Available online at www.sciencedirect.com

ScienceDirect

Knowledge-Based Systems 20 (2007) 249–254

Knowledge-Based SYSTEMS

www.elsevier.com/locate/knosys

An empirical study of three machine learning methods for spam filtering

Chih-Chin Lai *

Department of Computer Science and Information Engineering, National University of Tainan, Taiwan 700, Taiwan

Received 10 July 2005; accepted 1 May 2006
Available online 5 September 2006

Abstract

The increasing volumes of unsolicited bulk e-mail (also known as spam) are bringing more annoyance for most Internet users. Using a classifier based on a specific machine-learning technique to automatically filter out spam e-mail has drawn many researchers' attention. This paper is a comparative study the performance of three commonly used machine learning methods in spam filtering. On the other hand, we try to integrate two spam filtering methods to obtain better performance. A set of systematic experiments has been conducted with these methods which are applied to different parts of an e-mail. Experiments show that using the header only can achieve satisfactory performance, and the idea of integrating disparate methods is a promising way to fight spam.

Keywords: Spam filtering; Machine learning

1. Introduction

In recent years, e-mails have become a common and important medium of communication for most Internet users. However, spam, also known as unsolicited commercial/bulk e-mail, is a bane of e-mail communication. A study estimated that over 70% of today's business e-mails are spam [1]; therefore, there are many serious problems associated with growing volumes of spam such as filling users' mailboxes, engulfing important personal mail, wasting storage space and communication bandwidth, and consuming users' time to delete all spam mails. Spam mails vary significantly in content and they roughly belong to the following categories: money making scams, fat loss, improve business, sexually explicit, make friends, service provider advertisement, etc., [13]. One example of a spam mail is shown as Fig. 1.

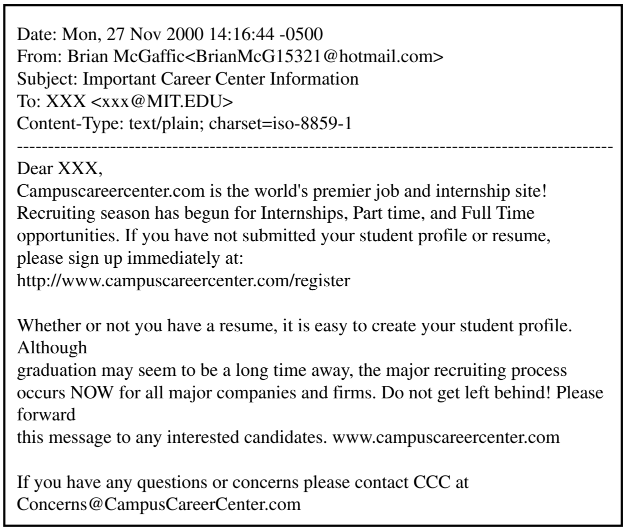
Several solutions have been proposed to overcome the spam problem. Among the proposed methods, much interest has focused on the machine learning techniques in spam filtering. They include rule learning [4,6], Naïve Bayes [2,9], decision trees [5], support vector machines [7,8,16] or combinations of different learners [10]. The common concept of these approaches is that they do not require specifying any rules explicitly to filter out spam mails. Instead, a set of training samples (pre-classified e-mails) is needed. A specific machine-learning technique is then used to "learn" and "produce" the classification model from this data. From the machine learning viewpoint, spam filtering based on the textual content of e-mail can be viewed as a special case of text categorization, with the categories being spam or non-spam [8].

Sahami et al. [9] employed Bayesian classification technique to filter junk e-mails. By making use of the extensible framework of Bayesian modeling, they can not only employ traditional document classification techniques based on the text of e-mail, but they can also easily incorporate domain knowledge to aim at filtering spam e-mails.

Androutsopoulos et al. [2–4] presented a series of papers that extended the Naïve Bayes (*NB*) filter proposed by Sahami et al. [9], by investigating the effect of different number of features and training-set sizes on the filter's performance. Meanwhile, they compared the performance of *NB* to a memory-based approach, and they found both above-mentioned methods clearly outperform a typical keyword-based filter.

