Real-time statistical rules for spam detection

Quang-Anh Tran, Haixin Duan, Xing Li
Network Research Center, Tsinghua University, Beijing 100084, China

Summary
Spam detections fall into two categories: rule-based and statistical-based. The former refers to the detection which is performed by looking for spam-liked patterns in an email. Since the rules can be shared, they have been popularized quickly. The rules, however, are built manually it is hard to keep them up with the variation of spam. The statistical-based method, on the other hand, is possible to make the detector retrained quickly, but knowledge obtained from this method is unable to be shared among the servers. We, therefore, proposed a statistical rule-based method for spam detection. A widely used rule set - Chinese_rules.cf, for SpamAssassin to catch spam written in Chinese is generated by this method. It can be updated automatically and can also be shared among servers. A generating process of the Chinese_rules.cf is described. Factors that control the rule's performance are discussed.

Keywords
statistical rule-based; spam; detection; Chinese

1. Introduction

Along with the popularity of the Internet services and applications, spam has become one of the major problems in computer network security. Many solutions have been proposed to solve the problem, but they are not completely satisfactory. The sender authentication technology, including SPF (Sender Policy Framework), Caller-ID, Domain Keys and rDNS (Reverse Domain Name Service), tries to identify the legitimacy of the sender of email messages. They can deal with the problem of email forgery, such as Phishing (Goth (2005)), but they can not completely stop spam because spammers can use the sender authentication too. Moreover, they more or less need to change the existing infrastructure of the Internet, i.e. the DNS; therefore, it takes time to widely adopt the technology. Discussion upon this technology has been a subject in the literature of Geer (2004). The second technology is spam detection, which checks the whole email message, especially the content, to identify spam. Since it does not need to change any existing infrastructure of the Internet, this technology has been widely used. In future, despite the sender authentication technology becomes popular the spam detection technology will still be useful for catching spam that pass the sender authentication.

Spam detections fall into two categories: rule-based and statistical-based. The former refers to the detection which is performed by looking for spam-liked patterns in an email, e.g. subject contains "Free". SpamAssassin is the most widely used rule-based system for spam detection. Schwartz (2004) provided a clear, concise guide of deploying the SpamAssassin. The statistical-based method, on the other hand, tries to solve a two-class categorization problem; it uses a training dataset of spam and ham to train the detector. Bayesian algorithms are the most widely used statistical-based method for spam detection. Androutsopoulos (2000) and Graham (2002) had typical works on this subject. Drucker (1999) and Özgür (2004) employed other statistical-based methods such as Neural Network, Support Vector Machines for spam detection.

We define two characteristics to evaluate a spam detector: time characteristic and space characteristic. The time characteristic tells how quick a detector can be generated thus it can be adapted to the variation of spam; the space characteristic identifies how easily a detector can be share among servers (or users), therefore, the knowledge of spam can be popularized quickly. The advantage of rule-based method is that it has good space characteristic. A rule created by someone can be shared to others; therefore, the knowledge of spam can be popularized quickly. The rules,

however, are built manually it is hard to keep them up with the variation of spam. Thus this method has bad time characteristic. The statistical-based method, on the other hand, is possible to make the detector retrained quickly, as long as the training dataset updated in time, the detector can be kept up with the variation of spam. Therefore, the "time characteristic" of this method is very good. The disadvantage of this method is that the knowledge of detector is unable to be shared among servers. Therefore, this method has bad space characteristic. This paper proposes a trade-off between rule-based and statistical-based called statistical rule-based method, in which, rules are generated automatically by a statistical method. This method have all the advantages of the rule-based and the statistical-based method: Since it is a kind of rules, its space characteristic is good; since the rules are generated automatically, its time characteristic is good. A comparison in theory between our method and traditional methods are shown in table 1.

