Web Spam Detection by Learning from Small Labeled Samples

Jaber Karimpour
Department of Computer Science, University of Tabriz, Tabriz, Iran

Ali A. Noroozi
Department of Computer Science, University of Tabriz, Tabriz, Iran

Somayeh Alizadeh
Department of Industrial Engineering, K. N. Toosi University of Technology, Tehran, Iran

ABSTRACT
Web spamming tries to deceive search engines to rank some pages higher than they deserve. Many methods have been proposed to combat web spamming and to detect spam pages. One basic method is using classification, i.e., learning a classification model from previously labeled training data and using this model for classifying web pages to spam or non-spam. A drawback of this method is that manually labeling a large number of web pages to generate the training data can be biased, non-accurate, labor intensive and time consuming. In this paper, we are going to propose a new method to resolve this drawback by using semi-supervised learning to automatically label the training data. To do this, we incorporate Expectation-Maximization algorithm that is an efficient and an important algorithm of semi-supervised learning. Experiments are carried out on the real web spam data, which show the new method, performs very well in practice.

General Terms
Information Retrieval, Search Engine, Machine Learning.

Keywords
Adversarial Information Retrieval, Web Search, Web Spam Detection, Semi-supervised Learning, Expectation Maximization Algorithm.

1. INTRODUCTION
With the explosive growth of information on the web, the web has become the most successful and giant distributed computing application today. Billions of web pages are shared by millions of organizations, universities, researchers, etc. Web search provides great functionality for distributing, sharing, organizing, and retrieving the growing amount of information [1, 2, 3]. As a result, search engines have become more and more important and are used by millions of people to find necessary information. It has become very important for a web page, to be ranked high in the important search engines' results. As a result many techniques are proposed to influence ranking and improve the rank of a page. Some of these techniques are legal and are called Search Engine Optimization (SEO) techniques, but some are not legal or ethical and try to deceive ranking algorithms. They try to rank pages higher than they deserve [4].

Web spam refers to web content that get high rank in search engine results despite low information value. Spamming not only misleads users, but also imposes time and space cost to search engine crawlers and indexers. That is why crawlers try to detect web spam pages to avoid processing and indexing them.

Various methods have been proposed to combat web spamming and to detect spam pages. One important and basic method is considering web spam detection as a binary classification problem [5]. It tries to build a classifier from previously labeled web pages. This classifier is later used to classify web pages to spam or non-spam (good pages).

Labeling is done manually by some experts of spamming techniques. It is a difficult task because it is hard to find some experts familiar with spamming techniques and capable of distinguishing spam and non-spam pages. Even if you can find some, there is a possibility of biased labeling because there is no exact rule dedicating spam pages [6]. It also takes a long time to review a large set of web data and label them.

We claim the labeling problem can be resolved by using semi-supervised learning for the labeling phase. We propose a new method that uses the Expectation-Maximization (EM) algorithm to learn a classification model from a small set of labeled examples and a large set of unlabeled examples. This classification model labels the training data with spam or non-spam. These data can be used by any supervised learning algorithm like naïve Bayesian to classify web pages to spam or non-spam.

The rest of the paper is organized as follows. Section 2 provides a background on web spam along with its detection methods. Section 3 describes semi-supervised learning and the proposed method to resolve the labeling problem. Experimental results are discussed in Section 4, and finally, Section 5 concludes the paper.

2. RELATED WORK
In this section, we describe web spam and spamming techniques. Then, various web spam detection methods are discussed. TrustRank and binary classification are two important methods discussed here.

2.1 Web Spam
Web search has become very important in the information age. People frequently use search engines to find a company or product, so getting a high ranking position in a search engine's result becomes crucial for businesses. By studying ranking algorithms of various search engines, a lot of techniques have been proposed for a web page to be ranked high in a search engine's results. Unfortunately, these techniques result in web spamming which refers to

Actions intended to mislead search engines into ranking some pages higher than they deserve [4].

