2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology

A Personalized Spam Filtering Approach Utilizing Two Separately Trained Filters

Wei-Lun Teng, Wei-Chung Teng
Department of Computer Science and Information Engineering
National Taiwan University of Science and Technology
Taipei, Taiwan
{M9615066, weichung}@mail.ntust.edu.tw

Abstract

By feeding personal e-mails into the training set, personalized content-based spam filters are believed to classify e-mails in higher accuracy. However, filters trained by both spam mails and personal mails may have difficulty classifying e-mails with the same characteristics of both spam and ham. In this paper, we propose a two-tier approach of using two filters trained only with either personal mails or spam mails. E-mails classified as legitimate mails by the legitimate mail filter may pass, while the remaining e-mails are processed by the spam filter in an ordinary way. Experiments in this paper are performed on two mail servers–one equipped with ordinary spam filter, and the other equipped both the legitimate mail filter and the spam filter. By combining the two filters with tuned thresholds, a much lower false positive rate is observed under the same false negative rate comparing to the ordinary filter.

Keywords—personalized spam filtering, content-based, two-tier

1. Introduction

While e-mail remains to be one of the most popular applications on the Internet, the threat posed by unsolicited bulk e-mail, also known as spam, still put every end user at risks. Some earlier studies show that spam has occupied around 80% of the incoming mails [1], and a more recent statistics from spam filter vendor even shows an increase of spam at over 90%. Thanks to the rapid developments of spam filtering and other anti-spam technologies, the end users do not need to face that many spam mails everyday when they open their mail boxes; that is, without proper spam filters, spam will become an onerous burden rather than an annoying trouble.

Spam filters use at least one feature of a spam to identify every passing e-mail, and to block out those justified as spam. E-mail service providers (ESP) and relay mail servers may proceed with the filtering by collecting information such as the number of times the same mail has been sent, or to check with the blacklist of each domain; but that is not enough. To see from the user's point of view, personal filters have the advantage to access white list and other user data to further verify the legitimacy of an e-mail. Currently there are many existing spam filtering techniques. We can classify these techniques into two major types: the rule-based techniques and the content-based techniques.

Rule-based filters require a set of weighted rules manually created and maintained, where every rule maps to one specific feature of an e-mail. When filtering an incoming e-mail, the filter sequentially go through every rule. Whenever a rule is triggered, the weight of this rule is added to the score of the incoming mail. If the score of an e-mail exceeds a predefined threshold, this e-mail is identified as spam. Rule-based approaches, when well tuned, show very accurate filtering result. Popular spam filter such as SpamAssassin [2] implemented a classic ruled-based filter from its early version, and it shows consistently good performance. However, as most spammers keep developing new tricks to get away from existing spam filters, users need to continuously keep their rules fine tuned, yet most end users don't have enough skill and motivation to maintain their rule set [3].

Content-based approaches, on the contrary, allow the filters automatically list out the features of spam or even legitimate mails from the collected samples. This type of approach usually adopts machine learning skills to help reasoning out the features of spam, and therefore it is also called the learning-based approaches. Currently a large amount of researches have been done to increase the accuracy of content-based spam filter. Early researches use Naive Bayesian algorithm [4-6],

978-0-7695-3496-1/08 $25.00 © 2008 IEEE
DOI 10.1109/WIIAT.2008.257

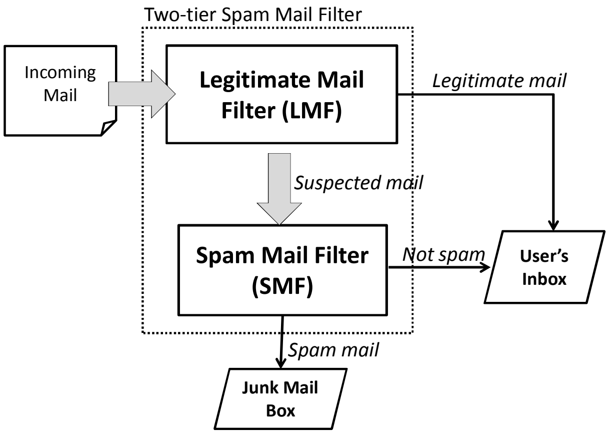

Figure 1. The structure of two-tier mail filter.

and later Support Vector Machine (SVM) [8] [9] and Nearest Neighbor (NN) classifiers became popular on mail filtering tasks. Compared with rule-based approach, content-based approach alleviates the burden of human by analyzing and maintaining the rule set automatically; that is, to continuously learn from the incoming mails to maintain an "implicit rule set". When implemented in user site, content-based approach may have more accurate and personalize filtering result by training through both spam and personal legitimate mails [9] [10]. In fact, without training legitimate mails from user's inbox, the performance of a spam filter may decrease drastically [11]. However, machine learning skills also require constant user's feedbacks to prevent misclassifications.

