

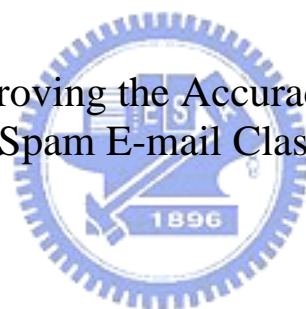
國立交通大學

電機資訊學院 資訊學程

碩士論文

針對中文改善垃圾信件過濾準確度之研究

Improving the Accuracy of
Chinese Spam E-mail Classification



研究生：馮寶永

指導教授：簡榮宏 教授

中華民國九十四年六月

針對中文改善垃圾信件過濾準確度之研究
Improving the Accuracy of Chinese Spam E-mail Classification

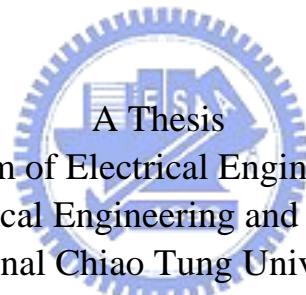
研究生：馮寶永

Student : Hendry Foeng

指導教授：簡榮宏

Advisor : Rong-Hong Jan

國立交通大學
電機資訊學院 資訊學程
碩士論文



Submitted to Degree Program of Electrical Engineering and Computer Science
College of Electrical Engineering and Computer Science

National Chiao Tung University

in Partial Fulfillment of the Requirements

for the Degree of

Master of Science

in

Computer Science

June 2005

Hsinchu, Taiwan, Republic of China

中華民國九十四年六月

授權書

(博碩士論文)

本授權書所授權之論文為本人在 國立交通 大學(學院) 電機資訊 系所

資訊 組 九十三 學年度第 二 學期取得 碩 士學位之論文。

論文名稱：_____

1. 同意 不同意

本人具有著作財產權之論文全文資料，授予行政院國家科學委員會科學技術資料中心、國家圖書館及本人畢業學校圖書館，得不限地域、時間與次數以微縮、光碟或數位化等各種方式重製後散布發行或上載網路。

本論文為本人向經濟部智慧財產局申請專利的附件之一，請將全文資料延後兩年後再公開。(請註明文號: _____)

2. 同意 不同意

本人具有著作財產權之論文全文資料，授予教育部指定送繳之圖書館及本人畢業學校圖書館，為學術研究之目的以各種方法重製，或為上述目的再授權他人以各種方法重製，不限地域與時間，惟每人以一份為限。

上述授權內容均無須訂立讓與及授權契約書。依本授權之發行權為非專屬性發行權利。依本授權所為之收錄、重製、發行及學術研發利用均為無償。上述同意與不同意之欄位若未鈎選，本人同意視同授權。

指導教授姓名：簡榮宏

研究生簽名：

(親筆正楷)

學號：9167582

(務必填寫)

日期：民國 _____ 年 _____ 月 _____ 日

-
1. 本授權書請以黑筆撰寫並影印裝訂於書名頁之次頁。
 2. 授權第一項者，所繳的論文本將由註冊組彙總寄交國科會科學技術資料中心。
 3. 本授權書已於民國 85 年 4 月 10 日送請內政部著作權委員會（現為經濟部智慧財產局）修正定稿。
 4. 本案依據教育部國家圖書館 85.4.19 台(85)圖編字第 712 號函辦理。

國立交通大學

論文口試委員會審定書

本校 電機資訊學院專班 _____ 資訊 _____ 組 馮寶永 君

所提論文

(中文) 針對中文改善垃圾信件過濾準確度之研究

(英文) Improving the Accuracy of

Chinese Spam E-mail Classification

合於碩士資格水準、業經本委員會評審認可。



口試委員： _____

指導教授： _____

班主任： _____

中華民國 _____ 年 _____ 月 _____ 日

針對中文垃圾信改善信件過濾準確度之研究

學生：馮寶永

指導教授：簡榮宏 教授

國立交通大學電機資訊學院 資訊學程（研究所）碩士班

摘 要



垃圾信件在網際網路中已經成為了極大的威脅，不僅浪費網路資源，也浪費使用者的時間。本篇論文分析製造垃圾信的各種方法與阻擋垃圾信件的各種過濾機制，此類過濾方法最有名的是採用機率與統計的分析。由於大部分的過濾垃圾信件系統只是處理英文信件，於是，論文將針對中文垃圾信，採用 Bayesian classifier 的方法過濾中文垃圾信件。從實驗結果可以得知這個方法可以提升過濾的準確度。

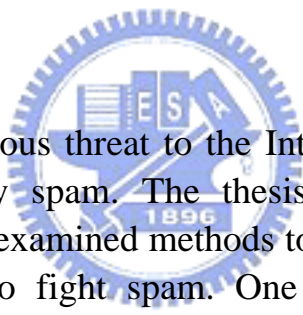
Improving the accuracy of Chinese SPAM email classification

Student : Hendry Foeng

Advisors : Dr. Rong-Hong Jan

Degree Program of Electrical Engineering Computer Science
National Chiao Tung University

ABSTRACT

The logo of National Chiao Tung University is a circular emblem. It features a gear-like outer border. Inside the circle, there are stylized Chinese characters and the year '1896'. The letters 'E', 'S', and 'A' are also visible within the design.

Spam is now become a serious threat to the Internet. This thesis examines the problems and impact caused by spam. The thesis also examined methods and techniques to generate spam, and examined methods to classify spam. There are many filtering algorithms introduced to fight spam. One of the most popular filtering algorithms is statistical based filtering, which is based on Bayesian classification theorem. However, these algorithms focus on the English e-mails. This thesis presents a Chinese e-mail classifier, which is based on Bayesian classifier to classify Chinese spam e-mails. The experiment results show that the proposed Chinese e-mail classifier could perform a high accuracy.

Acknowledgements

I extend my sincere gratitude and appreciation to many people who made this master's thesis possible. Special thanks are due to my advisor Dr. Rong-Hong Jan for his guidance in my thesis work. Many thanks are due to my parents, my sister, my brother, my friends and my girlfriend for their love, patience and their big support during these three years.



