Camera-ready version of Proceedings of 12th International Conference on Knowledge-Based Intelligent Information and Engineering Systems (KES 2008), Part II, LNAI 5178, pp. 774 - 781, September, 2008

A Spam Filtering Method Learning from Web Browsing Behavior

Taiki Takashita, Tsuyoshi Itokawa, Teruaki Kitasuka, and Masayoshi Aritsugi

Department of Computer Science and Communication Engineering, Graduate School of Science and Technology, Kumamoto University, Kumamoto 860-8555, Japan
reo@st.cs.kumamoto-u.ac.jp,
{itokawa, kitasuka, aritsugi}@cs.kumamoto-u.ac.jp

Abstract. In this paper a spam filtering method is proposed. We focus on user behavior that most email users browse the Web. The method reduces troublesome maintenance of the spam filter, since the filter learns from Web browsing behavior in the background. The method uses Web browsing behavior of each user to learn ham words. Ham words are picked up from browsed Web pages using TF-IDF and stored in the database called ham words list. For each received email, the method extracts keywords from the email, including Web pages of the URLs. If some keywords are in the ham words list, the email is treated as a ham. In our experiments, several spam emails which cannot be detected by a Bayesian filter are detected as spams.

1 Introduction

Spam email accounts for 90 to 95 percent of all email in 2007, up from an estimated five percent of email in 2001, and spam emails become the worse form of junk advertising than postal junk mails and telemarketing calls[1]. Spam filtering is required for not only technical reasons such as overspend the network bandwidth and email storage, but also social issues such as child safety, phishing email, and so on. We are already hard to find ham emails without a kind of anti-spam technologies.

The major anti-spam technologies are categorized into sender-side technologies or receiver-side technologies. The filtering methods which this paper concerns are categorized the latter. The former is to prevent spammer from sending email. Outbound port 25 blocking of ISP is an example of the sender-side technologies.

In this paper, we focus on user behavior that most email users also browse the Web. Conventional spam filters use information extracted from emails. The proposed method learns the user preference from Web browsing behavior. The merit of the method is reduction of maintenance task of the filter, since it learns the user preference in background of browsing behavior. We show the basic concept and design of our method, and results of preliminary experiments in this paper.

This paper is organized as follows. In Section 2, related work is introduced. The proposed method is described in Section 3. In Section 4, evaluation results of the proposed method are shown. Finally we conclude the paper in Section 5.

2 Related Work

Anti-spam research and development is an ongoing battle with both spammers and spam fighters. It's becoming ever more sophisticated[2]. There are many researches of spam filtering. Work on spam filtering can be divided into two categories: content-based approach and collaborative approach. On the content-based approach, the classification of an email is based on an analysis of the content of the email. The second is collaborative approach, which depends on the collaboration of groups of users to share information about spam[3].

Both approaches are widely used in these days. Collaborative filtering is employed by Web mail service providers. Bayesian filtering, which is a kind of content-based approach, is mostly built in MUA (mail user agent).

The collaborative filtering is a server-based approach. It shares spam information between many users. Once a user reports a received email as a spam to the server, the server updates information of spam. Then, other users will see the email that has the same content with a spam flag or in a spam folder. This filtering is in broad category of folksonomy. Since the filter is maintained globally, unique preference of each user is hard to be reflected into the filter. To make custom filter for each user, the filter maintenance is needed somewhat by the user as same as Bayesian filter.

Bayesian filter is a content-based approach[4]. Many MUAs adopt the filter to detect spam. Basic concept of Bayesian filter is based on Bayesian combination of the spam probabilities of individual words in an email. All received emails are classified into spam or ham, according to the threshold of the probabilities. To classify emails, the filter has corpuses of spams and hams. These corpuses are maintained by users themselves, since the probability of false classification depends on them.

In addition, there are many kinds of content-based approaches. In [5], the method that processes email messages as image data is proposed. When we manually filter the spam, we glance at a message as an image instead of reading it carefully. The method detects spam by transforming a received email into image in accordance to HTML tag structure.