Theoretically, a user gives feedback by labeling misclassified mails. With well designed user interface, labeling a mail may be as easy as deleting one. However, users need to go through all filtered mails to check if there is any misclassified one. Therefore getting rid of this time consuming task is exactly what we need spam filters for!

Since almost all spam filters are implemented as a binary classifier over one threshold, a practical solution to this problem is to increase the threshold such that misclassified legitimate mail doesn't exist, i.e. the false positive (FP) rate approaches zero. The drawback of this method is obviously: the number of passed spam will inevitably increase if we loosen the restrictions of filter by increasing the threshold. In fact, this is the dilemma of all spam filters: if we raise the threshold, the false negative (FN) rate will increase. On the contrary, decreasing the threshold will improve the sensitivity of a spam filter, and more spam are detected and blocked, but the result will be a higher FP rate.

To resolve this problem, we propose a two-tier filtering structure to adjust the distribution of FP rate and FN rate such that low FP rate may be reached without drastic increase of FN rate.

2. The Two-tier Spam Filter Structure

Current spam filter systems usually apply the most popular approach to employ multiple filters and to calculate the total score as a weighted sum [2] [11] [12]. This kind of systems usually shows high accuracy and comfortable precision. However, from the end user's point of view, the labeling effort can be reduced only when the FN rate becomes low enough to be ignored. Therefore, the proposed two-tier filter structure gives emphasis on eliminating misclassified legitimate mail rather than the influence of accuracy and FP rate. Figure 1 shows the structure and the filtering process of the two-tier spam filter.

The first stage of spam filtering is a filter called legitimate mail filter (LMF). The only purpose of LMF is to report a score which indicates the possibility of a legitimate incoming mail message. LMF is trained only by the legitimate mails inside the user's inbox. One of the reasons that personal mails preferred to be separately trained is that the features of personal mails, comparing to spam, tends to keep consistent. On the contrary, spammers always try new skills to avoid spam filter, so the features of spam may vary from time to time. Therefore, the LMF alone is expected to generate stable output with high accuracy after short learning period. Another reason is that every user has different definition to legitimate mails. For example, some advertisement mails regarded as spam by some users might be considered as useful emails by the others. When a spam-like mail has the information user may be interested in, ordinary spam filters have a bigger chance to block this mail, whereas the LMF in our two-tier structure will let this mail pass through.

The second stage is an ordinary spam mail filter (SMF) trained without personal mails. The purpose of SMF is the same with any other spam filters: to classify the incoming mail as spam or legitimate mails. In Figure 1, the LMF looks like in cooperation with SMF; however, LMF is designed to work independently. To filtering an incoming mail, LMF first decide the similarity of the incoming mail with a legitimate mail database, and then place the mail into the user's inbox if it is classified as legitimate mail. Otherwise, the incoming mail will be regarded as suspect mail and being filtered by SMF. That is, SMF will examine the suspect mail to decide whether it is a spam or not.

In the two-tier filtering structure, e-mails are classified as legitimate mails by LMF if the mail contains the information user may be interested in, and regard the remaining mails as suspect mails which will be processed by SMF. We believe using this novel structure not only can compensate the dilemma of conventional spam filter, but also can filter mails according to different user's preferences.

3. The Legitimate Mail Filter

We designed a LMF to classify incoming mails at the first stage. LMF will place the mail into user's inbox if the mail is determined to be legitimate mails.

Keywords contained in the header or contents of the e-mail can be regarded as the mail's feature. For keywords picking, we use TF-IDF (Term Frequency–Inverse Document Frequency) technique [15] [16] to get the keywords of legitimate mails. Within these keywords, we use cosine similarity theorem [18] [19] to compute a similarity score between the incoming mail and legitimate in user's inbox. The incoming mail will be determined as legitimate mail if the mail's score is higher than the threshold we set. The following paragraphs 3.1 and 3.2 explain how LMF is trained and how it classifies mails.