Contents

	Abstract (in Chinese)	i
	Abstract (in English)	ii
	Acknowledgements	iii
	Contents	iv
	List of Tables	v
	List of Figures	vi
1.	Introduction	1
2.	Generation of Spam	3
2.1	Distribution Model of SPAM mails	3
2.2	E-mail Harvesting Techniques	4
2.3	Methods of creating Spam messages	4
2.4	Spam Categorization.....	5
3.	SPAM Blocking and Filtering Techniques.....	8
3.1	Pattern Matching Based Filtering	8
3.2	DNS Based Filter	8
3.2.1	DNS Blackhole Lists.....	8
3.2.2	DNS Lookup.....	10
3.3	Signature Based Filtering	11
3.4	Challenge and Response	12
3.5	Statistical Based Filtering	13
4.	Implementation of Anti Spam for Chinese Spam e-mail.....	15
4.1	Problem Description	16
4.2	System Architecture	16
4.3	System Implementation	19
5.	Experiments and Performance Evaluation.....	20
5.1	Testing Environment.....	20
5.2	Testing Methodology.....	21
5.3	Experiment Result.....	22
6.	Conclusion and Future Work	27
	References	28

List of Tables

Table 2-1	Spam categorization	7
Table 5-1	Training and Experiment Data	22
Table 5-2	Experiment data	26



List of Figures

Figure 2-1	Distribution of SPAM 1	3
Figure 2-2	Distribution of SPAM 2	4
Figure 2-3	Fraud purpose spam example	6
Figure 3-1	DNS Based Black List	9
Figure 4-1	System Architecture	16
Figure 4-2	Anti-Spam System Core Modules	17
Figure 5-1	Simulation Architecture	21
Figure 5-2	False Positive Rate	23
Figure 5-3	False Negative Rate	24
Figure 5-4	Filtering Accuracy	25



Chapter 1

Introduction

Recent years, as the population of Internet is growing rapidly, e-mail becomes one of the most popular applications used in Internet, because of its easy to use and effective. Most of users use e-mail application as communication tools. Unfortunately, many marketers make use of e-mail to promote or sell their products by mailing to users without subscription from users. These annoying e-mails are also called as spam. Spam is defined as unsolicited e-mail which most of them sent from anonymous sender. Most of spam e-mail contains marketing purpose content and a number of annoying e-mails such as virus, fraud, etc. Anonymous sender is defined as non-traceable source.



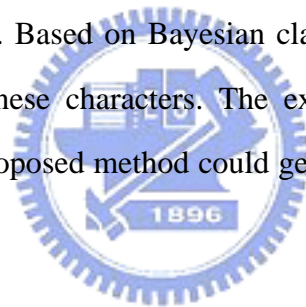
Today, spam becomes a serious threat to Internet. According to figures from Symantec [1], the volume of spam e-mail crossed the 50 percent barrier as long ago as July 2003. In December 2004, culled from network traffic traveling through its servers, showed that 67 percent of e-mail is spam.

With the speedy growth of spam, the problems are getting more and more serious, ranging from the general annoyance to users of having to filter dozens of unwanted email from their inbox to companies having financial loss due to the cost of productivity. For example, the huge amount of spam has lead to significant decreases in worker productivity, network throughput, data storage space, and mail server efficiency. Each employer spent a portion of time to review and delete the unwanted email, which lead to a decrease in productivity. The increased network traffic caused by spam has a bad effect on network performance. It could also affect the CPU loading of the mail server, when the CPU of the mail server is overloaded, legitimate

message will be delayed while the queue is processed.

A number of spam filtering techniques have been introduced to fight spam, such as Pattern matching based filtering; DNS based filtering [2]; Challenge and response system [3], Statistical based filtering (e.g. Bayesian classifier [4]), etc. This thesis will discuss these techniques later in chapter 3. Statistical based filtering, which based on Bayesian classifier [4] is the best current solution to filter spam e-mail. It could filter above 98% of spam with low false positive rate.

Most of the anti-spam system only parses English word as keyword. Many of these anti-spam systems could not perform very well when filtering e-mail using native language, such as Chinese language. To address this problem, we propose a method to classify Chinese spam e-mail. Based on Bayesian classification algorithm, we add a Chinese parser to parse Chinese characters. The experiment results show that by adding Chinese parser, our proposed method could get a more accuracy in classifying Chinese spam messages.



In the next chapter we will discuss about the distribution of spam, and the technique of spam generation. In chapter 3, we will describe methods of spam blocking and filtering. Then we will describe about the implementation of Chinese parser in chapter 4. We will describe test and experiment results in chapter 5. Finally in chapter 6, we give conclusion and possible future work.

Chapter 2

Generation of SPAM

Before we continue to the methods of anti-spam filtering, first we discuss about the characteristic of spam message and techniques to send spam e-mail.

2.1 Distribution model of SPAM mails

1. Spammer – Open relay mail server – Destination mail server

Email application use SMTP [5] (Simple Mail Transfer Protocol) to transfer between mail servers. Email server that configured as open relay allows relaying mails to any destination address. A lot of email server administrator did not disable relay function; this would let spammers to use their email server to send spam mail. A spammer may discover which email server configured as open relay, and then be able to send a lot of messages (spam e-mail) from his computer.

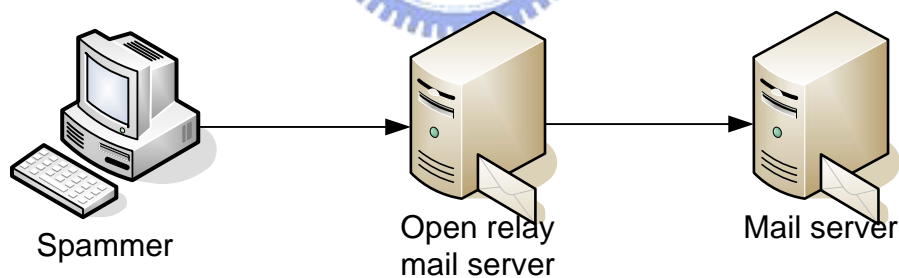


Figure 2-1. Distribution of SPAM

2. Spammer – destination mail server

Another method to send spam mail is directly sent from spammer's computer. Spammers could configure their computer as SMTP [5] server, and then send a lot of spam messages from their server directly to destination.

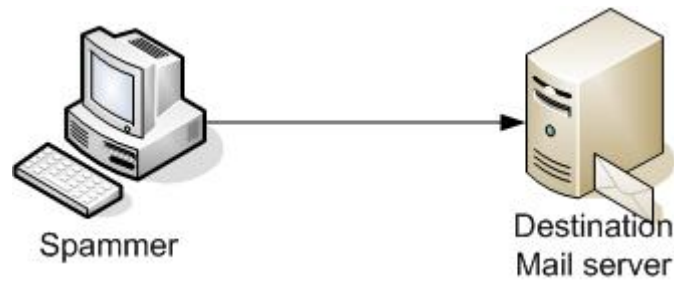


Figure 2-2. Distribution of SPAM

2.2 E-mail Harvesting Techniques

Spammers also need e-mail lists as their destination user. Email database could be collected by scanning web pages for e-mail addresses and associated names. By scanning HTML source for mailto: tags, e-mail addresses can be found. Another way is by renting or purchasing email lists [6], such as <http://www.rent-a-list.com>, <http://www.optininc.com>.

Spammers could also use “dictionary attacks” [7] technique in which the spammer generates a number of likely-to-exist addresses out of names and common words. For example, if there is someone with the address `roger@company.com`, where “company.com” is a popular ISP or mail provider, it is likely that he frequently receives spam.

2.3 Methods of creating Spam messages.

The creation of a spam e-mail message is important in defeating anti-spam filters. Some of the techniques are:

(1) Blank HTML

This technique involves sending email messages, which contain no text, but contain an image that could not be parsed by any spam filters.

Example: `<html></html>`

(2) Invisible text

This technique attempt to hide legitimate text inside a message, since many anti-spam filters compare the number of spam words with non-spam words to calculate percentage of spam, by adding such legitimate text inside will affect the calculation.

Example: HTML comments tag: `<!-- non spam message >`

(3) Letter spacing or special character spacing

Spammers try to prevent filters from recognizing the tokens in the mail by breaking them up, by using white space in the middle of words.