* Tel.: +886 62606123; fax: +886 62606125.
E-mail address: cclai@mail.nutn.edu.tw

0950-7051/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.knosys.2006.05.016



Date: Mon, 27 Nov 2000 14:16:44 -0500
From: Brian McGaffic<BrianMcG15321@hotmail.com>
Subject: Important Career Center Information
To: XXX <xxx@MIT.EDU>
Content-Type: text/plain; charset=iso-8859-1

Dear XXX,
Campuscareercenter.com is the world's premier job and internship site!
Recruiting season has begun for Internships, Part time, and Full Time
opportunities. If you have not submitted your student profile or resume,
please sign up immediately at:
http://www.campuscareercenter.com/register

Whether or not you have a resume, it is easy to create your student profile.
Although
graduation may seem to be a long time away, the major recruiting process
occurs NOW for all major companies and firms. Do not get left behind! Please
forward
this message to any interested candidates. www.campuscareercenter.com

If you have any questions or concerns please contact CCC at
Concerns@CampusCareerCenter.com

Fig. 1. An example of a spam mail.

Drucker et al. [7] used support vector machine (*SVM*) for classifying e-mails according to their contents and compared its performance with Ripper, Rocchio, and boosting decision trees. They concluded that boosting trees and *SVM* had acceptable test performance in terms of accuracy and speed. However, the training time of boosting trees is inordinately long. Woitaszek et al. [17] utilized a simple *SVM* and a personalized dictionary to identify commercial electronic mail. The SVM-based mail classification system was implemented as an add-in for *Microsoft Outlook XP*, allowing desktop users to quickly identify unsolicited e-mail.

Although it is a popular topic in machine learning, very few approaches using instance-based nearest neighbor techniques are presented for spam filtering. Trudgian and Yang [14] examined the performance of the *kd*-tree nearest neighbor algorithm for word based spam mail classification and compared it to other common methods.

Several attempts have been made to evaluate the performance of machine-learning methods on spam filtering task; however, these studies focused on features which extracted from message body only. Here we study different parts of an e-mail that can be exploited to improve the categorization capability, by giving experimental comparisons of three respective machine learning algorithms. These techniques are Naïve Bayes (*NB*), *k*-nearest neighbor (*k-NN*), and support vector machines (*SVMs*). We considered the following four combinations of an e-mail message: all (*A*), header (*H*), subject (*S*) and body (*B*). The above-mentioned three methods with these features are compared to help evaluate the relative merits of these algorithms. In addition to using a single method for spam filtering, we adopted an integrated approach which considered two different methods to anti-spam filtering and evaluated its performance.

The rest of this paper is organized as follows. Section 2 gives a brief review of three machine learning algorithms and details of the integrated approach. Section 3 provides the considered features and experimental results designed to evaluate the performance of different experimental settings are presented in Section 4. The conclusions and directions for future works are summarized in Section 5.

2. Machine learning methods and proposed combined apparoch

2.1. Naïve Bayes

The Naïve Bayes (*NB*) classifier is a probability-based approach. The basic concept of it is to find whether an e-mail is spam or not by looking at which words are found in the message and which words are absent from it. This approach begins by studying the content of a large collection of e-mails which have already been classified as spam or legitimate. Then when a new e-mail comes into some user's mailbox, the information gleaned from the "training set" is used to compute the probability that the e-mail is spam or not given the words appearing in the e-mail.