Table 1. Statistical ruled-based method vs. traditional methods

	Time characteristic	Space characteristic
Rule-based	Bad	Good
Statistical-based	Good	Bad
Statistical rule-based	Good	Good

Chinese_rules.cf (http://www.ccert.edu.cn/spam/sa/Chinese_rules.cf), a widely used rule set for SpamAssassin to catch spam written in Chinese (GB2312), is generated and updated automatically by our statistical rule-based method. This paper will discuss technical issues in maintaining the Chinese_rules.cf. This paper is structured as follows. Section 2 presents a short description of SpamAssassin rules. Chinese_rules.cf is a third party drop-in custom rule set for SpamAssassin to catch spam written in Chinese. Section 3 describes the framework for maintaining the Chinese_rules.cf. The framework shows good time and space characteristics of the Chinese_rules.cf. Section 4 analyzes the procedure for generating the Chinese_rules.cf, including some factors that may affect the performance of the rule set. This affection will be discussed in experiments in Section 5. A short description of Chinese_rules.cf in progress will be presented in Section 6. Finally, we will make some discussions and conclude with remarks

2. SpamAssassin rules

SpamAssassin is common software for determining how likely an email message is spam. It uses rule-based spam detection method that compares different parts of email messages with a large set of rules. Each rule adds or removes points from a message's spam score. A message with a high enough score is reported to be spam. Here is an

```
body        DEAR_FRIEND     /^\s*Dear Friend\b/i
describe    DEAR_FRIEND     Dear Friend? That's not very
dear!
```

example of rule in SpamAssassin:

The rule DEAR_FRIEND checks to see if a body part of an email message matches the regular expression "/^\s*Dear Friend\b/i" and adds a score of 0.542 to the message's spam score. An anatomy of a rule was described in details in Schwartz (2004). Due to there is no rule for Chinese mail before, SpamAssassin can not catch Chinese spam effectively. We implement the Chinese_rules.cf as a third party drop-in custom rule set for SpamAssassin to catch spam written in Chinese. Since the differences between rules for Chinese and for English are at body and subject parts of email message, the Chinese_rules.cf contains only body and subject rules. The Chinese_rules.cf is created and maintained automatically by the statistical rule-based method. We will discuss the procedure for generating Chinese_rules.cf in following sections.

3. Framework for maintaining Chinese_rules.cf

A framework to create and maintain the Chinese_rules.cf is proposed as shown in figure 1.

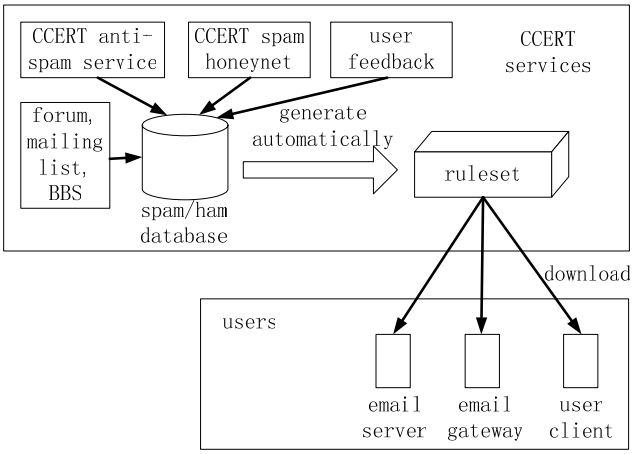

Fig. 1, Framework for maintaining Chinese_rules.cf

In Fig. 1, spam/ham database contains very new and luxuriant Chinese spam and ham. Spam messages come from a variety ways, including CCERT anti-spam service which receives and processes reported Chinese spam all over the world, the CCERT spam honeynet which receives any message that try to send to any unknown user under the domain “ccert.edu.cn”, and user feedback. The ham messages come from well-known SMTH BBS in China, usually 20,000 users online at the same time. Since a post in BBS includes “subject” and “content” parts, we treat them as the “subject” and “body” part of an email. The SMTH BBS is maintained by many students in Tsinghua University, so we assume that the posts in that BBS can present “subject” and “body” part of ham message. Since the Chinese_rules.cf contains only subject and body rules, using the post in SMTH BBS is enough.