Example: T h i s i s S P A M

T-h-i-s i-s S-P-A-M



(4) Vertical slicing

This method tries to prevent filters from recognizing the tokens in the e-mail by slicing the message text into vertical strips.

Example: T I S
 H S P
 I A
 S M

2.4 Spam Categorization

According to the contents of the spam e-mail, most of the spam message can be categorized into:

(1) Marketing purpose

This advertising related spam occupied most of the spam e-mails. For example: advertising product on sale, diet program, job hunting, etc. Table 2-1 shows a more detailed spam categorization.

(2) Fraud, annoying or disturbing purpose

Fraudulent message is employed to encourage users to open the spam message. For example, by altering the subject line in the message header to imply the message is not spam. Fraud may also be present in the content of the messages, for example, to advertise illegal financial schemes. Figure 2.3 shows an example of fraud message which inform user to update his/her bank account information on a website.

(3) Mail containing virus

Mail that contains virus is also considered as a spam message.

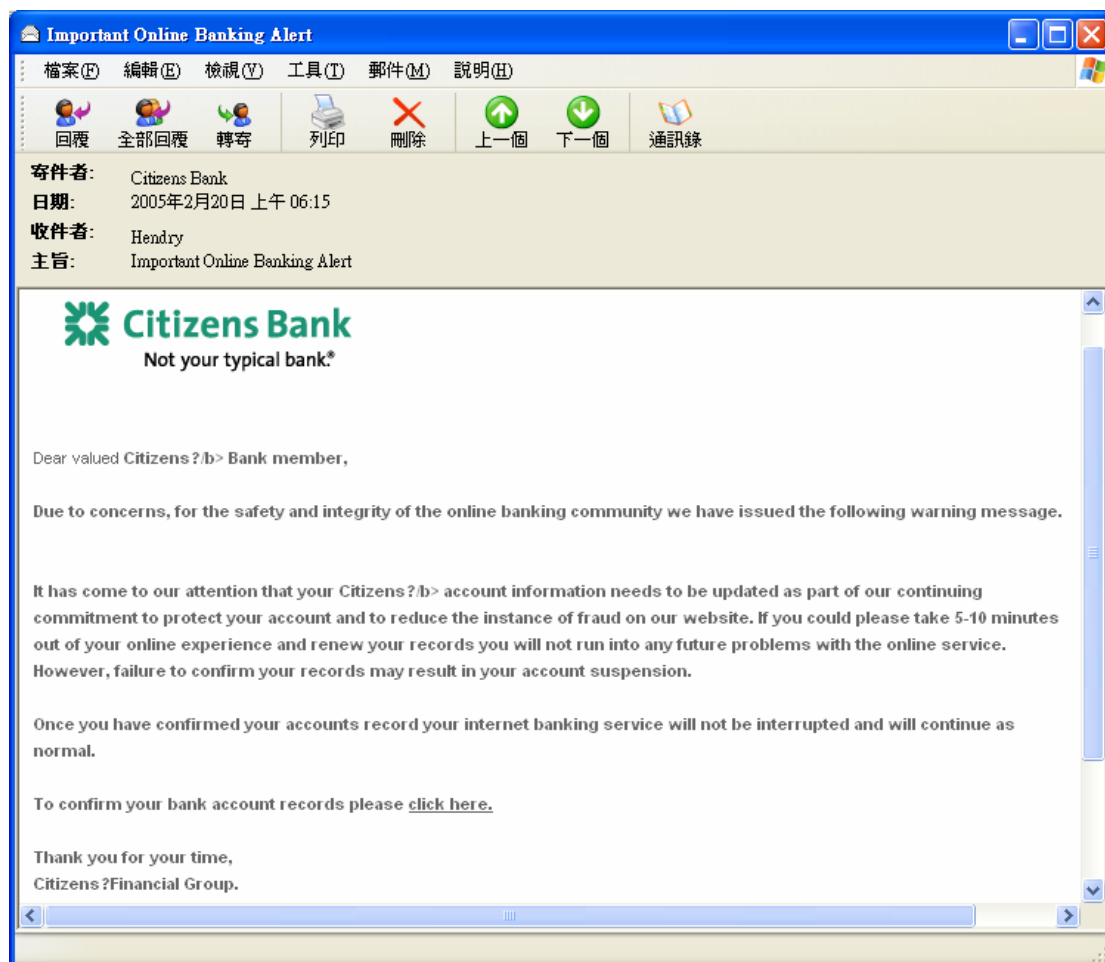


Figure 2.3 Fraud purpose spam example

Product on sale	Real estate	Phone bill saving
Diet program	Horoscope related	Web design service
Sex and porn related	Vacation	Consumer electronics
Newsletters	Books	Chat room
Finance related	Virus news	Audio video related
CD-ROM related	Possible virus	Online gaming
Job hunting	Health and medicine	Workshop and seminar
Product on sale	Real estate	Phone bill saving
Diet program	Horoscope related	Fraud purpose

Table 2-1. Spam categorization



Chapter 3

SPAM blocking and filtering techniques

The continuous growth of spam has resulted in equally growth of spam filter programs. Methods and techniques used by these anti-spam programs mostly based on pattern matching based filtering, DNS based filtering [2], challenge and response system [3], statistical based filtering [4], and signature based filtering, which will be describe in this chapter. Spam blocking and filtering techniques can be grouped into few categorizes:

3.1 Pattern matching based filtering

Pattern matching based filtering also known as rule based filtering relying on users specifying lists of words or regular expression which is categorized as spam mail, including sender, title and body of message. Mail server would then reject any email containing the phrase.

Example: if e-mail title contains “Viagra” then move to spam folder

if sender = spam@spam.org then move to spam folder

The disadvantage of this technique is that it relies on manually constructed pattern matching rules that need to be tuned over time which is a time wasting. Furthermore, any legitimate email which matches the rule will cause a false positive.

Spammers can trick the pattern matching filter by changing the phrases and spelling they use, which will pass the pattern matching filter.

3.2 DNS based filter

3.2.1 DNS Blackhole Lists (DNSBL)

DNS Blackhole Lists [2] is a means by which an Internet site may publish a list of IP addresses, in a format which can be easily queried by computer programs on the Internet. As the name suggests, the technology is built on top of the Internet DNS or Domain Name System [8]. DNSBL [2] are chiefly used to publish lists of addresses linked to spamming. Most mail transport agent (mail server) software can be configured to reject or flag messages which have been sent from a site listed on one or more such lists.

