Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

A Boosted Semi-supervised Learning Framework for Web Page Filtering

Zhu He, Xi Li and Weiming Hu
National Laboratory of Pattern Recognition
Institute of Automation, Chinese Academy of Sciences
No. 95 East Zhongguancun Road, 100190, Beijing, China
hezhucn@gmail.com

***Abstract*—The World Wide Web provides great convenience for users to obtain information. However, there exists much harmful information on the internet, such as pornographic content and prohibited drugs' information. Thus, how to filter harmful web pages on the internet is quite an important issue. In general, the problem of harmful web page filtering is converted to that of web page classification, which needs plenty of well labeled training samples. However, the cost of labeling a large set of web pages is very expensive. To address this problem, we adopt a semi-supervised framework for web page filtering. In this framework, each web page is represented by bags of different features, extracted using its HTML structure. Then a semi-supervised learning strategy is taken for efficiently obtaining well labeled training samples. Finally, a boosting classifier is utilized for harmful web page filtering. Experiments have demonstrated the effectiveness of our framework.**

***Keywords*—semi-supervised learning, web page filtering, machine learning**

I. INTRODUCTION

The amount of web-pages has been increasing dramatically with the rapid development of the World Wide Web. Up to the end of year 2008, **Google** had collected more than one trillion web pages. While the Internet provides great convenience for users to obtain all sorts of information that they need, there exists a lot of harmful web pages on the internet, such as pornographic web pages and prohibited drug selling web pages. Harmful information like this has a bad influence on users, especially teenagers. The emergence of these web pages involves the necessity of providing efficient filtering systems to secure the internet access. Therefore, how to recognize such harmful content on the Internet becomes quite an important issue for researchers.

Before the existence of the Internet, researchers had proposed many significant methods for plain text classification. And as the problem of web page classification emerges, text classification techniques have been inducted into this new field. Traditional text classification methods are widely used in web page filtering problems, such as k-nearest neighbors (kNN) [1], Naïve Bayes [2], Decision Trees [3] and Support Vector Machines (SVM) [4].

However, there exist some differences between web pages and plain texts. A web page is a resource of information, which is usually in HTML or XHTML format, and may provide navigation to other web pages via hypertext links. A web-page has internal structural information marked within HTML tags and external connections via hyperlinks. These tags and hyperlinks make a web page more complex, but can be also used as additional information in classification.

Recent efforts rely more on content analysis of web pages. Wu [5] proposed a CNN-like word net to extract and represent semantic and statistic features of texts, and analyzed different types of keywords and their interactions to recognize objective web pages. Du [6] proposed a general method for pornography web page filtering, which can be applied to filter web pages of other fields. Similarities between the input web page and all training webpage samples are averaged and compared with a threshold to determine the label of the input page. Ho [7] uses the Bayes classifier to recognize pornographic texts, based on structural and statistical analysis of web pages. Not only the influence of different terms on the weights of the Bayes network is considered, but different weights are also assigned to the same terms when they appear in different Web page components such as Title, Meta, and Body.

The shortcomings of these methods mentioned above are obvious: 1) Keywords analysis based filtering systems rely heavily on the construction of keyword lists, which costs much effort. And it might be difficult to find enough discriminative keywords in some fields. 2) Supervised learning methods require a large set of high-quality labeled samples, which are difficult to obtain. 3) Some of these methods do not considered the difference between web pages and plain texts. This may lead to insufficiently use of information in a web page and lower the filtering performance.

In recent years, semi-supervised learning (***SSL***) shows its conveniences and advantages in classification tasks when the available labeled sample set is quite small. Thus, we propose a semi-supervised learning framework in this paper. Our filtering demand is expressed by a set of webpage instances, indicating which pages are ones to be filtered and which are not. A web page filter is then created out of our system. The rest of this paper is organized as follows: In Section 2, our semi-supervised web page filtering framework is described in detail. In Section 3, we report our experimental results on a task of filtering prohibited drug web pages. In Section 4, the paper is concluded.