3.1. Training Legitimate Mail Filter

Before filtering new incoming mails, we must train the LMF first. Figure 2 shows the flowchart of LMF training. The first stage called mail parsing stage, which is to retrieve the text content by parsing MIME format. By using CPAN library MIME::Parser [13], we can decode mails of MIME format and get information from these mails such as mail title, body text and the attached files, etc. In this step, the system not only decodes the mails, but also gets the mail addresses of the senders to build up the white-list database.

In order to allow LMF to support Chinese email contents, we must use the Chinese word segmentation module to analyze the content of the mail. Unlike English, Chinese doesn't use spaces to segment each word. Therefore, we must include a preprocessor for segmentation. In the second stage, the content processing stage, an open source Chinese word segmentation module MMSEG [14] is used to implement the preprocessor and to get the words of each mail. After the Chinese words are retrieved, we must select the suitable words as the keywords to build up a database of legitimate mails. The retrieved words are then collected and added into a database in the third stage.

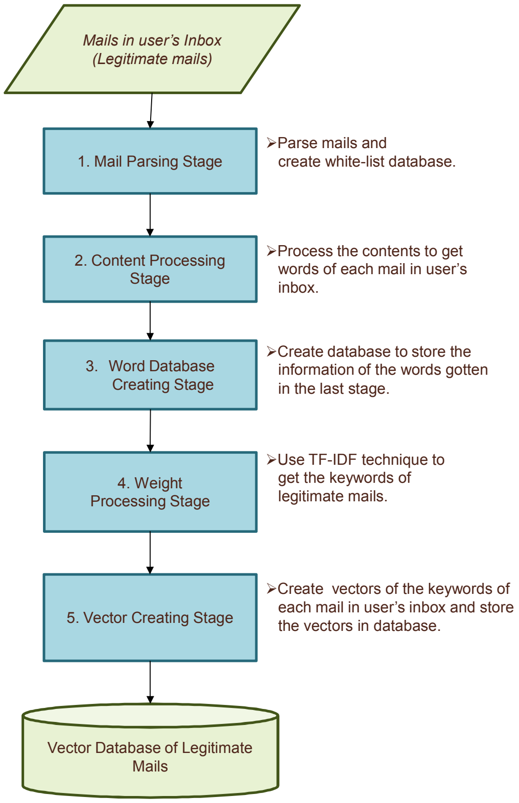
Mails in user's Inbox
(Legitimate mails)
1. Mail Parsing Stage
➢Parse mails and create white-list database.
2. Content Processing Stage
➢Process the contents to get words of each mail in user's inbox.
3. Word Database Creating Stage
➢Create database to store the information of the words gotten in the last stage.
4. Weight Processing Stage
➢Use TF-IDF technique to get the keywords of legitimate mails.
5. Vector Creating Stage
➢Create vectors of the keywords of each mail in user's inbox and store the vectors in database.
Vector Database of Legitimate Mails

Figure 2. Flowchart of training Legitimate Mail Filter.

TF-IDF is a common technique and is often used in information retrieval and text mining researches. In the fourth stage, we use TF-IDF to get the keywords from mails. Equation 1 shows the weight function of TF-IDF.

$$Weight_{ij} = TF_j * IDF_j = \frac{n_j}{n_{all}} * \log_2 \frac{|d|}{df_j} \qquad (1)$$

where n_j indicates the frequency of word j in the i^{th} mail, and n_{all} means the number of all words in the i^{th} mail. $|d|$ is the number of all mails and df_j is the number of word j which appears in the mails. By multiplying TF_j and IDF_j, we get the weight of word j in mail i.

However, there is a trend in Chinese language that longer words are usually more significant than the shorter ones, i.e. a word with four characters may have a more complete meaning comparing to a word with two characters. This is a feature that does not appear in English. Therefore, we take this factor into consideration when applying the function of TF-IDF. Equation 2 shows the improved TF-IDF function.

$$Weight_{ij} = (Length(j))^2 * \frac{n_j}{n_{all}} * \log_2 \frac{|d|}{df_j} \qquad (2)$$

In order to show that the length of the words will influence the significance of word, we allow the original TF-IDF formula to multiply by a value, and we define this value as the square of the word's length; i.e., $(length(j))^2$. In this way, longer words receive higher weights.