In fact, there is injustice in these activities.

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 is also called term spamming. There are two main content spamming techniques, which simply create synthetic contents containing spam terms: repeating some important terms and dumping many unrelated terms [7]. Ntoulas et al. [5] elaborate these techniques.

Link spamming misuses link structure of the web to spam pages. There are two main kinds of link spamming. Out-link spamming tries to boost the hub score of a page by adding out-links in it pointing to some authoritative pages. One of the

common techniques of this kind of spamming is directory cloning, i.e., replicating a large portion of a directory like Yahoo! in the spam page. In-link spamming refers to persuading other pages, especially authoritative ones, to point to the spam page. In order to do this, a spammer might adopt these strategies: creating a honey pot, infiltrating a web directory, posting links on user-generated content, participating in link exchange, buying expired domains, and creating own spam farm [4].

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 those [7]. Content hiding is used to make spam items invisible. One simple method is to make the spam terms the same color as the page background color. In cloaking, spam web servers return an HTML document to the user and a different document to a web crawler. In redirecting, a spammer can hide the spammed page by automatically redirecting the browser to another URL as soon as the page is loaded. In two latter techniques, the spammer can present the user with the intended content and the search engine with spam content [8].

2.2 Spam Detection Methods

Various methods have been proposed to combat web spamming and to detect spam pages. An important and basic one is TrustRank algorithm [9]. The idea of this algorithm is that spam pages often point to authoritative pages but authoritative and non-spam ones seldom points to spam pages. TrustRank works as follows: a small set of authoritative pages is selected by an expert to be the seed set. Then, the link structure of the web is utilized to discover other food and non-spam pages. In fact the trust of the seed set propagates the web through other pages.

Some link analysis models are proposed to counter the influence of linked-based manipulation. These models improve spam resilience of web link analysis and support more effective and robust ranking in comparison with existing algorithms such as PageRank [1, 2, 3].

Another common method is considering web spam detection as a problem of classification [5]. In these kinds of methods, some web pages are collected as training data and labeled as spam or non-spam by an expert. Then, a classifier model is learned from the training data. One can use any supervised learning algorithm to build this model. Further, the model is used to classify any web page to spam or non-spam. The key issue is to design features used in learning. Ntoulas et al. [5] propose some content-based features to detect content spam. Link-based features are proposed for link spam detection [10]. In [11] Liu et al. propose some user behavior features extracted from access logs of web server of a page. These features depict user behavior pattern when reaching a page (spam or non-spam). These patterns are used to separate spam pages from non-spam ones, regardless of spamming techniques used. Erdelyi et al. [12] investigate the tradeoff between feature generation and spam classification accuracy. They conclude that more features achieve better performance; however, the appropriate choice of the machine learning techniques for classification is probably more important than devising new complex features.

3. SEMI-SUPERVISED LEARNING

In Supervised Learning, the learning algorithm uses labeled data as training data to build a classification model. This model is used to classify future data. One of the drawbacks of this method is that a large number of labeled training instances are needed. Since, labeling is often done manually; it can be labor intensive and time consuming. To reduce the labeling effort, semi-supervised learning is suggested [7].

In semi-supervised learning, only a small set of instances is required to be labeled for each class. However, since a small set of instances are not sufficient for generating an accurate classifier, a large number of unlabeled instances are utilized to help. The generated classifier is used to predict class of the unlabeled instances [7].

In this article, we use the EM algorithm with naïve Bayesian classification as our semi-supervised learning algorithm.

3.1 EM Algorithm with Naïve Bayesian Classification

A popular algorithm of semi-supervised learning is Expectation-Maximization (EM) algorithm that uses naïve Bayesian classification to find and fill in missing data. It consists of two steps, the Expectation step (or E-step), and the Maximization step (or M-step). The E-step basically fills in the missing data based on the current estimation of the parameters. The M-step re-estimates the parameters to maximize the likelihood. This leads to the next iteration and so on. EM converges to a local minimum when the model parameters stabilize [13].