Given a feature vector $\vec{\mathbf{x}} = \{x_1, x_2, \ldots, x_n\}$ of an e-mail, where are the values of attributes $X_1, \ldots, X_n$, and n is the number of attributes in the corpus. Here, each attribute can be viewed as a particular word occurring or not. Let c denote the category to be predicted, i.e., $c \in$ {spam, legitimate}, by Bayes law the probability that $\vec{\mathbf{x}}$ belongs to c is as given in

$$P(c|\vec{\mathbf{x}}) = \frac{P(c) \cdot P(\vec{\mathbf{x}}|c)}{P(\vec{\mathbf{x}})}, \tag{1}$$

where $P(\vec{\mathbf{x}})$ denotes the a-priori probability of a randomly picked e-mail has vector $\vec{\mathbf{x}}$ as its representation, $P(c)$ is also the a prior probability of class c (that is, the probability that a randomly picked e-mail is from that class), and $P(\vec{\mathbf{x}}|c)$ denotes the probability of a randomly picked e-mail with class c has $\vec{\mathbf{x}}$ as its representation. Androutsopoulos et al. [2] notes that the probability $P(\vec{\mathbf{x}}|c)$ is almost impossible to calculate because the fact that the number of possible vectors $\vec{\mathbf{x}}$ is too high. In order to alleviate this problem, it is common to make the assumption that the components of the vector $\vec{\mathbf{x}}$ are independent in the class. Thus, $P(\vec{\mathbf{x}}|c)$ can be decomposed to

$$P(\vec{\mathbf{x}}|c) = \prod_{i=1}^{n} P(x_i|c) \tag{2}$$

So, using the *NB* classifier for spam filtering can be computed as

$$C_{NB} = \arg\ \max_{c \in \{spam, legitmiate\}} P(c) \prod_{i} P(x_i|c) \tag{3}$$

2.2. K-nearest neighbor

The most basic instance-based method is the *k*-nearest neighbor (*k-NN*) algorithm. It is a very simple method to classify documents and to show very good performance on text categorization tasks [16]. The procedure of *k-NN* method which is employed to e-mail classification is as follows: Given a new e-mail, the distance between the mail and all samples in the training set is calculated. The distance used in practically all nearest-neighbor classifiers is the Euclidean distance. With the distance calculated, the samples are ranked according to the distances. Then the *k* samples which are nearest to the new e-mail are used in assigning a classification to the case.

2.3. Support vector machine

Support vector machine (*SVM*) is a new and very popular technique for data classification in the machine learning community. The concepts of behind it are Statistical Learning Theory and Structural Minimization Principle [15]. *SVM* has been shown to be very effective in the field of text categorization because it has the ability to handle high-dimensional data by using kernels.

When using *SVM* for pattern classification, the basic idea is to find the optimal separating hyperplane that gives the maximum margin between the positive and negative samples. According to the idea, the spam filtering can be viewed as the simple possible *SVM* application – classification of linearly separable classes; that is, a new e-mail either belongs or does not to the spam category.

Given a set of training samples $\mathbf{X} = \{(\mathbf{x}_i,\ y_i)\}$, where $\mathbf{x}_i \in \mathrm{R}^m$ and $y_i \in \{+1,\ -1\}$ is the corresponding output for the *i*th training sample (here $+1$ represents spam and -1 stands for legitimate mail). The output of a linear *SVM* is

$$y = \boldsymbol{w} \cdot \boldsymbol{x} - b, \tag{4}$$

where y is the result of classification, $\boldsymbol{w}$ is the normal weight vector corresponding to those in the feature vector $\boldsymbol{x}$, and b is the bias parameter in the *SVM* model that determined by the training process. Maximizing the margin can be achieved through the following optimization problem:

$$\text{minimize } \frac{1}{2}\|\mathbf{w}\|^2, \tag{5}$$

$$\text{subjected to } y_i(\boldsymbol{w} \cdot \boldsymbol{x} + b) \geqslant 1, \forall i. \tag{6}$$

More and more researchers pay attention to *SVM*-based classifier for spam filtering, since their demonstrated robustness and ability to handle large feature spaces makes them particularly attractive for this work.

2.4. The integrated approach

Segal et al. [11] pointed out that no one anti-spam solution is the "right" answer, and that the best approach requires a multi-faceted one that combines various forms of filtering with infrastructure changes, financial changes, legal resource, and more. Based on this idea, here we presented an integrated approach that adopted *TF-IDF* (term frequency-inverse document frequency) method and *SVM*. The former is used to extract the features from an e-mail, and then the latter is applied to this information to produce a prediction whether an incoming e-mail is spam or not.