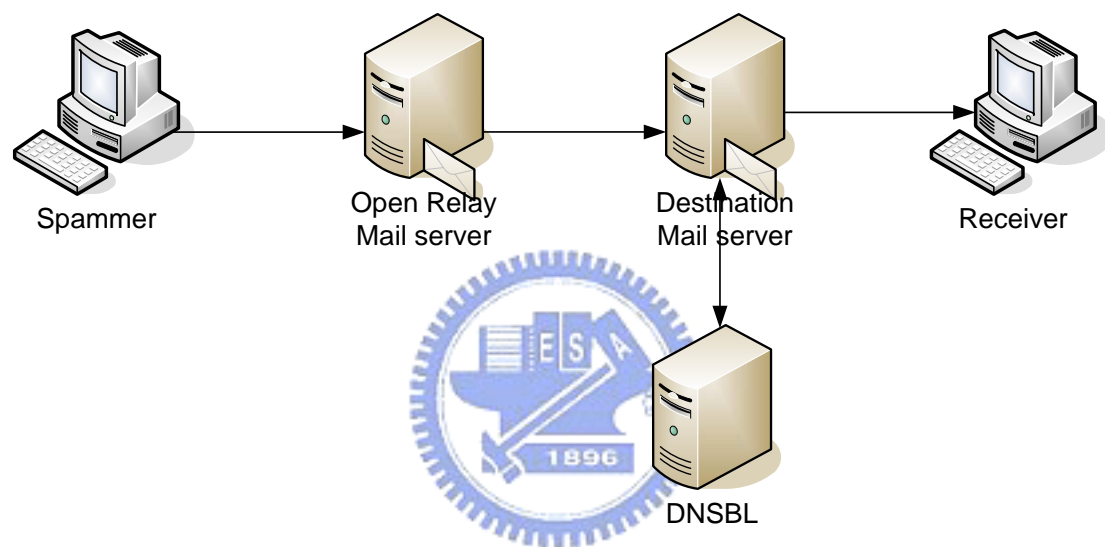


Figure 3-1. DNS Based Black List

When a mail server receives a connection from a client, and wishes to check that client against a DNSBL (for example, spammers.example.net), it does the following steps:

1. Reverse the bytes of the client's IP address, example: 192.168.42.23 reverse to 23.42.168.192.
2. Append the DNSBL's domain name: 23.42.168.192.spammers.example.net.
3. Look up this name in the DNS as a domain name ("A" record). This will return either an address, indicating that the client is listed; or an "NXDOMAIN" ("No such domain") code, indicating that the client is not.

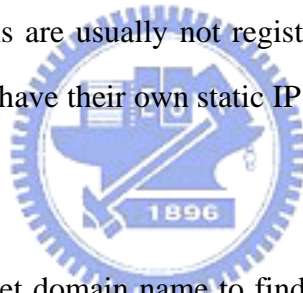
4. Optionally, if the client is listed, look up the name as a text record ("TXT" record).

Most DNSBLs publish information about why a client is listed as TXT records.

Looking up an address in a DNSBL is thus similar to looking it up in reverse-DNS. The differences are that a DNSBL lookup uses the "A" rather than "PTR" record type, and uses a forward domain (such as spammers.example.net above) rather than the special reverse domain in-addr.arpa.

3.2.2 DNS Lookup

DNS Lookup method tries to eliminate spam sent by e-mail servers connected through Internet dial-up connections, as well as most ADSL and cable connections. IP addresses of those connections are usually not registered to any DNS as a qualified host meaning that they do not have their own static IP and a registered Fully Qualified Domain Name (FQDN) [8].



A DNS lookup uses an Internet domain name to find an IP address, where a reverse DNS lookup is using an Internet IP address to find a domain name. Reverse DNS lookup technique is able to identify if the sending e-mail server is legitimate and has a valid host name.

Many spammers use misconfigured hosts to masquerade the source of the spam. A DNS query that does not recover a matching host name and IP address is a good indication that the message is spam.

DNS lookup is not always a good solution. Many legitimate e-mail servers are incorrectly configured, or have intentionally not registered a name with DNS, so a reverse query does not return a matching host name. Also, this anti-spam method runs DNS queries on a large number of e-mails and consumes valuable network resources.

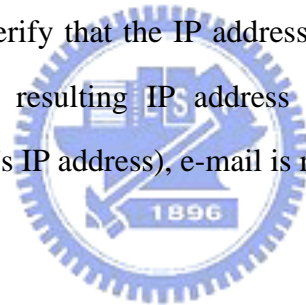
There are 3 ways to do DNS lookups technique to fight spam:

1. Reverse DNS lookup

When e-mail is sent from one server to another, the receiving server performs a reverse DNS lookup on the IP address of the incoming connection and checks if it matches what is in the header information of the e-mail. This is a means of finding out if the sender is attempting to spoof the address from where the e-mail is actually originating.

2. HELO lookup

When an e-mail is arrived, the receiving server will get the host name of the sending e-mail server from the SMTP HELO command, perform a simple DNS query (forward DNS lookup) and verify that the IP address is indeed the IP address of the incoming connection. If the resulting IP address does not match the incoming connection IP address (sender's IP address), e-mail is rejected.



3. Sender's address lookup

When ISPs check whether an incoming e-mail is accepted, they can do a DNS check on the sender's e-mail address. For example, if your address is user@domain.com, then the ISP does an nslookup on domain.com. If no records are found - the message is rejected. A variation of this method is checking if there is an MX DNS record of the domain.com. MX record returns an address like mx1.domain.com used to connect to the server that accepts messages for domain.com. Even if the domain in the sender's e-mail address is valid, but there is no e-mail server for domain.com, the message is rejected.

3.3 Signature Based Filtering

Signature-Based filters work by comparing the signatures between incoming email

and known spam. One way to calculate a signature for an email would be to assign a number to each character, then add up all the numbers. It would be unlikely that a different email would have exactly the same signature. Spam e-mail is collected by a honeypot [9], which maintain fake email addresses. Any email sent to these addresses must be spam. So when they see the same email sent to an address they're protecting, they know they can filter it out.

The way to attack a signature-based filter is to add random stuff to each copy of a spam, to give it a distinct signature. When you see random junk in the subject line of a spam, that's why it's there to trick signature-based filters.

The spammers have always had the upper hand in the battle against signature-based filters. As soon as the filter developers figure out how to ignore one kind of random insertion, the spammers switch to another. So, signature-based filters have never had very good performance.



The advantage of this method is that it has a very low false positive rate. It is due to the distinction of every e-mail.

Because of the signature distinction of every e-mail, it is easy to attack signature based filter by adding random stuff to the message which could produce a high false negative rate.

3.4 Challenge and response

Challenge and response [3] technique require senders to pass some tests before their message are delivered. When email server see a possible spam E-mail from somebody you've never corresponded with before, it will hold the mail and e-mail back a challenge to confirm that the person is a real sender and not a mailing robot, in particular a spammer. If sender responded the challenge, its domain will be added to

white list.

Challenge and Response system are extremely effective at eliminating spam, even for addresses that receive hundreds of spam messages per day. With Challenge and Response system, the only spam that gets delivered is spam that has been personally authorized by the spammer.

The disadvantage of this method is that some challenge and response systems interact badly with mailing list software. If a person subscribed to a mailing list begins to use Challenge and Response software, posters to the mailing list may be confronted by large numbers of challenge messages. Many regard these as junk mail equal in annoyance to actual spam.

Some Challenge and Response systems interact badly with other Challenge and Response systems. If two persons both use Challenge and Response and one e-mails the other, the two Challenge and Response systems may become trapped in a loop, each challenging the other, neither one willing to deliver the challenge messages or the original message.