We develop a statistical method to generate Chinese_rules.cf automatically based on the database. The algorithm will be focused in the next sections. Since the database is updated every minute, the Chinese_rules.cf has a good “time characteristic”. The Chinese_rules.cf is put on the web and can be downloaded automatically from the web by using the “wget” command under linux/unix environment. User (or server) all around the world can download and use the Chinese_rules.cf conveniently, therefore, the “space characteristic” is good. We call it as a real-time statistical rule for catching Chinese spam.

4. Procedure for generating Chinese_rules.cf

The procedure for generating Chinese_rules.cf consists of 3 steps: Step 1: Pattern retrieval; Step 2: Pattern selection; Step 3: Score assignment. We will describe technical issues in each step in the following subsection.

4.1 Pattern retrieval

Two sets of spam-liked patterns and ham-liked patterns are retrieved from the spam and ham database respectively. We use a Chinese word segmentation technique to retrieve the most frequent patterns in Subject and Body part of messages. A Chinese text appears to be a linear sequence of non-space or equally spaced alphabetic characters as they normally have an associated meaning. Most modern Chinese words consist of more than one ideographic character and the number of characters in words varies. The absence of word boundaries poses a problem for Chinese text retrieval, which is called Chinese word segmentation (Wu (1993)). Researchers, therefore, have been devising various solutions to Chinese word segmentation. Works in this subject was discussed in Foo (2004).

The Chinese word segmentation technique used in this paper includes the following issues: Dictionary based; Maximum Matching; and from left to right. Segmented texts are matched against a dictionary prior to being indexed. The Chinese word segmentation source code is provided by the Net-compass research group. This step can control average size of patterns to be retrieved. The average size of patterns is a factor that controls the performance of the rule set.

4.2. Pattern selection

We employ some pattern selection methods to select good patterns for subject rules and for body rules, separately, from the first step. Yang (1997) discussed traditional methods for pattern selection. The problem we are dealing with here is a bit different from the traditional categorization problem, that is we use only the spam-liked

patterns to detect spam while the traditional categorization problem use both spam-liked and ham-liked patterns to categorize spam and ham. Therefore, the formulas of pattern selection should be modified. For a pattern t, we compute a value V_{ts} and V_{th}, which can best evaluate the connection between t and spam, t and ham, respectively. Then, top N patterns that have the highest value of ratio $R_t = V_{ts}/V_{th}$ are selected. N is the size of the rule set, which is a factor that control the performance of the rule set. Given E is a hypothesis that a message occurs as spam and H is a hypothesis that a message contains pattern t, the formulas to compute V_{ts} and V_{th} are follows.

- Document Frequency (DF)

$$V_{ts} = P(H \mid E) = \frac{P(E \wedge H)}{P(E)} \quad (1)$$

$$V_{th} = P(H \mid \overline{E}) = \frac{P(\overline{E} \wedge H)}{P(\overline{E})} \quad (2)$$

- Conditional Probabilities and Bayes's Theorem (CP)

$$V_{ts} = P(E \mid H) = \frac{P(E \wedge H)}{P(H)} \quad (3)$$

$$V_{th} = P(\overline{E} \mid H) = \frac{P(\overline{E} \wedge H)}{P(H)} \quad (4)$$

- Mutual Information (MI)

$$V_{ts} = \log\left(\frac{P(E \wedge H)}{P(E)P(H)}\right) \quad (5)$$

$$V_{th} = \log\left(\frac{P(\overline{E} \wedge H)}{P(\overline{E})P(H)}\right) \quad (6)$$

- Information Gain (IG)

$$V_{ts} = -P(E)\log P(E) + P(E \wedge H)\log\left(\frac{P(E \wedge H)}{P(H)}\right) \quad (7)$$

$$V_{th} = -P(\overline{E})\log P(\overline{E}) + P(\overline{E} \wedge H)\log\left(\frac{P(\overline{E} \wedge H)}{P(H)}\right) \quad (8)$$