Algorithm EM(*L, U*)

1 Learn an initial naïve Bayesian classifier *f* from only the labeled set *L*

2repeat

// E-Step

3for each example d_i in *U* do

4 Use the current classifier *f*to compute $\Pr(c_j \mid d_i)$

5end

// M-Step

6learn a new naïve Bayesian classifier *f* from $L \cup U$ by computing $\Pr(c_j)$ and $\Pr(w_t \mid d_i)$

7 until the classifier parameters stabilize

Return the classifier *f* from the last iteration.

Fig1:**The EM algorithm with naïve Bayesian classification [7]**

Let the set of classes be $C = \{c_1, c_2, ..., c_{|C|}\}$.$|C|$, in spam classification, is 2, i.e., spam or non-spam. Figure 1 shows the EM algorithm, where L denotes the labeled set, U denotes the unlabeled set, and d_i denotes the documents (instances).

If the documents of the unlabeled set are regarded as having missing class labels (values), EM can fill in these missing class labels by the returned classifier. That is how EM is used in semi-supervised learning.

4. THE NEW METHOD

One of the most important and difficult phases of web spam detection by classification is labeling the training data. Labeling is done by some experts active in the area of adversarial information retrieval and aware of spamming techniques. It's the only phase done by a human.

Labeling is a difficult task because it is hard to find some experts familiar with spamming techniques and capable of

distinguishing spam and non-spam pages. Even if you can find some, it takes a long time to review a large set of web data and label them. There is also the possibility of biased labeling because there is no exact rule dedicating spam pages. Castillo et al. [6] explain they provided the experts some guidelines to help them with labeling, but
The evaluation of borderline cases is very subjective. Indeed websites that use spam techniques also provide some contents, so that is very difficult to classify them as spammers [6].
To resolve this problem, we use EM algorithm that learns a classification model from a small set of labeled examples and a large set of unlabeled examples. This classification model labels the unlabeled examples with spam or non-spam. After labeling, we will use these examples as the training data. Figure 2 shows the process of labeling the training data by learning from the small labeled samples, where *NB* is the naïve Bayesian learning algorithm, *f* is the classification model, *L* denotes the labeled samples of the training data, *U* is the unlabeled set, and *PU* (Predicted *U*) is the unlabeled set after being labeled by the classification model. The process stops when the new *PU* classes are not different from previous ones.
At first, the class of each sample of the *U* is predicted using the *L* and the naïve Bayesian learning algorithm. The set *U* along with the predicted classes are saved as *PU*. Then, a new classifier is learned from $L \cup PU$. This new classifier predicts the classes of samples of the *U*. The *U* along with the new predicted classes are saved as the new *PU*. If the new *PU* is not different from the previous one, it means the algorithm has converged, so the labeling process stops; else, the next iteration starts; A new classifier is learned from $L \cup PU$, etc.

The output of the process is the training data ($L \cup PU$), having class labels for all instances. These data can be used by any learning algorithm like naïve Bayesian, C4.5, etc. to build a classification model that can distinguish spam pages from non-spam ones. We use the naïve Bayesian algorithm in thenew method, because as you will see in part 4.2, it outperforms other learning algorithms. A search engine can use this model to detect spam as follows. When the crawler finds a new website, before indexing its pages, classifies them to spam and non-spam using this model. If a web page is detected as spam, that page or all of the web pages of the web site may not be indexed. As a result, not only the search engine doesn't suffer from indexing cost, but also gainsthetrust because users of this search engine remain satisfied with it.