Spammers and viruses send forged messages; email with other people's addresses in the "From" headers. A Challenge and Response system challenging a forged message will send its challenge to the uninvolved person whose address the spammer put in the spam. This effectively doubles the amount of unwanted email being distributed.

3.5 Statistical Based filtering

Statistical based filtering which based on Bayesian classification [4] is the best current solution to filter spam e-mail. It could filter above 98% of spam with low false positive rate. Bayesian classification method itself is not first applied in spam filtering.

It was previously introduced in machine learning which applied to text categorization, for example: Apte and Damerau, 1994 [10]; Lewis, 1996 [11]; Dagan et al., 1997 [12]; Sebastiani, 1999 [13]; etc. Bayesian classification was first applied to anti-spam filtering in 1998 by Sahami [4].

A Bayesian classifier is simply a Bayesian network applied to classification task. It contains node C representing the class variable and a node X_i for each of the attribute (word) in a message. Given a specific instance x (assign X_1, \dots, X_n to each attribute), following the Bayesian theorem, we compute the probability of $P(C = c | X = x)$ for each possible class C :

$$P(C = c | X = x) = \frac{P(X = x | C = c)P(C = c)}{P(X = x)} \quad (1)$$

Assume that each attribute X_i is conditionally independent (also called naïve Bayesian Classifier) of every other attributes, given the class variable C , this yields:

$$P(X = x | C = c) = \prod_i P(X_i = x_i | C = c) \quad (2)$$

From Bayes' theorem and the theorem of total probability, given the attribute $x = X_1, \dots, X_n$ of a document d , the probability that document d belongs to category c (spam or non-spam) is:

$$P(C = c | X = x) = \frac{P(C = c).P(X = x | C = c)}{\sum_{k \in \{spam, non-spam\}} P(C = k).P(X = x | C = k)} \quad (3)$$

From equation (2) and (3), we get:

$$P(C = c | X = x) = \frac{P(C = c). \prod_{i=1}^n P(X_i = x_i | C = c)}{\sum_{k \in \{spam, non-spam\}} P(C = k). \prod_{i=1}^n P(X_i = x_i | C = k)} \quad (4)$$

$P(X_i | C)$ and $P(C)$ can be computed as relative frequencies from training database. Each word has particular probabilities of occurring in spam email and in non-spam email. For instance, most email users will frequently encounter the word “Viagra” in spam email, but will seldom see it in non-spam email. The filter doesn't know these probabilities in advance, and must first be trained so it can build them up. To train the filter, the user must manually indicate whether a new email is spam or non-spam. For all words in each training email, the filter will accordingly adjust the words' spam and non-spam probabilities in its database. For instance, Bayesian spam filters will typically have learned a very high spam probability for the word "Viagra", but a very low spam probability (and a very high non-spam probability) for words seen only in non-spam email, such as the names of friends and family members.

After training, the spam and non-spam word probabilities are used to compute the probability that an e-mail with a particular set of words in it belongs to either the spam or non-spam category. Each word in the email contributes to the e-mail's spam probability. This contribution is called the posterior probability and is computed using Bayes theorem. The filter will mark the email as spam or non-spam according to its probability. E-mail marked as spam can then be automatically moved to a "SPAM" e-mail folder.

The advantage of Bayesian filters is that it could filter above 98% of spam with low false positive rate.

The disadvantage of Bayesian filters is that they need to be trained. The user has to tell the classifier whenever they misclassify an e-mail.

Chapter 4

Implementation of Anti Spam for Chinese Spam e-mail

4.1 Problem description

With anti-spam that is only parsed English e-mail, is not enough to cope with Chinese spam e-mail. In this chapter, focus on Chinese Spam e-mail, we present an anti-spam for Chinese spam e-mail which is based on Bayesian classification.

4.2 System Architecture

We choose Bayesian classifier method as our classification algorithm since it the best method to fight spam e-mail. This anti-spam system is built as e-mail proxy (see Figure 4.1). When e-mail client retrieve e-mails from server, all the e-mails would get through the classifier. When the anti-spam system first setup, it doesn't understand what is consider to be spam or legitimate e-mail. Therefore, it will have to be trained to meet our needs. The training process is done through anti-spam user interface. This user interface can be accessed through web client.

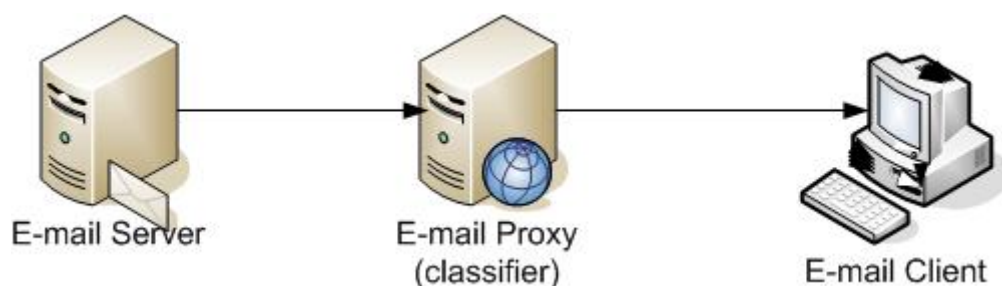


Figure 4-1. System architecture

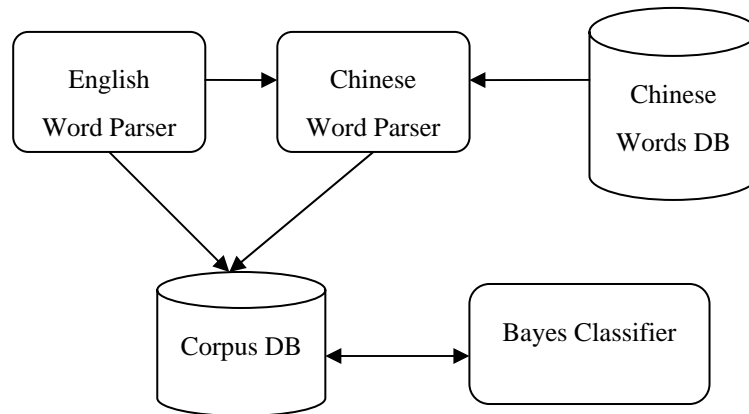


Figure 4-2. Anti-Spam System Core Modules

This anti spam system consists of some core modules, which play important roles in classification. The core modules are:

- (1) English word parser: this module performs English words parsing. Each word parsed from the messages will be saved to corpus (trained word) database.
- (2) Chinese word parser: this module performs Chinese words parsing. This module only parse Chinese phrase which is made up of more than 2 Chinese characters. In this case, we are not interested with single Chinese character because it is too common in spam or non-spam emails. In addition, parsing single Chinese character could also waste database resource.
- (3) Bayes Classifier: performs Bayes algorithm computation.
- (4) Corpus Database: trained words database, which records frequencies of every token in spam or non spam.
- (5) Chinese words database (CEDICT) [14]: A free Chinese words database available in the Internet.

When the system just setup, it doesn't understand what is consider to be spam or legitimate e-mail, the corpus database is empty. Therefore, it will have to be trained to meet our needs. The training performs the following actions:

- (1) Retrieve email contents line by line, perform text pre-processing to cope with

spam techniques describe in chapter 2.