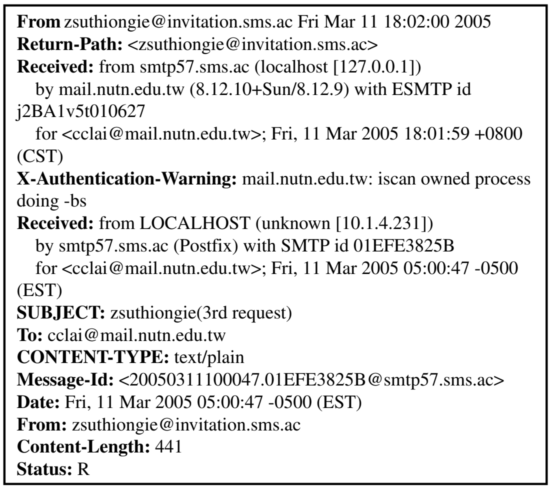
3. The structure of an e-mail

In addition to the body message of an e-mail, an e-mail has another part called the header. The job of the header is to store information about the message and it contains many fields, for example, tracing information about which a message has passed (Received:), authors or persons taking responsibility for the message (From:), intending to show the envelop address of the real sender as opposed to the sender used for replying (Return-Path:), unique of ID of this message (Message-ID:), format of content (Content-Type:), etc. Fig. 2 illustrates an example of the header in an e-mail. On the other hand, many spam messages may contain common text in the subject of the e-mail. A few such subjects that are used to clearly identify spam would include text like: "Get rich fast", "University diploma", "Save money", "Viagra online", "Credit repair", etc [1].

Besides comparing the classification performance among the considered learning algorithms, in this study we intended to figure out which parts of an e-mail have critical influence on the classification results. Therefore, four features of an e-mail: all (*A*), header (*H*), subject (*S*), and body (*B*) are used to evaluate the performance of three machine learning algorithms and the proposed integrated approach. Furthermore, we also considered four cases that whether the preprocessing (stemming or stopping) procedure was applied or not. The purpose of stemming is to lower the size of feature vector. Stopping is employed to remove common words, which are not very useful in classification task.

4. Experimental results

In order to test the performance of above-mentioned three methods, some corpora of spam and legitimate



```
From zsuthiongie@invitation.sms.ac Fri Mar 11 18:02:00 2005
Return-Path: <zsuthiongie@invitation.sms.ac>
Received: from smtp57.sms.ac (localhost [127.0.0.1])
    by mail.nutn.edu.tw (8.12.10+Sun/8.12.9) with ESMTP id
j2BA1v5t010627
    for <cclai@mail.nutn.edu.tw>; Fri, 11 Mar 2005 18:01:59 +0800
(CST)
X-Authentication-Warning: mail.nutn.edu.tw: iscan owned process
doing -bs
Received: from LOCALHOST (unknown [10.1.4.231])
    by smtp57.sms.ac (Postfix) with SMTP id 01EFE3825B
    for <cclai@mail.nutn.edu.tw>; Fri, 11 Mar 2005 05:00:47 -0500
(EST)
SUBJECT: zsuthiongie(3rd request)
To: cclai@mail.nutn.edu.tw
CONTENT-TYPE: text/plain
Message-Id: <20050311100047.01EFE3825B@smtp57.sms.ac>
Date: Fri, 11 Mar 2005 05:00:47 -0500 (EST)
From: zsuthiongie@invitation.sms.ac
Content-Length: 441
Status: R
```

Fig. 2. The header of an e-mail.