- Kullback-Leibler divergence (KL)

$$V_{ts} = \frac{P(E \wedge H)}{P(H)}\log\left(\frac{P(E \wedge H)}{P(E)P(H)}\right) \quad (9)$$

$$V_{th} = \frac{P(\overline{E} \wedge H)}{P(H)}\log\left(\frac{P(\overline{E} \wedge H)}{P(\overline{E})P(H)}\right) \quad (10)$$

Given a spam and ham datasets, for a pattern t, A and B are the number of times spam and ham message contain t, respectively; C and D are the number of times spam and ham message do not contain t, respectively. The values of the probabilities in (1) to (10) are computed as follows.

$$P(E) = \frac{A+C}{A+B+C+D} \quad (11)$$

$$P(\overline{E}) = \frac{B+D}{A+B+C+D} \quad (12)$$

$$P(H) = \frac{A+B}{A+B+C+D} \quad (13)$$

$$P(E \wedge H) = \frac{A}{A+B+C+D} \quad (14)$$

$$P(\overline{E} \wedge H) = \frac{B}{A+B+C+D} \quad (15)$$

Yang (1997) also introduced χ^2 statistical method, however, this method can not be modified to solve our "one-class" detection problem. Using which pattern selection method is a factor that control the performance of the rule set.

4.3 Score assignment

We follow the format of SpamAssassin's rule (Schwartz

(2004)) to create rules from the selected set of spam-liked patterns. The rule set contains two types of rule: subject rule and body rule. We use the Fast SpamAssassin Score Learning Tool provided by Henry Stern (available at http://spamassassin.apache.org) to assign scores for our new rules. This program implements a "Stochastic Gradient Descent" method of training a neural network. It uses a linear transfer function (16) and a logsig activation function (17) to map the weights to SpamAssassin score space.

$$f(x)=\sum_{i=1}^{N} w_i x_i \quad (16)$$

$$y(x)=\frac{1}{1+e^{-f(x)}} \quad (17)$$

Where w_i is the score for rule i and x_i is whether or not rule i is activated by a given message, the transfer function will return the score of the message. The gradient descent method is employed to train the neural network. It involves iteratively tuning the parameters of the network so that the mean error rate always decreases. Without getting into calculus, the error gradient for a perceptron with a linear transfer function, logsig activation function and mean squared error function is:

$$E(x)=y(x)(1-y(x))(y_{\exp}-y(x)) \quad (18)$$

and the weights are updated using the function:

$$w_i = w_i + \alpha E(x) x_i \quad (19)$$

where α is a learning rate. Since the SpamAssassin rule hits are sparse, the training set is randomly walked through, doing incremental updates instead of doing one batch update per epoch. This is so-called "Stochastic gradient descent" method.

5. Experiment

We use a dataset in the spam/ham database from 2005 Jan 1 to 2005 Jun 30, including 194,088 spam and 305,140 ham messages. The dataset is randomly divided into two equal size set: training and testing set. We use the training set to build the Chinese_rules.cf and use the testing test to evaluate the performance results. As proposed in section 4, three factors that affect the performance results of Chinese_rules.cf are pattern selection, rule number N and pattern average size. Experiment results with different values for each factor are follows.

5.1 Pattern selection factor

Fig. 2 shows the spam recall (at ham error = 0.05%) for the rule number and pattern size (bytes) pair are fixed to (500, 4), (300, 4) and (500, 6). (Note that each Chinese character is encoded by 2 bytes.) We can see that DF, CP and MI methods give almost the same performance. CP seems to give the best performance while KL give significantly worsen performance than others.