- (2) Parse the line processed in step 1, and retrieves the English words.
- (3) For the line processed in step 1, perform Chinese words parsing.
- (4) Update words to corpus database, along with the information of words' frequencies, classification, and the probability of the words.

Once the anti-spam system has been trained, the system could classify the future e-mail based on the corpus database. The classification flow performs:

- (1) E-mail retrieve is triggered from e-mail client. At the time, the anti-spam system, which is also act as e-mail proxy, retrieve e-mail from mail server.
- (2) Each of e-mail content will be parsed by Bayesian classifier module. Bayesian classifier will compute probability with the evidence collected in corpus database.

E-mail tokenization [15] is one of the most important steps in classification. Tokenizing the messages has a big influence on the overall results of the classification engine. The tokenization criteria for English message are:

1. Case is preserved.
2. Periods and commas are constituents if they occur between two digits. E.g. 192.168.1.1 (IP address), \$5-10 (a price range), etc.
3. Message that contains HTML tag, get marked according to its HTML tag. E.g. `` becomes `html:imgwidth99`.
3. Words that occur in e-mail header such as To, From, Subject, and Return-Path lines, or within urls, get marked accordingly. E.g. "hello" in the Subject line becomes "Subject:hello".

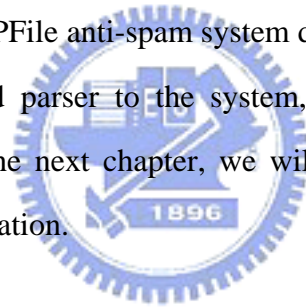
Tokenization method for Chinese message is different from the tokenization for English message. The tokenization criteria for Chinese message are:

- (1) Only Chinese phrase which consist of more than 2 characters are considered as a token.

- (2) The token must be found in Chinese words database.
- (3) Any non-word symbol which will be ignored.

4.3 System Implementation

Since there are many English Anti-spam systems which based on Bayesian classifier is available on Internet, we only implement the module for Chinese words parser. We choose POPFile [16] anti-spam system as our base system. POPFile is an open source POP3 proxy which uses a Bayesian filter to classify e-mail. The system runs on Windows, Mac OS X, Linux, Solaris, FreeBSD and OS/2. It is written in Perl programming language. According to the statistic report [17], POPFile performs an average accuracy of 98.1% across all users submitting accuracy statistics from their use of the program. Since POPFile anti-spam system doesn't parse Chinese words, we implement the Chinese word parser to the system, which is also written in Perl programming language. At the next chapter, we will discuss about the experiment details and performance evaluation.



Chapter 5

Experiments and performance evaluation

In this chapter, we will describe experiments comparing the Bayesian classifier without Chinese words parser with the Bayesian classifier with Chinese words parser in classifying Chinese spam e-mail.

As described in Chapter 4, we use POPFile [16] (an open source e-mail classifier) as email proxy. POPFile [16] is an automatic mail classification tool. The classification is done using a Bayes algorithm. In other words, it uses statistics to track which words are likely to appear in which messages. Once properly set up and trained, it will scan all email as it arrives and classify it based on your training. Since it doesn't support Chinese language parser, we implement the Chinese words parser module to the system.



5.1 Testing Environment

Testing environment can be summarized as follows:

Software specifications:

E-mail Proxy: POPFile version.0.2.2x [16]

E-mail Server: Microsoft SMTP server and TmPOP3 server [18]

E-mail Client: Microsoft Outlook Express 6.0 [19]

Developer language: Perl language

Operating System: Microsoft Windows XP Professional Edition with SP2

Hardware specification:

Athlon-XP 2500+ with 512MB RAM, 40 GB harddisk.

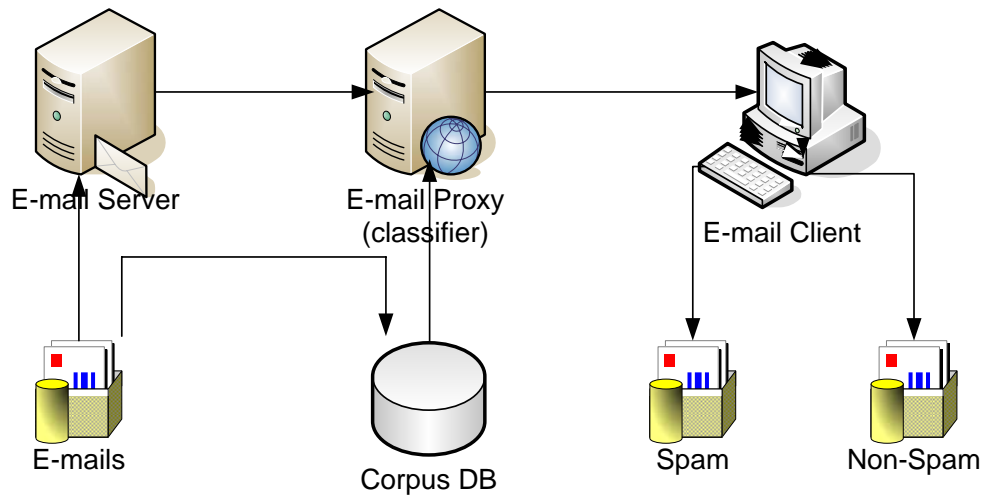


Figure 5-1. Simulation Architecture

5.2 Testing methodology

Testing procedure can be summarized as follow:

- (1) Select a number of spam and non-spam emails randomly from e-mail collection as experiment e-mails.
- (2) Select a small amount of spam and non-spam emails randomly from first step.
- (3) Train the classifier by parsing spam and non-spam emails selected in second step.
- (4) Put the experiment e-mails which selected in the first step to e-mail server.
- (5) Retrieve e-mail with e-mail client through e-mail proxy.

Table 5-1 shows the testing information about total of initial training e-mail and total of tested e-mail. The total of tested e-mail is ranging from 800 to 6000 e-mails, the total of training e-mail is ranging from 55 to 1200 e-mails.

no	Total of training data		Total of experiment data	
	good	spam	good	spam
1	15	40	100	699
2	30	80	100	699
3	30	60	90	680
4	30	60	90	680
5	20	50	102	500
6	50	100	100	1000
7	50	100	100	1000
8	50	100	100	1000
9	50	200	100	1000
10	50	200	100	1000
11	50	150	100	1500
12	50	300	100	1500
13	50	300	100	1500
14	50	200	200	2000
15	50	200	200	2000
16	50	200	250	2000
17	50	200	250	2000
18	200	600	1000	3000
19	200	800	1000	4000
20	200	800	1000	4000
21	200	1000	1000	5000
22	200	1000	1000	5000

Table 5-1. Training and experiment data

5.3 Experiment Result

In the experiment, we compare the false positive rate and false negative rate. The false positive is defined as non-spam e-mail which is classified as spam e-mail. The false negative is defined as spam e-mail which is classified as non-spam e-mail.

The experiment e-mails is collected from author's emails, which most of them are Chinese e-mails. Each of e-mail is saved as text file format, which is readable by SMTP server and Outlook Express.