Note that the maintenance of these filtering methods is very tedious and expensive task, since it usually takes long time to get the necessary information for the maintenance. In this paper, we propose a novel method that gets information from Web browsing.

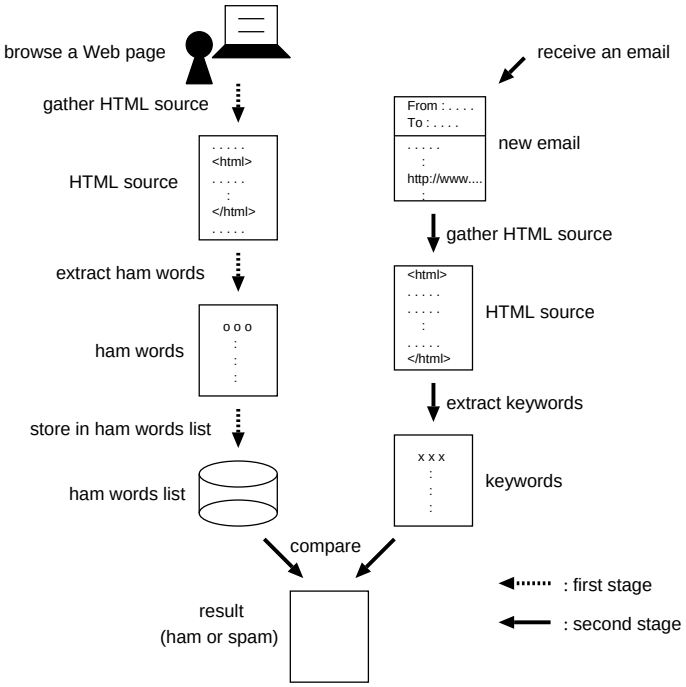

Fig. 1. Overview of the proposed method

3 Our Spam Filtering Method

3.1 Structures

On an assumption that most of the email users browse Web pages, we try to examine the spam filtering method which learns the user's preference from the Web browsing behavior. The user's preference of the email will be quite similar to that of the browsed Web pages. The proposed method provides an individual filter to each user. The filter has a database called ham words list, which is created from the Web browsing behavior of the user. The preference varies according to an interest of each user. Therefore we think that the content of the browsed Web pages is suitable to learn the preference of the user. Our approach was inspired by [10], in which an information management assistant gathers contextual information from user interactions and leverages it to support just-in-time information access.

The method needs to develop the interface between spam filter and the browser. However the method makes many users free from some part of the maintenance of the Bayesian filter or collaborative filtering systems.

The method consists of two stages: the first stage is a creation of ham words list, and the second stage is a filtering of received emails with ham words list. Fig.

1 shows the outline of the stages. At the first stage, when the user browses Web pages day-by-day, the HTML sources of the pages are gathered to extract the ham words. The words in the HTML source are processed according to TF-IDF (Term Frequency–Inverse Document Frequency)[6]. For each page, some words which have high TF-IDF score are stored in the ham words list.

At the second stage in Fig. 1, the method filters the received email with the ham words list created in the first stage. To judge the email, we have to pick up the words from the received email. The filter refers the one or more URLs in the email to retrieve keywords of the email. The flow of retrieval of keywords is similar to the first stage. Finally to determine the email is a ham or a spam, relevancy of the email to the user preference is tested through matching of ham words list and keywords of the email. The detail of each stage is described in the remaining part of this section.

3.2 The Ham Words List of Web Browsing Behavior

At the first stage, the ham words list is updated according to the Web browsing behavior of each user. The ham words list is used in the second stage to determine the received email is a ham or a spam.