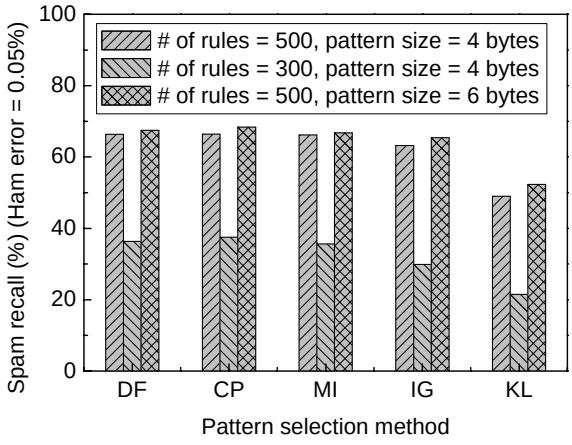

Fig. 2, Performances with different pattern selection methods

5.2 Rule number factor

Now, we fix the pattern selection method to CP, performance with different size of rule set are shown in Fig. 3. We can see that the performance curves (pattern size = 4 and 6) increase with size from 100 to 500, but they seem to be stop increasing when the size is greater than 500. As we know the smaller size of rule set is, the faster the detector. Therefore, we get size is equal to 500 is the best choice.

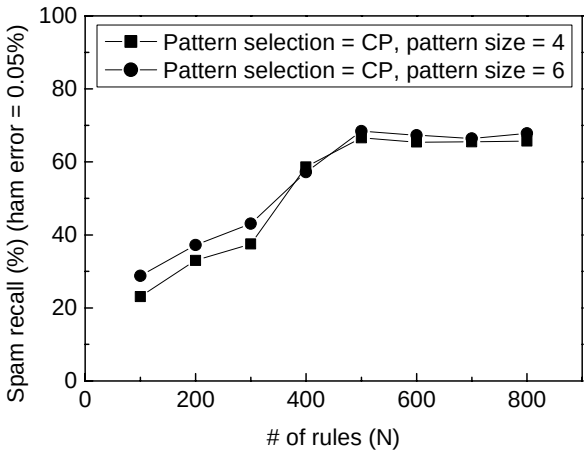

Fig. 3, Performances with different size of rule set

5.3 Pattern average size

Again, we fix the pattern selection method to CP, performance with different average size of rule set are shown in Fig. 4. Two curves with size of rule set = 500 and 800 all show that the best choice for average size of patterns is 6. When the average size of pattern is greater than 8, the curves drop down significantly.

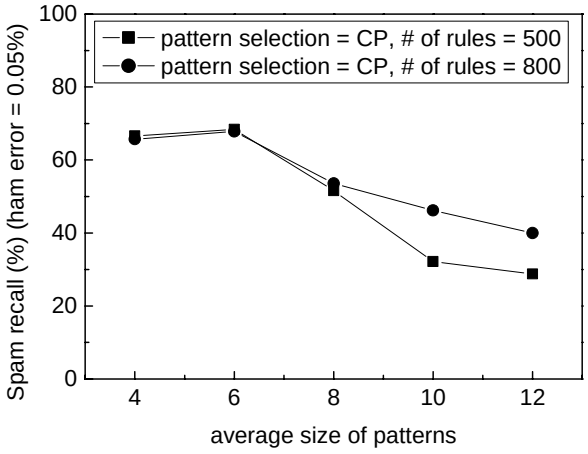

Fig. 4. Performances with different average size of patterns

6. Chinese_rules.cf in progress

As the results from experiments above, we generate the Chinese_rules.cf by the factors: pattern selection method is CP (conditional probabilities and Bayes's Theorem). The Chinese_rules.cf is updated one a week. Table 2 shows the performance of Chinese_rules.cf (updated 2005 Jul 19) for different thresholds.

Table 2. Performance of Chinese_rules.cf (updated 2006 Feb 13)

Threshold	Spam recall (150000)	Ham error (200000)
0.5	93.9%	4.5%
1.0	90.1%	1.9%
1.5	86.0%	0.9%
2.0	81.9%	0.4%
2.5	77.5%	0.1%
3.0	73.0%	0.1%
3.5	69.0%	0.0%
4.0	63.9%	0.0%
4.5	59.4%	0.0%

The rule set takes 0.0454 seconds to scan an email with size 4kb (Pentium 2.8G CPU). The performance is computed when using only the Chinese_rules.cf and the dataset is only Chinese email. In practice, Chinese_rules.cf is always put together with other default rules of SpamAssassin. Some default rules in SpamAssassin that describe the behavior of sending email may also match the Chinese spam, so that the performance in practice may even be better.