Cosine similarity [18] [19] is another technique often used to compare documents in text mining. It is a measure of similarity between two vectors of n dimensions by finding the angle between them, and we apply cosine similarity technique in the fifth stage. By using the selected words, we create a vector space and map each mail into a vector in this space. The number of the vector space dimension is very important. If the number of dimension is too large, the mapped vector may contain too many words which are unimportant, and it will also waste computer memory space and computing time. On the other hand, if the dimension is too small, it will be hard to completely represent the characteristics of a mail by its vector. Thus, the number of dimension must be flexible according to the number of mails in user's inbox. Equation 3 shows how to define the dimension of the vector space.

$$Max\_Dimension = \begin{cases} 100, & if\ |d| \leq 10; \\ |d| * \left\lfloor \frac{10}{\log|d|} \right\rfloor, & if\ |d| > 10; \end{cases} \qquad (3)$$

where $|d|$ is the number of mails in user's inbox, and *Max_Dimension* is the dimension of the vector space we finally decide.

Let $k=Max\_Dimension$ and choose k words with higher weight, and then mail i can be expressed in a vector v_i, where $v_i = (W_{i1}, W_{i2}, \ldots W_{ij}, \ldots W_{ik})$, and W_{ij} is the j^{th} word in the i^{th} mail. All mails in user's inbox will be transformed to vector format to build the legitimate database.

3.2. Filtering Legitimate Mails

Figure 3 shows the flowchart of filtering new incoming mail in LMF. To filter a new incoming mail, LMF at first transform the mail into a vector using the same method as described above. After then, we use the cosine similarity technique and white-list database to operate in coordination and compute the total score, which indicates the possibility of this incoming mail being legitimate. Equation 4 shows how we calculate the score of every incoming mail.

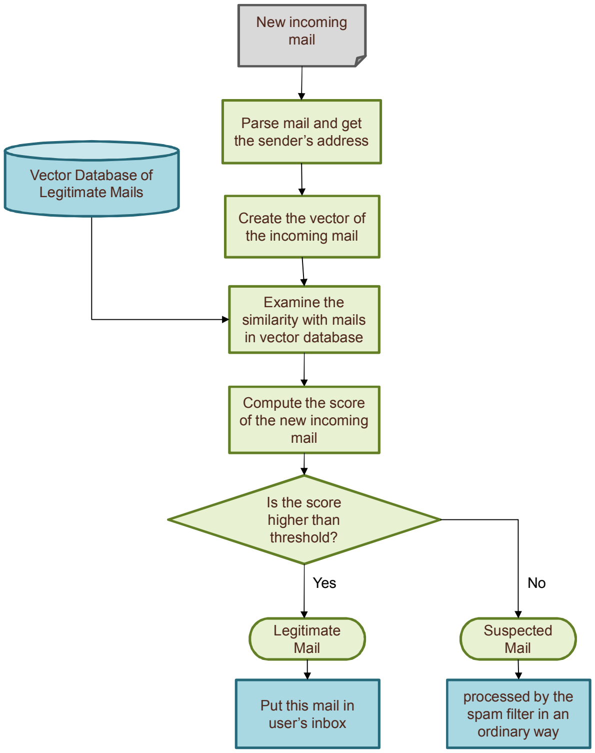
New incoming mail
Vector Database of Legitimate Mails
Parse mail and get the sender's address
Create the vector of the incoming mail
Examine the similarity with mails in vector database
Compute the score of the new incoming mail
Is the score higher than threshold?
Yes
No
Legitimate Mail
Suspected Mail
Put this mail in user's inbox
processed by the spam filter in an ordinary way

Figure 3. Flowchart of Legitimate Mail Filter on filtering new incoming mails.

