiments show that our method performs very well on the crawled data set. Some of the evidence for spam in this paper may be easily fooled by spammers, so we plan to use more natural language techniques to recognize artificially generated text in our future work, and more accurate machine learning method is also promising to be carried out on real world large-scale datasets.

Acknowledgements

This research was partially supported by the National Natural Science Foundation of China under grant of 60673157, the Ministry of Education key project under grant of 105071 and SEC E-Institute: Shanghai High Institutions Grid under grant of 200301.

References

1. B. Liu, Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data, Springer-Verlag Berlin Heidelberg, (2007)
2. A. Ntoulas, M. Najork, M. Manasse, et al., Detecting Spam Web Pages through Content Analysis. In Proceedings of the 15th International World Wide Web Conference (WWW'06), May 23–26, Edinburgh, Scotland, (2006)

3. Y. Gil, D. Artz, Towards Content Trust of Web Resources. In Proceedings of the 15th International World Wide Web Conference (WWW'06), May 23–26, Edinburgh, Scotland, (2006)
4. D. Fetterly, M. Manasse, M. Najork, Spam, Damn Spam, and Statistics: Using Statistical Analysis to Locate Spam Web Pages. In 7th International Workshop on the Web and Databases, (2004)
5. Z. Gyongyi, H. Garcia-Molina, Web Spam Taxonomy. In 1st International Workshop on Adversarial Information Retrieval on the Web, May (2005)
6. B. Davison, Recognizing Nepotistic Links on the Web. In AAAI-2000 Workshop on Artificial Intelligence for Web Search, July (2000)
7. R. Baeza-Yates, C. Castillo, V. Liopez, PageRank Increase under Different Collusion Topologies. In 1st International Workshop on Adversarial Information Retrieval on the Web, May (2005)
8. L. Page, S. Brin, et al., The PageRank Citation Ranking: Bringing Order to the Web. Stanford Digital Library Technologies Project, (1998)
9. S. Adali, T. Liu, M. Magdon-Ismail, Optimal Link Bombs are Uncoordinated. In 1st International Workshop on Adversarial Information Retrieval on the Web (AIRWeb'05), May (2005)
10. Z. Gyongyi, H. Garcia-Molina, J. Pedersen, Combating Web Spam with TrustRank. In 30th International Conference on Very Large Data Bases, Aug. (2004)
11. G. Mishne, D. Carmel, R. Lempel, Blocking Blog Spam with Language Model Disagreement. In 1st International Workshop on Adversarial Information Retrieval on the Web, May (2005)
12. C. Castillo, D. Donato, L. Becchett, et al., A Reference Collection for Web Spam. SIGIR Forum, 40(2), 11-24 (2006)
13. Y. B Cao, J. Xu, T. Y .Liu et al., Adapting Ranking SVM to Document Retrieval, In Proceedings of the 29th Annual International ACM SIGIR Conference On Research and Development in Information Retrieval, 186-193 (2006)
14. R. Herbrich, T. Graepel, K. Obermayer, Large Margin Rank Boundaries for Ordinal Regression. Advances in Large Margin Classifiers, 115-132 (2000)
15. B. Wu, V. Goel, B. D. Davison, Topical TrustRank: Using Topicality to Combat Web Apam. In Proceedings of the 15th International World Wide Web Conference (WWW'06), May 23–26, Edinburgh, Scotland, (2006)
16. Z. Gyiongyi, P. Berkhin, H. Garcia-Molina, et al, Link Spam Detection Based on Mass Estimation, In Proceedings of the 32nd International Conference on Very Large Databases (VLDB'06), (2006)
17. P. T. Metaxas, J. DeStefano, Web Spam, Propaganda and Trust, In 1st International Workshop on Adversarial Information Retrieval on the Web (AIRWeb'05), May (2005)
18. L. Nie, B. Wu and B. D. Davison. Incorporating Trust into Web Search. Technical Report LU-CSE-07-002, Dept. of Computer Science and Engineering, Lehigh University, (2007)
19. S. D. Kamvar, M. T. Schlosser, H. Garcia-Molina, The Eigentrust Algorithm for Reputation Management in P2P Networks. In Proceedings of the 12th International World Wide Web Conference (WWW'03), Budapest, Hungary,May (2003)

20. R. Guha, R. Kumar, P. Raghavan, and A. Tomkins. Propagation of Trust and Distrust. In Proceedings of the 13th International World Wide Web Conference (WWW'04), New York City, May (2004)
21. W. Wang, G. S. Zeng, L. L. Yuan, A Semantic Reputation Mechanism in P2P Semantic Web, In Proceedings of the 1st Asian Semantic Web Conference (ASWC), LNCS 4185, 682-688 (2006)
22. W. Wang, G. S. Zeng, Trusted Dynamic Level Scheduling Based on Bayes Trust Model. Science in China: Series F Information Sciences, 37(2), 285-296 (2007)
23. F. J. Provost, P. Domingos, Tree Induction for Probability-Based Ranking. Machine Learning, 52(3), 199-215 (2003)
24. H. Zhang, J. Su, Naive Bayesian Classifiers for Ranking, Proceedings of the 15th European Conference on Machine Learning (ECML'04), Springer (2004)
25. I. H. Witten, E. Frank, Data Mining - Practical Machine Learning Tools and Techniques with Java Implementation. Morgan Kaufmann, (2000)
26. F. Provost, T. Fawcett, Analysis and Visualization of Classifier Performance: Comparison under Imprecise Class and Cost Distribution. In Proceedings of the Third International Conference on Knowledge Discovery and Data Mining, AAAI Press, 43-48 (1997)
27. Y. Freund, R. E. Schapire, A Decision-theoretic Generalization of On-line Learning and an Application to Boosting. In European Conference on Computational Learning Theory, (1995)