Fig. 5 shows the number of unix/linux servers that uses the Chinese_rules.cf per month. Old user means the IP address of that server has been ever appeared in the last months. The increasing number of the Chinese_rules.cf's user confirms the recognition of the Chinese_rules.cf as well as the statistical rule-based method.

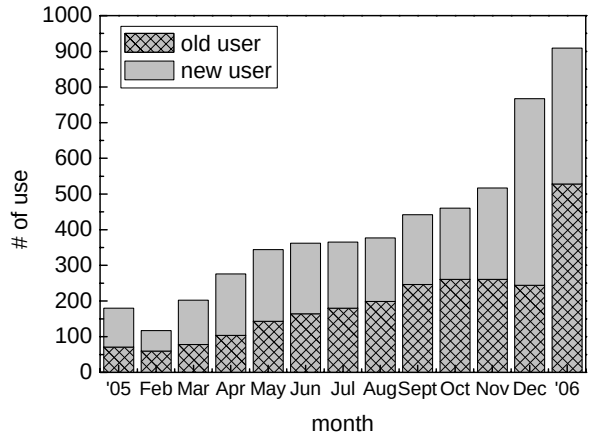

Fig 5, Number of unix/linux email server that use the Chinese_rules.cf

7. Conclusion

We proposed a statistical rule-based method to generate real-time rule set (Chinese_rules.cf) to catch Chinese spam. The rule set can be popularized quickly by sharing with others and is able to keep them up with the changes of spam, i.e. it has good space and time characteristics. In my opinion, this method is not only for spam written in Chinese, but also, for other languages.

For the case of Chinese_rules.cf, we have an experience that using the conditional probabilities (Bayes's Theorem), the number of rules is 500 and the average size of patterns is 6 bytes can generate best performance. Using a common PC, the Chinese_rules.cf takes 0.04 second to scan an email with average size of 5.0 K (attachment not counted). In other words, it can meet the need for filtering spam of an email server processing 2.16 millions emails per day. In practice, Chinese_rules.cf is always put together with other default rules of SpamAssassin, so it poses a problem of how the Chinese rules impact the total scores? This problem will be focused in our future works.

Acknowledgment

This research was supported by the National Natural Science Foundation of China under agreement numbers 60203004.

References

Androutsopoulos I., Koutsias, J., Chandrinos, K.V., Paliouras, G., Spyropoulos, C.D., 2000. An evaluation of Naive Bayesian anti-Spam filtering. Proceedings of the Workshop on Machine Learning in the New Information Age, 11th European Conference on Machine Learning, Barcelona, Spain, pp 9–17.

Drucker, H., Wu, D., Vapnik V., 1999. Support Vector Machines for spam categorization. IEEE Transaction on Neural Networks.10(5).1048-1054.

Foo, S., Li, H., 2004. Chinese word segmentation and its effect on information retrieval. Information Processing & Management. 40(1). 161-190.

Geer, D., 2004. Will new standards help curb spam? IEEE Computer. 2004(2), 14-16.

Goth, G., 2005. Phishing attacks rising, but dollar losses down. IEEE Security and Privacy. 3(1), 8.

Graham, P., 2002. A plan for spam. Web document, URL: http://www.paulgraham.com/spam.html.

Özgür, L., Güngör, T., Gürgen, F., 2004. Adaptive anti-spam filtering for agglutinative languages: a special case for Turkish. Pattern Recognition Letters. 25. 1819-1831.

Schwartz A., 2004. SpamAssassin, O'Reilly.

Wu, Z., Tseng, G., 1993. Chinese text segmentation for text retrieval: achievements and problems. Journal of the American Society for Information Science. 44(9). 532-542.

Yang, Y., Pedersen, J.O., 1997. A comparative study on feature selection in text categorization. Proceedings of the 14th International Conference on Machine Learning, pp 412–420.