The ham words list will be updated using Web pages that the user browses. For each Web page, HTML source of the page is processed to find new ham words. Words in the HTML source are weighted by TF-IDF. TF-IDF score of a word t_i in the HTML source d_j is defined as follows:

$$\begin{aligned} \text{TF-IDF}_{i,j} &= \text{TF}_{i,j} \times \text{IDF}_i \\ &= \frac{n_{i,j}}{\sum_k n_{k,j}} \times log \frac{|D|}{|\{d : d \ni t_i\}|} \end{aligned}$$

where $n_{i,j}$ is the number of occurrence of the word t_i in document d_j, $|D|$ is the total number of documents, and $|\{d : d \ni t_i\}|$ is the number of documents in D which contain the word t_i.

Currently $\text{TF-IDF}_{i,j}$ of each word t_i is calculated using Yahoo! API[7]. $|D|$ is treated as the total number of sites which Yahoo crawls. $|\{d : d \ni t_i\}|$ is treated as the number of sites returned from Yahoo! contextual Web search for the word t_i.

$n_{i,j}$ of TF-IDF can be calculated only from the HTML source of the Web page. In Section 4, we use emails and Web pages written in Japanese. Before we calculate TF-IDF score of each word, morphological analysis for Japanese language is required. We employ a tool of Japanese language morphological analysis called Sen[8]. Only common nouns and proper nouns are the candidates of words to add the ham words list.

Through preliminary evaluation, we find exceptional words which should not be treated as ham words in the list. When one of exceptional words is included in a ham words list, many false negatives occur. False negative means that a spam email is not detected as a spam. Exceptional words is selected heuristically. In the

experiments described in Section 4, we used 20 words as exceptional words, which are Japanese nouns of “search”, “register”, “free”, “member”, “site”, “image”, “login”, “password”, “point”, “year”, “category”, etc.

3.3 Filtering received emails

When a user or a mail server receives an email, the email is judged by the filter which uses the ham words list. The filter can be built into either SMTP server or MUA.

At first, the filter extracts the keywords of the email. In this work, we assume that keywords are extracted from the Web pages linked by URLs in the email body. The keywords of the email are selected according to the same policy of selection of ham words. The top k words of TF-IDF score are selected as keywords of the email. k is varied from 2 to 6 in Section 4.

To judge an email, the filter calculates conformance of the keywords of the email with the words in the ham words list. In the preliminary evaluation described in Section 4, we employ a simple calculation, i.e., if the keywords of the email are contained in the ham words list more than or equal to a threshold number, the email is judged as a ham. Otherwise, it is judged as a spam. The number of keywords is varied in Section 4. We will consider more sophisticated conformance calculation in future.

4 Experiments

4.1 Preliminary evaluation

The proposed method is evaluated through the following environments of a virtual user. We assume that the user has interests in eight categories: childcare, corporate stock, horse races, movies, fortune-telling, news of show business, recipes, and Internet auction. The ham words list was created by 838 Web pages, including about 300 pages of the above categories. 1,000 emails were used as target emails: 500 emails were hams, and the other 500 emails were spams. All emails are picked up from email magazines and actual spams received by the authors. Ham emails are from the magazines of the categories of user’s interest. Spam emails are actual spams and emails categorized into giveaway items, point programs, and adult of email magazines.

Three cases of experimental results are shown in Table 1. Both the number of keywords picked up from each email and threshold of judgement of ham are increased and decreased simultaneously. In all cases, the threshold is a half of the number of keywords. For example of case 1 in Table 1, two keywords are picked up from each email. If at least one of the keywords is found in the ham words list, the email is judged as ham. From Table 1, no ham is judged as a spam (false positive), and 270 spams are judged as hams in this case. In the comparison of keywords and the list, the TF-IDF score of each keyword is not referred in this evaluation.