$$Score = MAX(cs(u, v_i)) * 10 + WhiteListNum * \left(\frac{T}{3}\right) \qquad (4)$$

where u is the vector of new incoming mail, and v_i is the vector of the i^{th} mail we stores in legitimate mail database we created in training stage. Using cosine similarity technique, we can examine the similarity of vector u and vector v_i. $MAX(cs(u,v_i))$ means the maximum similarity between the new incoming mail and all legitimate mails in user's inbox. *WhiteListNum* is the number of times sender of this new incoming mail appears in white-list database, and T is the threshold we set. Equation 5 shows the formula of cosine similarity:

$$cs(u, v_i) = \frac{u \cdot v_i}{|u||v_i|} \qquad (5)$$

To normalize $u{\cdot}v_i$, we divides it by the Euclidean distance between u and v_i; i.e., $u{\cdot}v_i/(|u||v_i|)$. This ratio defines the cosine angle between the vectors, with values between 0 and 1. We use cosine similarity formula to compute the similarity numbers between new incoming mail and each mail in user's inbox.

New incoming mail will be given its own score by LMF, the score will decide whether the mail belongs to legitimate mail or not. If the mail's score is equal or greater than the threshold we set, the mail will be regarded as legitimate mail. Otherwise, the mail will belong to suspected mail if the score is smaller than the threshold.

4. Experiments

4.1. Experimental Setup

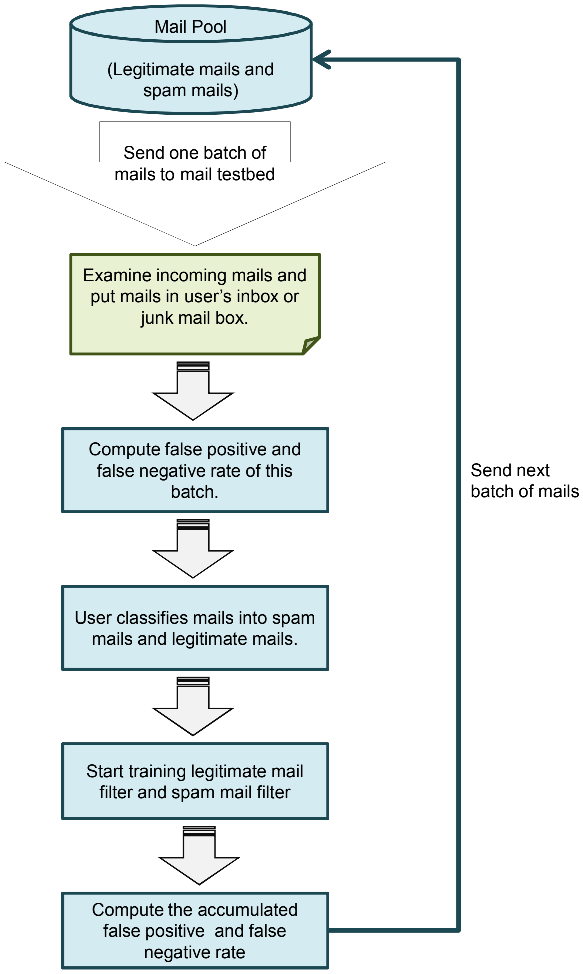
Mail Pool
(Legitimate mails and spam mails)
Send one batch of mails to mail testbed
Examine incoming mails and put mails in user's inbox or junk mail box.
Compute false positive and false negative rate of this batch.
User classifies mails into spam mails and legitimate mails.
Start training legitimate mail filter and spam mail filter
Compute the accumulated false positive and false negative rate
Send next batch of mails

Figure 4. Flowchart of the experimental testbed.

To evaluate the proposed two-tier filter, the experiments in this research are performed on two mail servers. One mail server runs ordinary spam filter, and the other one runs both the legitimate mail filter and the spam filter. Both mail servers use Postfix [17] as the mail transfer agent (MTA). Postfix is a mailer that started as an IBM research project, and is a promising alternative to the widely-used Sendmail program. Postfix aims to be fast, easy to administer, and secure. For SMF, we use SpamAssassin [2] and install it on both mail servers. SpamAssassin has both rule-based and Bayesian filters implemented, and it has good spam classification performance. Currently SpamAssassin is one of the most popular spam filtering agents nowadays.

We have gathered about 3000 e-mails for our experiments, including 1000 legitimate mails and 2000 spam mails, all from the same person. Both legitimate and spam mails are separated into 40 batches, with 25 legitimate mails and 50 spam mails in each batch. By this we simulate the user's mail receiving pattern for 40 days. When one batch of mail is sent, the spam filter does the classification automatically and organizes the mails into user's inbox or junk mailbox respectively. The user or the source provider reviews all the newly arrived mails and label out those misclassified ones. Finally, the system feedback all newly arrived mails to LMF and SMF for further training. Figure 4 shows the flowchart of the experimental testbed.