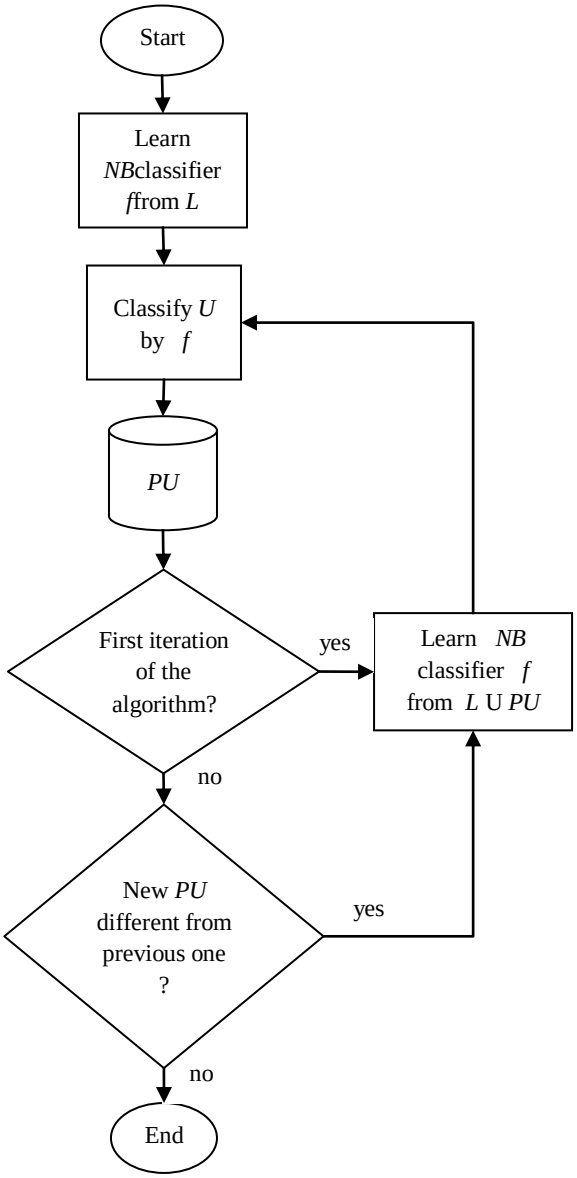

Fig2:Labeling the training data by learning from small labeled samples

5. EXPERIMENTAL RESULTS

The proposed method has been implemented via various learning algorithms. The result of the implementation is compared with various supervised methods. For the sake of universality, the proposed method is evaluated with different labeled sets and feature sets.

5.1 Data Set

We use WEBSPAM-UK2007 data set, a publicly available web spam data collection [14]. It is based on a crawl of the .uk domain done in May 2007. It includes 105 million pages and over 3 billion links in 114,529 hosts. The training set contains 3849 hosts.
Hosts of this reference collection were labeled by a group of volunteers as "normal", "borderline", or "spam". Each host was labeled by at least two persons independently [6].
The benefits of labeling at the host level instead of page level is that a large coverage can be obtained, meaning that the

sample includes several types of web spam, and the useful link information among them [15].
This data set contains content and link based features. Some of the important content features are "number of words in the page", "number of words in the title", "average word length", "compression ratio", "entropy", etc. These features are calculated for home page, page with maximum PageRank, and an average value for all pages of every host.

5.2 Performance Evaluation

In order to evaluate the proposed method, we used the training set of WEBSPAM-UK2007 data set. We partitioned this set into two sets, labeled (*L*) set and unlabeled (*U*) set. We used only content based features because they were enough to meet our purposes. The selected data set contains 2040 data, with 126 spam and 1914 non-spam pages. After labeling with the proposed method, we used the output data as training data and built a classification model with various learning algorithms. The model was tested with a test set of size 1809 data, with 82 spam and 1727 non-spam pages.
The evaluation of the overall process is based on a set of measures commonly used in machine learning and information retrieval [16] and suitable for the spam detection task. Given a classifier, we consider its confusion matrix in table 1.