Table 1
Performance of three machine learning algorithms in Corpus I

Features	Preprocessing		Methods							
	Stemming	Stopping	*NB*		*k-NN*		*SVM*		*TF-IDF + SVM*	
			20:80	30:70	20:80	30:70	20:80	30:70	20:80	30:70
A			86.67	85.16	40.77	39.46	90.13	91.50	90.53	90.64
	✓		81.75	82.34	40.36	41.24	95.75	93.66	92.67	93.43
		✓	83.67	83.43	82.41	81.04	93.21	90.24	92.22	92.18
	✓	✓	88.42	88.32	82.33	81.65	94.33	94.75	91.25	91.33
H			88.59	88.57	41.23	42.57	91.58	92.57	91.22	92.46
	✓		87.65	86.70	41.10	42.19	88.23	85.13	93.22	94.28
		✓	87.27	85.29	81.42	81.74	93.61	92.54	92.11	92.66
	✓	✓	84.48	86.34	82.23	81.06	92.21	92.18	91.22	92.57
S			81.40	82.11	41.17	40.05	87.36	86.26	88.11	87.56
	✓		85.11	84.56	38.13	40.58	91.45	90.48	92.05	92.43
		✓	85.60	85.64	93.33	92.29	92.54	89.23	90.18	90.55
	✓	✓	85.68	83.88	82.56	82.18	92.16	93.58	91.88	90.24
B			73.22	72.36	38.66	39.46	92.46	92.34	92.67	93.43
	✓		78.32	76.25	33.16	31.23	86.58	87.57	81.23	82.46
		✓	74.28	76.28	72.68	71.06	89.28	88.46	86.94	85.66
	✓	✓	78.12	77.59	75.26	74.66	88.16	87.96	82.11	82.66

e-mails had to be compiled. Although several corpora are freely available on the Internet, we decided to experiment with e-mails we collected. The reason for it is that we want to measure the contributions of the different parts of an e-mail in spam filtering, the message with raw e-mail format seems more appropriate for us. The following two corpora were used in the experiments.

4.1. Corpus I

This corpus consists of 16,843 messages, 11,291 of which are marked as spam and 5552 are as legitimate. The former is collected from the *Babletext*[1] web site and the latter is provided by the *SpamAssassin*[2] web site. The spam rate is 67.04%.

4.2. Corpus II

This corpus includes 24,038 spam mails which are from the *E. M. Canada*[3] web site and the same number of legitimate mails from the *SpamAssassin* web site, with spam rate of 81.24%.

Here, we run experiments with different training and testing sets. The first pair and second pair of training and testing set are created by splitting each corpus at a ratio of 20:80 and 30:70, respectively.

In spam filtering tasks, the performance is often measured in terms of accuracy. Let N_L and N_S denote the total numbers of legitimate and spam messages, respectively, to be classified by the machine learning method, and $n_{C\rightarrow V}$ the number of messages belonging to category C that the method classified as category V (here, C, $V \in$ {legit, spam}). The accuracy is defined as following formula:

$$\text{Accuracy} = \frac{n_{L\rightarrow L} + n_{S\rightarrow S}}{N_L + N_S} \tag{7}$$

The overall performances of considered learning algorithms in different experiments are shown in Tables 1 and 2. From the results, we found the following phenomena.

1. Superiority of *SVM* method. The best results were obtained when *SVM* was applied to all features (i.e., header, subject, and body). This shows that using all the features gave better performance than any of the other techniques.
2. Good performance of *NB* method. *NB* performed reasonably consistent and good in different experimental settings except for the body feature considered alone. It might be surmised that too much useless information in the body for *NB* classifier.
3. Poor performance of *k-NN* method. *k-NN* performed the worst among all considered methods and the poorest in all cases. However, if more preprocessing tasks are utilized (i.e., stemming and stopping procedures are applied together), the better *k-NN* performs.
4. No effect of stemming, but stopping can enhance the e-mail classification. Stemming did not make any significant improvement for all algorithms in performance, though it decreased the size of the feature set. On the other hand, when the stopping procedure is employed, that is, ignoring some words that occur very frequently and offer no real description about the mail, we can obtain much better performance in some method. The phenomenon is obvious especially in *k-NN* method.

[1] Availability: http://www.babeltext.com/spam/.

[2] The mails in *SpamAssassin* are freely available from http://spamassassin.apache.org/publiccorpus/.