4.2. Results

Through the experiments above, we observed that filtering results on both mail servers are under different thresholds. We use higher threshold on LMF, which means that the incoming mails will be placed into user's inbox only if the mail has high similarity with trained legitimate mails. On the other hand, if the incoming mail with lower similarity than the threshold we set, the mail will be processed by SMF. Table 1 and table 2 show that the FP and FN rates of two mail servers under different SMF thresholds coupled with fixed LMF thresholds (the values 8 and 9 are evaluated). Figure 5 shows the line chart of filtering result of two-tier spam filter (LMF threshold = 8) and traditional spam filter. Comparing with ordinary filters, we found that the two-tier spam filter has a much lower FP rate under the same FN rate. The two-tier spam filter successfully lessen the misjudgment rate of legitimate mails by increasing only a little of FN rate. Among the spam mails misclassified by LMF, 80% of mails contained the information which user may be interested in. So the LMF filter mails in a way that users desire. Through improving the structure of traditional spam filter, two-tier spam filter successfully resolve the problem of conventional spam filter, lower the false positive rate and ease the painful user filtering task.

TABLE 1. False negative rate under three different system configurations

Thresholds of SpamAssassin	1	2	3	4	5	6	7	8	9
SpamAssassin only	0.87%	1.18%	2.43%	3.68%	4.56%	7.06%	8.56%	11.75%	13.87%
SpamAssassin with LMF, threshold = 8	1.7%	2.05%	3.4%	4.65%	5.15%	7.35%	9.45%	12.85%	14.6%
SpamAssassin with LMF, threshold = 9	1.0%	1.35%	2.75%	3.75%	4.55%	6.9%	8.9%	12.05%	14.0%

TABLE 2. False positive rate under three different system configurations

Thresholds of SpamAssassin	1	2	3	4	5	6	7	8	9
SpamAssassin only	68.12%	45.70%	30.37%	13.37%	8.25%	6.12%	4%	3.75%	1.87%
SpamAssassin with LMF, threshold = 8	12.5%	9%	5.8%	3.5%	2.9%	2.4%	1.6%	1.4%	1%
SpamAssassin with LMF, threshold = 9	13.1%	9%	6.0%	3.6%	2.9%	2.4%	1.6%	1.4%	1%

5. Conclusion

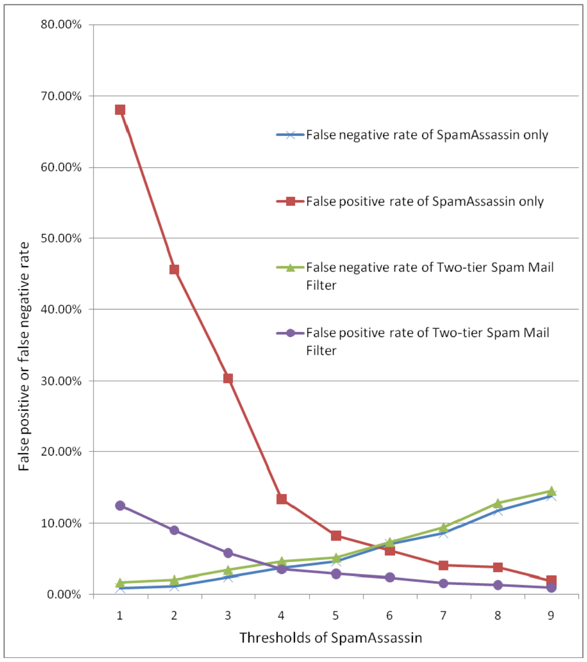

Figure 5. The Impact of the two-tier structure on FP rate and FN rate.

In this paper we argue that personalized content-based spam filters should be divided into two parts: a legitimate mail filter and a spam mail filter. The proposed two-tier filtering structure puts more weight on reducing the number of misclassified legitimate mails, and the two thresholds of this structure brings higher flexibility on adjusting the values of FP rate and FN rate couple. To implement a legitimate mail filter for Chinese mail, we modified TF-IDF algorithm and implemented an SVM-based filter. Combined LMF and SpamAssasin, we performed experiments to measure the distributions of FP rate and FN rate. Early results show that compared with ordinary spam filter, our two-tier structure reaches much lower FN rate with slight increase to FP rate observed.