978-1-4244-2794-9/09/$25.00 ©2009 IEEE

II. SEMI-SUPERVISED FRAMEWORK

In this framework, we first extract three different types of features from each web page, and then use an ***SSL*** strategy to obtain more high-quality labeled samples. After that, we construct three classifiers by boosting over the expanded training set. Finally, these three classifiers are incorporated into a hierarchical filtering structure.

A. Data Representation

A web page is a resource of information formed by structured text and other elements. It has structural information marked within HTML tags, and usually has hypertext links which provide navigation to other web pages. Due to the differences between plain texts and web pages, apply text classification methods on them directly may lead to insufficient use of information.

Figure 1. An Example of Features in a Web Page.

In our framework, the training data includes two sets of web pages. One is a very small set of manually labeled web pages; the other is a large set of unlabeled web pages collected automatically from the internet by crawler. For a better use of the information in a web page, we divide a page into different information sources based on its HTML tags [8]. More specifically, these tags are:

- Web page URL (URL)
- TITLE
- METADATA
- BODY (mainly content text)
- ANCHOR TEXT and HREF (ANCHOR)
- Multi-media content (e.g. images and videos)

Among these sources we choose content text, title and anchor text to represent the web page (Figure 1), as they usually contain relatively rich textual information and generally appear in most web pages. And the elimination of the other components would not affect the performance significantly because comparing to the chosen features they either contain much less textual information and relatively more noise, or usually are left blank by the author of the page. Each chosen information source in a web page can be represented by a bag of words. Therefore each web page from both the labeled and unlabeled training set is then represented as three bags of features. We refer to these bag-of-words feature sets by ***Title***, ***Content*** and ***Anchor***, respectively.

B. Real Adaboost Algorithm

Boosting is one of the most important recent developments in classification methodology. Our work is based on ***AdaBoost***, the most popular boosting algorithm, introduced by Freund and Schapire [9]. It consists in combining low quality classifiers or weak hypotheses, with a voting scheme to produce a classifier better than any of its components (Figure 2).

Given: Training samples $(x_1, y_1)\ldots(x_n, y_n)$, *where* $x_i \in \chi$, $y_i \in \{-1, +1\}$

1. *Assign initial weights* $w_i^{(1)}$ *to the samples,* $w_i^{(1)} = 1/n$;
2. *For* $t = 1, \cdots, T$:
   a) *Train weak learner according to the distribution of sample weights* $w_i^{(t)}$:
   b) *Get weak hypothesis* $h^{(t)}: \chi \to \mathbb{R}$
   c) *Choose* $\alpha^{(t)} \in \mathbb{R}$
   d) *Update weights:*

$$w_i^{(t+1)} = \frac{w_i^{(t)} \exp\left(-\alpha^{(t)} y_i h^{(t)}(x_i)\right)}{Z^{(t)}}$$

Where $Z^{(t)} = \sum_{i=1}^{n} w_i^{(t)} \exp\left(-y_i \alpha^{(t)} h^{(t)}(x_i)\right)$ *is a normalization factor.*

Output: Strong hypothesis $H(x) = sign\left(\sum_{t=1}^{T} \alpha^{(t)} h^{(t)}(x)\right)$

Figure 2. A Generalized version of Adaboost Algorithm.

If we allow the weak hypotheses to be real-valued rather than binary, which means we restrict ***h***: ***X***→[0, 1], the algorithm is called Real Value Adaboost, or ***Real Adaboost*** [10] (Figure 3). In this paper, we use ***Real Adaboost*** algorithm, with stumps as weak classifier, to implement classifiers in our filtering framework.