[3] See http://www.em.ca/%7Ebruceg/spam/.

Table 2
Performance of three machine learning algorithms in Corpus II

Features	Preprocessing		Methods							
	Stemming	Stopping	*NB*		*k-NN*		*SVM*		*TF-IDF + SVM*	
			20:80	30:70	20:80	30:70	20:80	30:70	20:80	30:70
A			86.67	85.23	40.22	40.75	92.24	90.13	92.47	92.66
	✓		84.96	82.34	40.18	40.26	93.87	93.94	94.12	93.26
		✓	83.43	83.67	80.59	81.24	92.56	93.67	93.17	93.22
	✓	✓	83.87	84.33	82.34	81.59	94.28	94.59	91.33	92.66
H			88.23	87.61	40.08	39.23	92.12	92.34	94.25	92.88
	✓		91.22	90.89	36.31	38.12	92.18	90.02	93.28	92.17
		✓	91.46	89.29	79.57	80.74	93.33	92.47	94.52	94.59
	✓	✓	89.48	89.34	78.24	79.53	92.39	92.21	93.68	94.44
S			81.40	89.44	33.26	41.88	90.57	90.47	92.57	90.87
	✓		88.34	87.56	40.17	38.12	92.64	91.88	90.01	89.52
		✓	82.15	84.32	81.87	79.31	89.23	92.36	90.17	90.24
	✓	✓	89.42	88.55	79.59	80.11	90.68	90.36	93.44	90.27
B			74.39	75.22	40.13	39.51	82.29	83.41	84.97	85.44
	✓		78.32	80.32	33.16	32.08	86.58	87.56	85.84	86.38
		✓	79.28	78.58	72.68	70.14	88.48	87.42	87.59	87.44
	✓	✓	78.12	78.55	72.74	71.74	87.96	88.16	84.32	85.67

5. Good performance with header. Among considered machine learning algorithms, the performance with header or subject information was almost as good as that with all features. This means that some useful information can be derived from the header or subject and only some parts (e.g., header, subject line) of an e-mail can aim at obtaining better performance.
6. Poor performance with body. The performance of each algorithm diminishes in the body. The result seems to show that although the feature space is large in all e-mail's body, a little relevant information can be used for classification.
7. Good performance with the integrated approach. On the average, the integrated approach can produce satisfactory performance as *SVM* does. Among all considered features, those in the header can be more reliable in discriminating spam than terms in other parts. Our observation implies that *TF-IDF* extracts particular features to give strong evidence for *SVM* to classify whether a mail is spam or not.

5. Conclusion

The detection of spam e-mail is an important issue of information technologies, and machine learning algorithms play a central role in this topic. In this paper, we presented an empirical evaluation of three machine learning algorithms for spam filtering. These approaches, *NB*, *k-NN*, and *SVM*, were applied to different parts of an e-mail in order to compare their performance. We also examined an integrated configuration that considered two methods for anti-spam. Experimental results indicate that *NB* and *SVM* yield better performance than *k-NN*. On the other hand, using two different scenarios actually improves the performance of anti-spam filtering. The phenomenon also found, at least with our test corpora, that classification with the header was almost as accurate as that with all features of an e-mail. Some avenues for future work include.

5.1. Comparison more machine learning algorithm

Except the approaches considered in this paper, a variety of machine-learning based methods, such as neural network, decision trees, and maximal entropy model, have been proposed for spam filtering. A fair comparison these approaches will be a matter of great interest to readers.

5.2. Consideration other types of feature in the e-mail

In addition to header, subject, and body, some specific features may aim at classifying the spam e-mails. For instance, the number of attachments and their types can be used in classification. On the other hand, more and more spam e-mails are HTML format; therefore, the html tags may be a useful feature.

Acknowledgements

This work was supported by the National Science Council, Taiwan, ROC, under Grant NSC 92-2218-E-024-004 and NSC 94-2213-E-024-005. The author would like to thank anonymous referees for their valuable comments and suggestions, Mr. Ming-Chi Tsai for his help to do the experiments.