The future research will include more experiments on various combinations of the threshold couples, and to derive the equations for threshold mapping.

6. Acknowledgment

This work is supported in part by the Taiwan Information Security Center (TWISC) under the National Science Council Grants No. NSC96-2219-E-011-008.

7. References

[1] Messaging Anti-Abuse Working Group, MAAWG Email Metrics Program, *First Quarter 2006 Report. June 2006*. Available: http://www.maawg.org/about/FINAL_1Q2006_Metrics_Report.pdf.

[2] The Apache SpamAssassin Project [Online]. Available: http://spamassassin.apache.org/

[3] J. Clark, I. Koprinska and J. Poon, "LINGER - A Smart Personal Assistant for E-mail Classification," in *Proc. of the 13th Int. Conf. on Artificial Neural Networks (ICANN'03)*, 2003, pp. 274-277.

[4] M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz, "A Bayesian Approach to Filtering Junk E-mail," *AAAI*

Workshop on Learning for Text Categorization, 1998, AAAI Technical Report WS-98-05.

[5] P. Graham, "Better Bayesian Filtering," in *Proc. of MIT Spam Conference 2003*. Available: http://www.paulgraham.com/better.html

[6] A.K. Seewald, "An Evaluation of Naive Bayes Variants in Content-based Learning for Spam Filtering," *Journal of Intelligent Data Analysis*, 2007, vol. 11, no. 5, pp. 497-524.

[7] H. Drucker, D. Wu, and V.N. Vapnik, "Support Vector Machine for Spam Categorization," *IEEE Trans. on Neural Networks*, 1999, vol. 10, pp. 1048–1054.

[8] A. Kolcz and J. Alspector, "SVM-based Filtering of E-mail Spam with Content-specific Misclassification Costs", in *Proc. of the TextDM Workshop on Text Mining*, 2001.

[9] K.N. Junejo and A. Karim, "PSSF: A Novel Statistical Approach for Personalized Service-side Spam Filtering," in *Proc. of the 2006 IEEE/WIC/ACM Int. Conf. on Web Intelligence*, 2007, pp. 228 – 234.

[10] K.N. Junejo, M.M. Yousaf, and A. Karim. "A Two-pass Statistical Approach for Automatic Personalized Spam Filtering," *Proc. of ECML/PKDD Discovery Challenge Workshop*, 2006, pp. 16-27.

[11] V. Cheng and C.H. Li, "Personalized Spam Filtering with Semi-supervised Classifier Ensemble," in *Proc. of the 2006 IEEE/WIC/ACM Int. Conf. on Web Intelligence,* 2006, pp. 195-201.

[12] L. Pelletetier, J. Almhana, and V. Choulakian, "Adaptive Filtering of SPAM," in *Proc. of the 2nd Annual Conf. on Communication Networks and Service Research (CNSR'04)*, 2004, pp. 218-224.

[13] The CPAN Search Site, *MIME::Parser - experimental class for parsing MIME streams* [Online]. Available: http://search.cpan.org/dist/MIME-tools/lib/MIME/Parser.pm

[14] C. H. Tsai, "MMSEG: A Word Identification System for Mandarin Chinese Text Based on Two Variants of the Maximum Matching Algorithm" [Online]. Available: http://technology.chtsai.org/mmseg/

[15] G. Salton and C. Buckley, "Term-weighting Approaches in Automatic Text Retrieval," *Information Processing and Management: an International Journal*, 1988, vol. 24, issue 5, pp. 513–523.

[16] K.S. Jones, "A Statistical Interpretation of Term Specificity and Its Application in Retrieval," *Journal of Documentation*, 1972, vol. 28, issue 1, pp. 11–21.

[17] The Postfix Project [Online]. Available: http://www.postfix.org

[18] Cosine Similarity and Term Weight Tutorial [Online]. Avalable:http://www.miislita.com/information-retrieval-tutorial/cosine-similarity-tutorial.html

[19] G. Salton, A. Wong, and C. S. Yang, "A Vector Space Model for Automatic Indexing," *Communications of the ACM*, 1975, vol.18, pp. 613–620.