Table 1.Confusion matrix

	Classified spam	**Classified non-spam**
Actual spam	a	b
Actual non-spam	c	d

We consider the following measures:

Recall= (# of spam / # of all data)*R(s) + (# of non-spam / # of all data)*R(n) (1)

Precision = (# of spam / # of all data)*P(s) + (# of non-spam / # of all data)*P(n) (2)

F-score = (2 * Recall * Precision) / (Recall + Precision) (3)

Where R(s), R(n), P(s), and P(n) are spam recall, non-spam recall, spam precision, and non-spam precision, respectively, and are defined as follows.

R(s) = a / (a + b) (4)

R(n) = c / (c + d) (5)

P(s) = a / (a + c) (6)

P(n) = b / (b + d) (7)

First, the new method is implemented with naïve Bayesian, Bayesian Network, and C4.5 (Decision Tree) learning algorithms. In this experiment, the EM algorithm with naïve Bayesian classification uses the labeled set (*L*) with 20 spam and 20 non-spam, randomly chosen, data to label the unlabeled set. Then, naïve Bayesian, Bayesian Network, and C4.5 learning algorithms are applied to build a classifier. This classifier is tested by the test set explained above. From Table 2, we can see that the naïve Bayesian algorithm outperforms two other learning algorithms. As a result, we implement the new method by the naïve Bayesian algorithm in the rest of the experiments.

Table 2.Implementation of the new method with various learning algorithms (with a labeled set of 20 spam and 20 non-spam data)

Algorithms	**Precision**	**Recall**	**F-score**
Naïve Bayesian	0.918	0.813	**0.86**
Bayesian Network	0.917	0.794	**0.848**
C4.5	0.914	0.78	**0.817**

We compare the new method with naïve Bayesian, Bayesian Network, and C4.5 (Decision Tree) methods. These supervised methods use manually labeled data as the training data. The labeled set (*L*) for the new method contains 20 spam and 20 non-spam, randomly chosen, data. As you can see in Table 3, the proposed method outperforms the naïve Bayesian and the Bayesian Network and is comparable to the C4.5 algorithm. In fact, the new method not only resolves the labeling problem, but also performs very well in comparison with supervised methods.

Table 3.Evaluation of the new method. We compare this method (with a labeled set of 20 spam and 20 non-spam data) with supervised methods of naïve Bayesian, Bayesian Network, and C4.5.

Methods	**Precision**	**Recall**	**F-score**
The new method	0.918	0.813	**0.86**
Naïve Bayesian	0.914	0.148	**0.196**
Bayesian Network	0.94	0.795	**0.852**
C4.5	0.931	0.945	**0.937**

In table 4, you can see the performance of the new method with different labeled sets. A randomly chosen labeled set of sizes 10, 20, and 40, with equal number of spam and non-spam data, achieves a relatively high performance, but a labeled set of size 20 with 15 spam and 5 non-spam is not enough to build an accurate classifier and achieve high performance.
To complete the evaluation task, feature reduction is performed by PCA method. We apply PCA to the feature set three times, with variances 0.95, 0.90, and 0.80; and reduce the number of features to 34, 24, and 15, respectively. The results are shown in table 5. PCA handles sparse data very well [16]. The data set contains many sparse data, so as expected, the feature reduction by PCA improves the performance.

Table 4.Evaluation of the new method with different labeled sets

Labeled sets	**Precision**	**Recall**	**F-score**
20 spam, 20 non-spam	0.918	0.813	**0.86**
10 spam, 10 non-spam	0.9	0.812	**0.851**
15 spam, 5 non-spam	0.944	0.539	**0.662**
5 spam, 5 non-spam	0.905	0.833	**0.867**

Table 5.Evaluation of the new method with different feature sets. The selected labeled set contains 10 spam, 10 non-spam data.

Feature sets	Precision	Recall	F-score
97 original features	0.918	0.813	**0.86**
34 features produced by PCA	0.922	0.799	**0.852**
24 features produced by PCA	0.923	0.841	**0.878**
15 features produced by PCA	0.921	0.879	**0.899**

6. CONCLUSION AND FUTURE WORK

Most of web spam detection methods need a large set of training data. These data should be labeled to spam or non-spam. Manual labeling of a large set of web data can be biased, non-accurate, time-consuming, and labor intensive.