Performance metrics used in this chapter are defined as follows:

False positive rate = $1 - (\text{classified as non-spam} / \text{total non-spam})$

False negative rate = $1 - (\text{classified as spam} / \text{total spam})$

Accuracy = $(\text{classified non-spam} + \text{classified spam}) / (\text{total non spam} + \text{total spam})$

As we can see the false positive rate in Figure 5-2, it shows that the false positive rate of Bayesian classifier with Chinese parser is better than Bayesian classifier without Chinese parser. Some of the experiments perform the same false positive rate. When we checked to false positive e-mails, their contents include most of Spam words, which might influence the Bayesian filter. Therefore, the tokenization technique is very important.

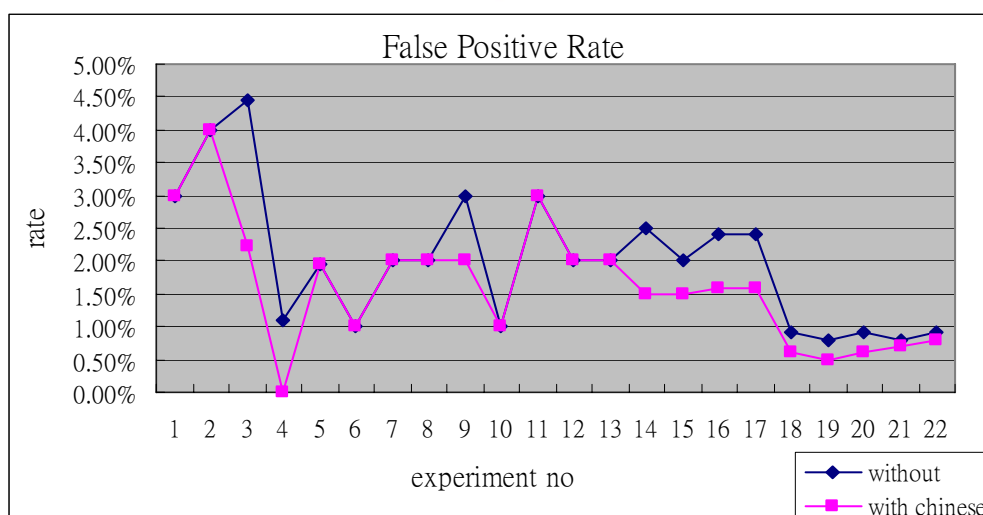


Figure 5-2. False Positive Rate

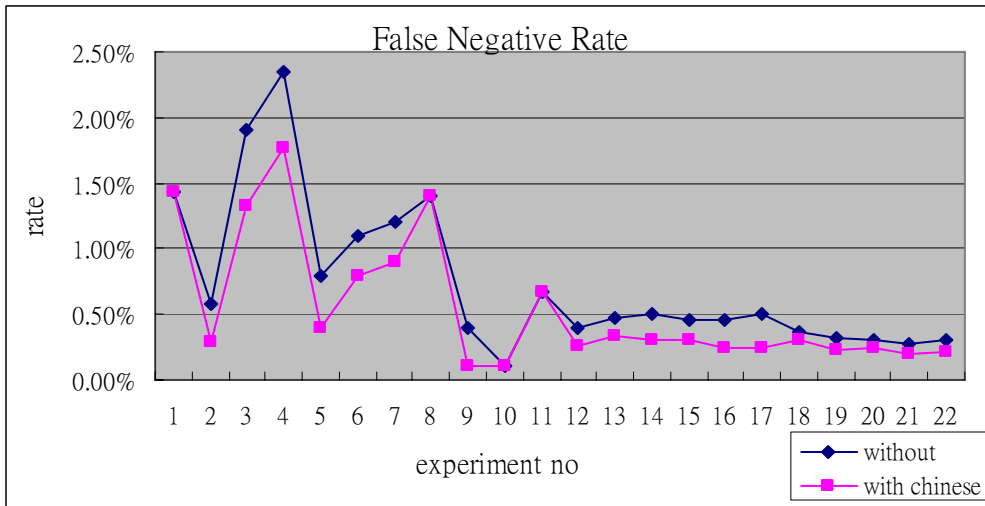


Figure 5-3. False Negative Rate

Figure 5-3 shows the false negative rate of the experiment. It shows that the Bayesian filter with Chinese parser performs better than the Bayesian filter without Chinese parser almost in every experiment. The main reason why the Bayesian with Chinese parser performs better is related to the contents of the e-mails, which most of them are Chinese e-mails.

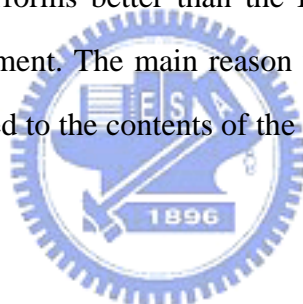


Figure 5-4 shows the overall filtering accuracy of spam and non-spam e-mail between Bayesian filter with Chinese parser and Bayesian filter without Chinese parser. It shows that the accuracy of Bayesian filter with Chinese parser performs better.

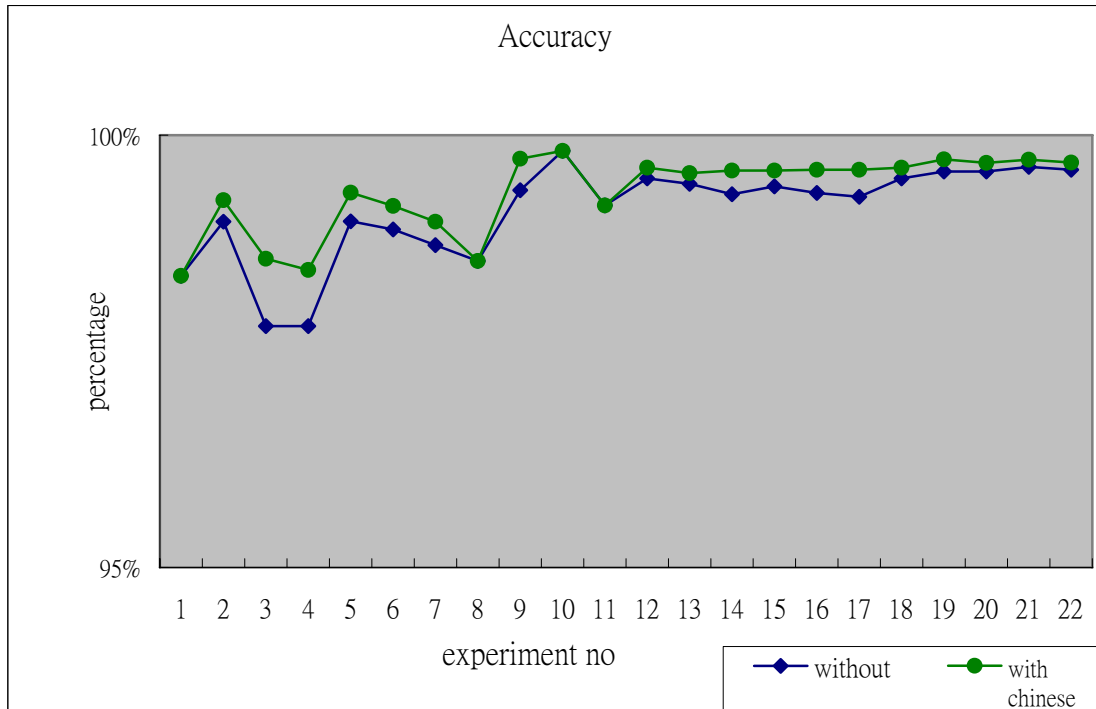


Figure 5-4. Filtering Accuracy

Table 5-2 shows the more details of our experiment's data, including the false positive rate and false negative rate of every experiment.