References

[1] Aladdin Knowledge Systems, Anti-spam white paper, <http://www.eAladdin.com>.

[2] I. Androutsopoulos, J. Koutsias, K. Chandrinos, G. Paliouras, C.D. Spyropoulos, An evaluation of naive bayesian anti-spam filtering, in: Proc. of the Workshop on Machine Learning in the New Information Age: 11th European Conference on Machine Learning, 2000, pp. 9–17.
[3] I. Androutsopoulos, J. Koutsias, K. Chandrinos, G. Paliouras, C.D. Spyropoulos, An experimental comparison of naive bayesian and keyword-based anti-spam filtering with encrypted personal e-mail messages, in: Proc. of the 23rd Annual International ACM SIGIR Conf. on Research and Development in Information Retrieval, 2000, pp. 160–167.
[4] I. Androutsopoulos, G. Paliouras, V. Karkaletsis, G. Sakkis, C.D. Spyropoulos, P. Stamatopoulos, Learning to filter spam e-mail: a comparison of a Naïve Bayesian and a memory-based approach, in: Proc. of the workshop: Machine Learning and Textual Information Access, 2000, pp. 1–13.
[5] X. Carreras, L. Márquez, Boosting trees for anti-spam email filtering, in: Proc. of fourth Int'l Conf. on Recent Advances in Natural Language Processing, 2001, pp. 58–64.
[6] W.W. Cohen, Learning rules that classify e-mail, in: Proc. of AAAI Spring Symposium on Machine Learning in Information Access, 1996, pp. 18–25.
[7] H. Drucker, D. Wu, V.N. Vapnik, Support vector machines for spam categorization, IEEE Trans. Neural Netw. Vol. 10 (No. 5) (1999) 1048–1054.
[8] A. Kołcz, J. Alspector, SVM-based filtering of e-mail spam with content-specific misclassification costs, in: Proc. of TextDM'01 Workshop on Text Mining, 2001.
[9] M. Sahami, S. Dumais, D. Heckerman, E. Horvitz, A Bayesian approach to filtering junk e-mail, in: Learning for Text Categorization – Papers from the AAAI Workshop, 1998, pp. 55–62.
[10] G. Sakkis, I. Androutsopoulos, G. Paliouras, V. Karkaletsis, C.D. Spyropoulos, P. Stamatopoulos, Stacking classifiers for anti-spam filtering of e-mail, in: Proc. of the 6th Conf. on Empirical Methods in Natural Language Processing, 2001, pp. 44–50.
[11] R. Segal, J. Crawford, J. Kephart, B. Leiba, Spamguru: an enterprise anti-spam filtering system, in: Proc. of First Conf. on Email and Anti-Spam, 2004.
[13] F. Smadja, H. Tumblin, Automatic spam detection as a text classification task, in: Proc. of Workshop on Operational Text Classification Systems, 2002.
[14] D.C. Trudgian, Z.R. Yang, Spam classification using nearest neighbour techniques, in: Proc. of Fifth International Conf. on Intelligent Data Engineering and Automated Learning, 2004, pp. 578–585.
[15] V. Vapnik, The Nature of Statistical Learning Theory, Springer, 1995.
[16] Y. Yang, An evaluation of statistical approaches to text categorization, J. Inform. Retrieval vol. 1 (1999) 67–88.
[17] M. Woitaszek, M. Shaaban, R. Czernikowski, Identifying junk electronic mail in microsoft outlook with a support vector machine, in: Proc. of the 2003 Symposium on Applications and the Internet, 2003, pp. 166–169.

Chih-Chin Lai received the B.S. degree from National Chiao-Tung University in 1993, and the Ph.D. degree from National Central University in 1999. He is currently with Department of Computer Science and Information Engineering, National University of Tainan, Tainan, Taiwan, ROC, as an Associate Professor. His current research interests include evolutionary computation, pattern recognition, web intelligence, and image processing. Dr. Lai is a member of Taiwan Association of Artificial Intelligence (TAAI).