Given: Training samples $(x_1, y_1)\ldots(x_n, y_n)$, *where* $y_i \in \{-1, +1\}$

1. *Assign initial weights* $w_i^{(1)}$ *to the training samples,* $w_i^{(1)} > 0$, $\sum_{i=1}^{n} w_i^{(1)} = 1$;
2. *For* $t = 1, \cdots, T$:
   a) *Estimate weighted conditional probability:* $p^{(t)}(x) = \hat{P}_w(y = 1 \mid x) \in [0,1]$;
   b) *Let*

$$f(x) = \frac{1}{2}\log\left(\frac{p^{(t)}(x)}{1 - p^{(t)}(x)}\right);$$

   c) *Re-calculate sample weights*

$$w_i^{(t+1)} = \frac{w_i^{(t)} \exp\left(-y_i f^{(t)}(x_i)\right)}{Z^{(t)}},$$

Where $Z^{(t)} = \sum_{i=1}^{n} w_i^{(t)} \exp\left(-y_i f^{(t)}(x_i)\right)$ *is a normalization factor.*

Output: Strong classifier $H(x) = sign\left(\sum_{t=1}^{T} f^{(t)}(x)\right)$

Figure 3. Real Adaboost Algorithm.

C. Semi-supervised Learning Strategy

Suppose we have manually labeled a small set of web page data, which consists of both positive and negative samples. We denote this labeled set by L. A sample from L is represented as $l=<l_1^{(i)}, l_2^{(i)}, l_3^{(i)}>$; l_1, l_2 and l_3 correspond to the features of ***Content***, ***Title*** and ***Anchor*** in a web page. We also have a large set of unlabeled web pages crawled by spider, denoted as U. Each unlabeled web page is also split as three bags of words, corresponding to the ***Content***, ***Title*** and ***Anchor*** features. Our goal is to find an effective way to predict the label of a new web page. In this problem, L is small. So it is unlikely to train a satisfactory classifier using supervised training methods. To solve this problem, we use semi-supervised learning method on the unlabeled set of web pages, U, to obtain better performance.

The pseudo code of our training algorithm is shown in Figure 4. Given a set of labeled training samples L and a set of unlabeled training samples U, the algorithm then iterates the following procedure. We first use the data of the labeled set L to train three different classifier h_1, h_2 and h_3, and each of the three classifiers considers only the ***Content***, ***Title*** or ***Anchor*** features respectively. Second, h_1, h_2 and h_3 are applied on the unlabeled set U. Each classifier considers its own type of features, and the n_1 samples it most confidently labels as positive, and the n_1 samples it most confidently labels negative are selected. The selected samples are added to L, along with the labels assigned by the classifiers that selected it. Finally after the classifier construction and the sample selection, we get an updated labeled set L and an updated unlabeled set U to continue the procedure.

Given:
*a set **L** of labeled training samples*
*a set **U** of unlabeled samples*
Loop N times:
*1) Use **L** to train a classifier h_1, h_2 and h_3. Classifier h_1 considers only the **Content** features, h_2 considers the **Title** features and h_3 considers **Anchor** features respectively;*
*2) Apply h_1, h_2 and h_3 on **U**, each of the classifier labels p_1 positive and n_1 negative samples from **U** respectively;*
*3) Update **L** and **U**: Remove these self labeled samples from **U** to **L**.*
Loop end

Figure 4. Semi-supervised Learning Algorithm.

After performing the learning algorithm above, we have got a new set of labeled data, ***L'***. ***L'*** contains both the original labeled sample set L and the samples that are most confidently selected and labeled by the temporarily trained classifiers. Then we are able to construct classifiers over a much larger "labeled" training set ***L'***.

We also train classifiers with only the labeled data set L to compare ***SSL*** method to supervised learning method. The conducted experiment (See Table II and III) shows the effectiveness of our SSL training framework.

D. Hierarchical Classification Structure

Over the new "labeled" training set ***L'***, we can train three distinct classifiers, h_1, h_2 and h_3, and each of the three classifiers considers only the ***Content***, ***Title*** or ***Anchor*** features respectively. When a new web page comes, the page is then parsed into ***Content***, ***Title*** and ***Anchor*** features. Each type of feature is processed by the corresponding classifier. As we described in Section 2, the outputs of these classifiers are real numbers. The sign of the output is the predicted label given by the classifier. It is reasonable to consider the absolute value of the output as a confidence degree about their predictions.