Table 1. Number of errors using the proposed method

Case	Parameter		Results	
	# of keywords	Threshold	# of false positive	# of false negative
1	2	1	0 (0%)	270 (54%)
2	4	2	8 (1.6%)	251 (50.2%)
3	6	3	7 (1.4%)	240 (48%)

Table 2. Number of errors using Bayesian filter

	# of false positive	# of false negative
Bayesian filter (Thunderbird)	195 (39%)	25 (5%)
Conjunction	–	2 (0.4%)

In this experiment, unfortunately, our method did not achieve good results, since the number of false negatives was very high. The following is the short discussion of the results. The important requirement of spam filtering is low probability of false positive occurrence. False positive is the misjudgement of a ham as a spam. The method can probably stand for the requirement, since no false positive occurred in the case of low threshold. False positive implies a kind of lost of emails, when the case of that the filter brings the ham into a quarantine (i.e. spam folder) instead of inbox. Nowadays user may receive over 100 spam per day. It's hard to find a few ham in a quarantine which contains a large number of spams.

On the other hand, false negative occurred in high probability around 50%. The results show that the method is not enough to judge spam precisely. However the disadvantage will be avoided by combination of conventional spam filters. In Section 4.2, we will discuss this combination.

4.2 Comparison with a Bayesian Filter

The method is compared with the spam filter based on Bayesian filter. As a Bayesian filter, we used Thunderbird[9] that is the popular MUA with Bayesian spam filter. In the experiment, Bayesian filter learns 1,000 ham and 1,000 spam emails before filtering. These emails were selected from archive of email magazines of the same categories in Section 4.1. 1,000 target emails are used as same as Section 4.1.

Table 2 shows the number of false positives and false negatives using Thunderbird. Bayesian filter of Thunderbird misclassifies 39% of ham emails into spams, and 5% of spams into hams.

Firstly, we discuss the false negative which is misjudgement of a ham. False negative using Bayesian filter occurs for 25 emails. These emails contain very short text in mail body and URL. Fig. 2 (a) and (b) are typical examples of the email of false negative by Bayesian filter (the message is written in Japanese). It

おはようございます。
今井さやかちゃん出勤です♪
http://m.garden-gal.com/index.php?ac=gal_detail&cp=652
午前中のご予約に空きがでました !!
お急ぎおといあわせください。
・ 0354285131

解除するには下記 URL にアクセスして下さい。
http://www.emaga.com/tool/automail.cgi?code=garden01&mail=(omitted)