no	init		experiment		result without Chinese parser					result with Chinese parser					ratio		overall accuracy				
	good	spam	good	spam	good	spam	false pos	false neg.	Un classified	good	spam	false pos	false neg.	Un classified	init good: exp good	init spam: exp spam	init: spam	without	with chinese		
1	15	40	100	699	97	689	3	9	1	97	689	3	9	1	0.15	0.06	0.07	98.37%	98.37%		
2	30	80	100	699	96	695	4	4	1	96	697	4	2	0	0.30	0.11	0.14	99.00%	99.25%		
3	30	60	90	680	86	667	2	11	4	88	671	2	9	0	0.33	0.09	0.12	97.79%	98.57%		
4	30	60	90	680	89	664	1	16	0	90	668	0	12	0	0.33	0.09	0.12	97.79%	98.44%		
5	20	50	102	500	100	496	2	4	0	100	498	2	2	0	0.20	0.10	0.12	99.00%	99.34%		
6	50	100	100	1000	99	989	1	11	0	99	992	1	8	0	0.50	0.10	0.14	98.91%	99.18%		
7	50	100	100	1000	98	988	2	11	1	98	991	2	9	0	0.50	0.10	0.14	98.73%	99.00%		
8	50	100	100	1000	98	986	2	13	0	98	986	2	12	1	0.50	0.10	0.14	98.55%	98.55%		
9	50	200	100	1000	97	996	2	3	2	98	999	2	1	0	0.50	0.20	0.23	99.36%	99.73%		
10	50	200	100	1000	99	999	1	1	0	99	999	1	1	0	0.50	0.20	0.23	99.82%	99.82%		
11	50	150	100	1500	97	1490	2	10	1	97	1490	2	10	1	0.50	0.10	0.13	99.19%	99.19%		
12	50	300	100	1500	98	1494	2	6	0	98	1496	2	4	0	0.50	0.20	0.22	99.50%	99.63%		
13	50	300	100	1500	98	1493	2	7	0	98	1495	2	5	0	0.50	0.20	0.22	99.44%	99.56%		
14	50	200	200	2000	195	1990	5	10	0	197	1994	3	6	0	0.25	0.10	0.11	99.32%	99.59%		
15	50	200	200	2000	196	1991	4	9	0	197	1994	3	6	0	0.25	0.10	0.11	99.41%	99.59%		
16	50	200	250	2000	244	1991	6	9	0	246	1995	4	5	0	0.20	0.10	0.11	99.33%	99.60%		
17	50	200	250	2000	244	1990	6	10	0	246	1995	4	5	0	0.20	0.10	0.11	99.29%	99.60%		
18	200	600	1000	3000	991	2989	9	11	0	994	2991	6	9	0	0.20	0.20	0.20	99.50%	99.63%		
19	200	800	1000	4000	992	3987	8	13	0	995	3991	5	9	0	0.20	0.20	0.20	99.58%	99.72%		
20	200	800	1000	4000	991	3988	9	12	0	994	3990	6	10	0	0.20	0.20	0.20	99.58%	99.68%		
21	200	1000	1000	5000	992	4986	8	14	0	993	4990	7	10	0	0.20	0.20	0.20	99.63%	99.72%		
22	200	1000	1000	5000	991	4985	9	15	0	992	4989	8	11	0	0.20	0.20	0.20	99.60%	99.68%		

Table 5-2. Experiment data

Chapter 6

Conclusion and Future work

The speedy growth of spam e-mail has led the growth of spam blocking and filtering techniques. One of the best filtering techniques is Bayesian classification technique. We have presented a simple Chinese words parser to Bayesian Classification to classify Chinese spam e-mails. Our experiments show that with adding Chinese parser to the classifier performs more accurate in e-mail classification.

Although we have achieved a more accurate classification, but there is also a room to make the classifier performs better, that is tokenizing technique and a richer Chinese word database. Methods of training could also perform better accuracy. In our experiment, we only do initial training. It could be much more accurate in classification if we do regular training which makes the corpus database richer. Finally, we hope this thesis has some contribution to fight spam.

References

- [1] Symantec: Spam growth slowing at last, URL: http://www.infoworld.com/article/05/01/12/HNspamslowing_1.html.
- [2] DNS Black List, URL: <http://en.wikipedia.org/wiki/DNSBL>.
- [3] Challenge and Response at SMTP level, URL: <http://jamesthornton.com/writing/challenge-response-at-smtp-level.html>.
- [4] M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz, “A Bayesian approach to filtering junk E-mail”, *In Learning for Text Categorization – Papers from the AAAI Workshop*, pp. 55–62, 1998.
- [5] J. Postel, “Simple Mail Transfer Protocol”, RFC 788, 1981.
- [6] A. Cournane and R. Hunt, “An Analysis of the Tools used for the Generation and Prevention of Spam”, *Computers and Security*, Vol. 23, No. 2, pp. 154-166, 2004.
- [7] Dictionary Attack, URL: http://en.wikipedia.org/wiki/Dictionary_attack.
- [8] P. Mockapetris, “Domain Names: implementation and specification”, RFC 1035, 1987.
- [9] Spam honeypots, URL: http://en.wikipedia.org/wiki/Honeypot#Spam_honeypots
- [10] C. Apte and F. Damerau, “Automated Learning of Decision Rules for Text Categorization”, *ACM Transactions on Information Systems*, Vol. 12, No. 3, pp. 233–251, 1994.
- [11] D. Lewis, “Feature Selection and Feature Extraction for Text Categorization”, *Proceedings of the DARPA Workshop on Speech and Natural Language*, pp. 212–217, 1992.
- [12] I. Dagan, Y. Karov, and D. Roth, “Mistake-Driven Learning in Text Categorization”, *Proceedings of the 2nd Conference on Empirical Methods in Natural Language Processing*, pp. 55–63, 1997.
- [13] F. Sebastiani, “Machine Learning in Automated Text Categorization”, *ACM Computing Surveys (CSUR)*, Vol.34, No.1, pp.1-47, 1999.
- [14] CEDICT (Chinese-English Dictionary) Project, URL: <ftp://ftp.cc.monash.edu.au>

/pub /nihongo/cedict.html.

[15] A Plan for SPAM, URL: <http://www.paulgraham.com/spam.html>.

[16] POPFile, URL: <http://popfile.sourceforge.net>.

[17] POPFile Statistics, URL: http://popfile.sourceforge.net/popfile_stats.html.

[18] Trademark Software, URL: <http://www.tmssoft.com/>.

[19] Microsoft Outlook Express, URL: <http://www.microsoft.com/windows/ie/default.mspx>.