Experiments are conducted to measure the classification performance on each type of features (See Table II and III). According to the experimental results, the classifier on ***Content*** feature outperforms the two other classifiers in both error rate and the standard deviation of error rate, which means the ***Content*** classifier is more accurate and stable than the other two classifiers. We could explain this from another aspect: the ***Content*** feature based web page classifier is more reliable as the ***Content*** feature usually contains richer textual information, which well characterizes a web page. So we adopt a hierarchical structure to organize the classifiers. ***Content*** classifier h_1 is set as the top layer of the filtering structure. Classifiers h_2 and h_3 are fused by sum-rule score fusion to be set as the bottom layer of the classification structure, shown in Figure 5. The transfer from top to bottom is based on the comparison between h_1's output and a confidence threshold θ, which is manually set from 0.7 to 0.9.

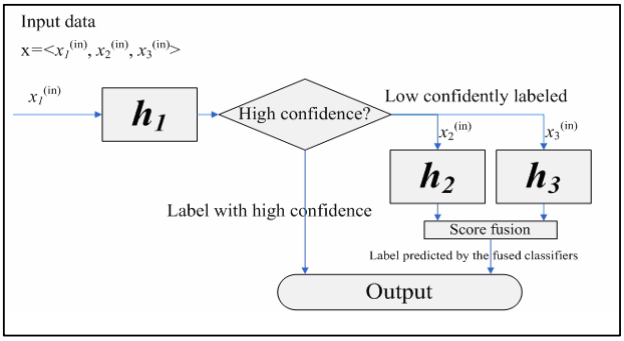

Figure 5. Hierarchical Classification Structure.

III. EXPERIMENTS

We choose the task of prohibited drugs' information (e.g. selling, experience or producing) webpage filtering to test the performance of our framework. We collected 811 English web pages from different prohibited drugs' information web sites and 819 from normal sites. In the normal set, we intended to include a number of drug-related web pages (such as harmless web pages about hemp clothes, or medical drugs), which usually contain keywords that appears in target web pages and might be misclassified by some methods. We randomly divide

TABLE I. DETAILS OF THE DATA SETS

	Drug info. set	*Normal set*	*Total*
Labeled L	**32**	**32**	**64**
Unlabeled U	**200**	**200**	**400**
Test set	**579**	**587**	**1166**
Total	**811**	**819**	**1630**

our dataset into three subsets, summarized in Table I. All the web pages in our dataset are parsed into ***Content***, ***Title*** and ***Anchor*** features. Then these word features are stemmed [11], and the stop-words are removed. Then we use ***tf/idf*** method [12] to generate feature vectors for each type of features respectively.

To compare ***SSL*** to supervised learning, ***Content***, ***Title*** and ***Anchor*** classifiers are trained using only the labeled training samples in ***L***. We also combine these three classifiers by a majority voting scheme. Error rates and their standard deviations are calculated and summarized in Table II and III. The result shows that due to the use of the unlabeled data, our method reduces classification error by up to 26%, and the standard deviation of the error rate is smaller, indicating the ***SSL*** classifiers are more stable. The results show that semi-supervise learning method outperforms supervised method when available labeled sample set is small.

TABLE II. ERROR RATE FOR CLASSIFYING WEB PAGES

	Content	***Title***	***Anchor***	***Combined***
Supervised	0.1345	0.2362	0.1426	0.1236
SSL	0.1034	0.2194	0.1206	0.0912

TABLE III. STANDARD DEVIATIONS OF THE CLASSIFICATION ERROR

	Content	***Title***	***Anchor***
Supervised	0.01238	0.02134	0.01608
SSL	0.00913	0.01811	0.01373

In order to evaluate the performance of our proposed method, we compare our filtering results with a typical ***SSL*** method proposed by Blum [13]. We set the confidence threshold θ = 0.75 and use ***accuracy*** (***acc***), ***precision*** (***prec***), ***false positive rate*** (***fpr***) and ***F1 Score*** as evaluation criterias. The standard deviations of these parameters are included to show the stability of each method on small labeled sets. The testing results are shown in Table IV and V. According to the results, our hierarchical-structured filtering framework outperforms the comparing one in both performance and stability.