(a) mail body

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html><head><title>渋谷 ホテル型性感 クラブ ガーデン/在籍ギャルデータ</title>
<meta name="keywords" content="渋谷, 渋谷駅, 道玄坂, 百軒店, ヘルス, 派遣型ファッションヘル
ス, ホテルヘルス, ホテヘル, 性感マッサージ,AV 女優,OL, 学生, ギャル, マット,AF">
<meta name="description" content="ClubGarden は渋谷、渋谷駅発のホテルヘルス、ファッションヘル
スです。現役 AV 女優、OL、学生、女子大生、在籍ギャル多数!!詳細な在籍ギャル達のプロフィール等を掲載し
ています。当店では、性感コース、AF コースをご用意しています。"></head><body bgcolor="#daebfc">
<div align="center">
アイドル級のかわいさです !! <br>
<img src="TempImgs/68b3b12e8eebdf96db847b066fea7254.jpg" border="0"><br>S
<a href="/index.php?sessionid=12040913369261951&amp;ac=gal(omitted)</a><br></div>
<hr size="1">
<font color="#bffda9">●</font>今井さやか<font color="#bffda9">●</font><br>
19 歳 T.160<br>
B88(F)W57H85<hr size="1">
ルックス最高レベルです、恋しちゃいます !! <hr size="1">
<a href="/index.php?sessionid=12040913369261951&amp;ac=gal_profile&amp;cp=652">詳
細</a><br>
<a href="/index.php?sessionid=12040913369261951&amp;ac=gal_shift&amp;cp=652">出勤予定
表</a>
<hr size="1">
  営業時間<br>
9：00～23：50<br>
<a href="tel:0354285131">03-5428-5131</a>
<hr size="1">
<a href="/index.php?sessionid=12040913369261951&amp;ac=top">へ</a><br>←戻る
<hr size="1">
<font size="1">
<center>&#169;2007-2008 ClubGarden. All Rights Reserved.</center>
</font>
</body></html>
```

(b) HTML source of first URL in mail body

Fig. 2. An example email of false negative (written in Japanese)

contains only 5 sentences, two URLs, and telephone number in the body. This kind of email is hard to detect as a spam by current Bayesian filter.

Applying the proposed method to these 25 false negative emails, 23 emails can be judged as spams. For all 25 emails, words in the email body is hard to judge correctly. However the method can pick up keywords of the email from the HTML source of URLs in the email body. The results show that the proposed method can cover a weakness of Bayesian filter. To reduce false negative, we can reexamine the email by the proposed method, after an email is determined as a ham by Bayesian filter.

Secondly, we discuss the false positive using only Bayesian filter. The results will not be used for explaining ineffectiveness of Bayesian filter. High probability of false positive of Thunderbird is probably caused by the learning environment of this experiment. All learned hams are from email magazines. For an email magazine, all emails of this magazine are completely judged as spams. There are the following two types of false positive emails.

– many verbose lines of advertisement in the mail body.

– high variability of keywords, e.g., in the category of news of show business there are very wide variety of keywords.

We pick up some false positive emails to analyze the proposed method. The keywords of each email selected by the proposed method are right keywords in each category.

By comparing with Bayesian filter, we conclude that the proposed method has effective situations which are hard to adapt the Bayesian filter.

5 Conclusion

We proposed a spam filtering method that uses Web browsing behavior in this paper. The method retrieves the preference of each user through Web browsed Web pages. We reported preliminary results of experiments. The results show that several spams which Bayesian filter cannot classify as spams can be judged as spams. These spams seem to be hard to classify precisely by Bayesian filter, since they contain a short body of email such as a few sentence and URLs.

As future work, we will consider the scheme to combine with other filters such as Bayesian filter. To combine several filters, we have to manage discrepancy between judgements of filters. The experiments with sophisticated data set such as [11] are also included in our future work.

References

1. Barracuda Networks, Inc.: Barracuda Networks Releases Annual Spam Report. Press Release, 2007. http://www.barracudanetworks.com/ns/news_and_events/index.php?nid=232
2. J. Goodman, G. V. Cormack, and D. Heckerman: Spam and the Ongoing Battle for the Inbox. Communication of ACM, Vol. 50, No. 2, pp. 24–33, 2007.
3. P. Cunningham, N. Nowlan, S. J. Delany, M. Haahr: A Case-Based Approach to Spam Filtering that Can Track Concept Drift. Proc. ICCBR'03 Workshop on Long-Lived CBR Systems, 2003.
4. P. Graham: A Plan for Spam. 2002. http://www.paulgraham.com/spam.html
5. N. Kumagai and M. Aritsugi: On Applying an Image Processing Technique to Detecting Spams. Proc. 21st International Conference on Data Engineering Workshops (ICDEW'05), p. 1172, 2005.
6. G. Salton and C. Buckley: Term-Weighting Approaches in Automatic Text Retrieval. Information Processing and Management, Vol. 24, No. 5, pp. 513–523, 1988.
7. Yahoo! Inc.: Yahoo! Search Web Services. ttp://developer.yaoo.com/search/
8. Sen. (in Japanese) http://ultimania.org/sen/
9. Mozilla: Thunderbird. http://www.mozilla.com/thunderbird/
10. J. Budzik and K. J. Hammond: User Interactions with Everyday Applications as Context for Just-in-time Information Access. Proc. 5th International Conference on Intelligent User Interfaces, pp. 44–51, 2000.
11. I. Androutsopoulos, J. Koutsias, K. V. Chandrinos, and C. D. Spyropoulos: An Experimental Comparison of Naive Bayesian and Keyword-based Anti-Spam Filtering with Personal E-mail Messages. Proc. 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '00), pp. 160–167, 2000.