In this paper, we proposed a new method based on the EM algorithm with naïve Bayesian classification to resolve the labeling problem. The new method learns a classification model from a small set of labeled data to label a large set of unlabeled data. After labeling, these labeled data are used to learn a classifier that could classify web pages to spam or non-spam.

Experiments showed that the proposed method not only resolves the labeling problem, but also performs very well in comparison with the supervised methods. We saw that with a labeled set of size 20 or 40, we could achieve a high performance, with savings in time, and cost.

The EM algorithm was utilized in the proposed method, as an important and efficient algorithm of semi-supervised learning. Other semi-supervised learning algorithms like Co-training can be used and compared with the EM algorithm. One can apply optimization methods like Particle Swarm Optimization, Imperialist Competitive Algorithm, etc. to select an efficient subset of features to improve the learning process and achieve a higher performance. These two propositions can improve web spam detection process.

7. REFERENCES

[1] Caverlee, J., Webb, S., Liu, L., Rouse, WB. 2009. A Parameterized Approach to Spam-Resilient Link Analysis of the Web. IEEE Transactions on Parallel and Distributed Systems. 20: 1422-38.

[2] Caverlee, J., Liu, L. 2007. Countering Web Spam with Credibility-Based Link Analysis. In Proceedings of the twenty-sixth annual ACM symposium on Principles of distributed computing (PODC '07). 157-166.

[3] Caverlee, J., Webb, S., Liu, L. 2007. Spam-Resilient Web Rankings via Influence Throttling. 21st IEEE International Parallel and Distributed Processing Symposium (IPDPS).1-10

[4] Gyongyi, Z., Garcia-Molina, H. 2005. Web Spam Taxonomy. First International Workshop on Adversarial Information Retrieval on the Web (AIRWeb'05).

[5] Ntoulas, A., Najork, M., Manasse, M.,Fetterly, D. 2006. Detecting spam web pages through content analysis. In Proceedings of the 15th international conference on World Wide Web. 83-92.

[6] Castillo, C., Donato, D., Becchetti, L., et al. 2006. A reference collection for web spam. SIGIR Forum. 11-24.

[7] Liú, B. 2011. Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data. Springer.

[8] Wang, W., Zeng, G. Tang, D. 2010. Using evidence based content trust model for spam detection. Expert Systems with Applications. 37: 5599-606.

[9] Gyongyi, Z., Garcia-Molina, H., Pedersen, J. 2004. Combating Web Spam with TrustRank. In Proceedings of 30th Intl. Conf. on Very Large Data Bases (VLDB'04). 576-587.

[10] Becchetti, L., Castillo, C., Donato, D., Leonardi, S., Baeza-Yates, R. 2006. Link-based characterization and detection of Web Spam. 2nd Int Workshop on Adversarial Information Retrieval on the Web (AIRWeb'06). 1-8.

[11] Liu, Y., Cen, R., Zhang, M., Ma, S.Ru, L. 2008. Identifying web spam with user behavior analysis. In Proceedings of the 4th international workshop on Adversarial information retrieval on the web. 9-16.

[12] Erdelyi, M., Garzo, A.,Benczur, AA. 2011. Web spam classification: a few features worth more. In Proceedings of the 2011 Joint WICOW/AIRWeb Workshop on Web Quality. 27-34.

[13] Mitchell, T. 1997. Machine Learning. McGraw-Hill.

[14] Yahoo Research. 2007.Web Spam Collections, http://barcelona.research.yahoo.net/webspam/datasets/, accessed May 2011.

[15] Castillo, C., Donato, D., Gionis, A., Murdock, V., 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. 423-30.

[16] Han, J., Kamber, M., Pei, J. 2011. Data Mining: Concepts and Techniques. Elsevier.