TABLE IV. TESTING RESULTS (θ = 0.75)

	acc	***pre***	***fpr***	***F1***
Our Method	**0.9316**	**0.9287**	**0.0687**	**0.9306**
Blum ***SSL***	0.9189	0.9239	0.0763	0.9179

Each method was run by 15 times to calculate the average values.

TABLE V. STANDARD DEVIATIONS OF THE TESTING RESULTS

	std***(acc)***	std***(prec)***	std***(fpr)***	std***(F1)***
Our Method	**0.0097**	**0.0152**	**0.0163**	**0.0109**
Blum ***SSL***	0.0127	0.0184	0.0210	0.0125

Each method was run by 15 times to calculate the standard deviations.

IV. CONCLUSION

In this paper, we have proposed a boosted semi-supervised web page filtering framework, based on a textual and structural content analysis of HTML documents. In web page filtering problems, the available labeled sample set is usually small, and manually labeling web pages will cost a lot of human labor. However, unlabeled web page samples are available freely and in abundance on the internet. So it is important to make use of the unlabeled data. The experimental results show the efficiency of using unlabeled data and the significant improvements by using a hierarchical classification structure in our prohibited drug information filtering task.

The filtering demand of our framework is expressed by a small set of webpage instances, indicating which pages are the ones to be filtered and which are not. It is convenient to apply our framework on other filtering tasks simply by constructing a small labeled web page set manually, and a large set of unlabeled web pages from the internet using web page crawler.

To extend our work, we wish to import common sense knowledge into our filtering method. It is also significant to combine multi-media and textual information together in web page filtering.

REFERENCES

[1] Y.Yang and X. Liu, "A re-examination of text categorization methods," in Proceedings of the 22nd Annual International ACM SIGIR conference on Research and development in information retrieval, pp. 42-49, 1999.

[2] D.D. Lewis and M Ringuette, "A Comparison of Two Learning Algorithms for Text Categorization," in Proceedings of the 3rd Annual Symposium on Document Analysis and Information Retrieval, 1994.

[3] T. Mitchell, Machine Learning. McGraw Hill, 1996.

[4] V. Vapnic, The Nature of Statistical Learning Theory, Springer, New York, 1995.

[5] O. Wu and W. Hu, "Web Sensitive Text Filtering by Combining Semantics and Statistics," in Proceedings of 2005 IEEE International Conference on Natural Language Processing and Knowledge Engineering, pp. 663-667, 2005.

[6] R. Du, R. Safavi-Naini and W. Susilo, "Web Filtering Using Text Classification," in Proceedings of the 11th IEEE International Conference on Network, pp. 325 - 330, 2003.

[7] W.H. Ho and P.A. Watters, "Statistical and Structural Approaches to Filtering Internet Pornography,", in Proceedings of IEEE International Conference on System, Man and Cybernetics, vol. 5, pp. 4792-4798, Oct. 2004.

[8] Nitin Agarwal, Huan Liu, and Jianping Zhang, "Blocking objectionable web content by leveraging multiple information sources," in SIGKDD Explorations, vol. 8, pp. 17–26, 2006.

[9] Y. Freund and R. E. Schapire, "A Decision-theoretic Generalization of On-line Learning and An Application to Boosting," in Proceedings of the 2nd Euro. Conference on Computational Learning Theory, pp. 23-37, 1995.

[10] R. Schapire, and Y. Singer, "Improved boosting algorithms using confidence-rated predictions," Machine Learning, vol. 37, pp. 297–336, 1999.

[11] Porter, An algorithm for suffix stripping, Program, Vol. 14, no. 3, pp 130-137, 1980.

[12] G. Salton, & M. J. McGill, Introduction to Modern Information Retrieval, McGraw-Hill, New York, 1983.

[13] A. Blum and T. Mitchell, "Combining Labeled and Unlabeled Data with Co-training", In Proceedings of the eleventh annual conference on computational learning theory, pp. 92–100, 1998.