Application of Linear Classifier on Chinese Spam Filtering

Yongqin Qiu
Beijing Language and Culture University, Beijing, China
Email: qyq0310@blcu.edu.cn

Yan Xu and Dan Li
Beijing Language and Culture University, Beijing, China
Email: {xuy, ld09}@blcu.edu.cn

***Abstract*—Spam is a key problem in electronic communication. Especially in large-scale email systems. Content-based filtering is one mainstream method of combating this threat in its forms, an e-mail filtering system can learn directly from a user's mail set, but the previous Content-based filtering methods are hard to find a balance between efficiency and effectiveness. Such algorithms of text categorization as Naïve Bayes, kNN, Decision Tree and Boosting can be applied in spam filtering. However, the effectiveness of Naïve Bayes is limited and it is not fit for instant feedback learning. Others algorithm such as SVM are more effective but complicated to compute. Because in a real email system a large volume of emails often need to be handled in a short time, efficiency will often be as important as effectiveness when implementing an anti-spam filtering method. So we intend to find a linear classifier to solve this problem, two online linear classifiers: the Perception and Winnow were explored for this task, which are two fast linear classifiers. The training of these two methods is online and mistake driven. Furthermore, they are suitable for feedback. We employ the two methods in three benchmark corpora, including PU1, Ling spam and 2005-Jun, the experiments in public e-mail corpus show an effective result. We conclude that the two online linear classifiers have a state-of-the-art performance for filtering spam, especially for Chinese spam emails.**

***Index Terms*—anti-spam, information filtering, Winnow, Perception, linear classifier.**

I. INTRODUCTION

Electronic communication is increasingly plagued by unwanted or harmful content known as spam. The most well known form of spam is email spam, which remains a major problem for large email systems. A 1997 study [1] found that spare messages constituted approximately 10% of the incoming messages to a corporate network. Since May 2003 the amount of spam exceeded legitimate emails [2]. From China's anti-spam alliance data show that: in July 2009, spam accounted for an average total 89% [3]. Of the total e-mail on average 10 e-mails, only one is a normal e-mail, and the rest are all spam. In October of Symantec's report, China's top ten spam ranked seventh in the country of origin[4], the report from Anti-spam center of ISC showed that the spam in Chinese take a proportion of 21.6%[5]. So we can see that how important for us to find a competitive Method to filter spam, especially Chinese spam. As we all know that the increasing volume of spam has become a serious threat not only to the Internet, but also to society. For the business and educational environment, spam has become a security issue. Spam has gone from just being annoying to being expensive and risky. The enigma is that spam is difficult to define. What is spam to one person is not necessarily spam to another. Fortunately or unfortunately, spam is here to stay and destined to increase its impact around the world. It has become an issue that can no longer be ignored; an issue that needs to be addressed in a multi-layered approach: at the source, on the network, and with the end-user. How to find an efficient spam filtering methods has always been a concerned problem by all sectors of society.

The task of spam filtering can be seen as a special problem of text classification, the text classification's duty is: in assigns under the category system, according to the text content, maps automatically in the category which it assigns. Category system generally by man-power according to application demand structure. Text classification based on content needs supervision, in the other words it needs a certain amount of good classified texts and examples, the system gains the essential information from the training text, and constructs the classifier. However, the spam filtering is not as the same as TC problem totally. According to our analysis, spam has the following three characteristics: first, in a real email system a large volume of emails often need to be handled in a short time; efficiency will often be as important as effectiveness when implementing an anti-spam filtering method. Second, email receivers may pay more attention to the precision than the recalled rate of the filtering system. Because they will read some spam rather than missing one important email. Third, the contents of both the legitimate and spam email classes may change dynamically over time, so the anti-spam filtering profile should be easily updatable to reflect this change in class definition.

For the above characters, an effective and efficient classifier, which can be easily and effectively updated, is

doi:10.4304/jsw.6.1.116-123

the goal for anti-spam filtering. In the numerous text classification algorithms, one kind is the linear classifier, the most major characteristic this kind of approaches is the computation is simple, it can control the computing complexity in linearity range; and its training process is the mistake driven, moreover, the effect is also very good in many situations. This paper is an effort to introduce linear classifier for spam filtering task. Two online linear classifiers: the Perceptron and Winnow are investigated in our experiments. These two algorithms' computation is simple, and their speed is quick, what's more, they are quite stable, and the effect is also good. Moreover, this online mistake-driven training way is suitable for the spam filtration's immediate feedback study. When feedback, in view of each sample, if classified badly, then adjust the weights. Because Perceptron and Winnow algorithms have the above these merits, they will have the practical application value very much in the task of spam filtration, which may be used in the client side and the server to filter emails.

We employ the two methods in three benchmark corpora, including PU1, Ling spam and 2005-Jun. We design a series experiments to compare different feature selection methods, different classifiers and different corpora. The experiments in public e-mail corpus show an effective result and Perceptron and Winnow algorithms are very competitive classifiers for anti-spam filtering tasks, especially for Chinese spam filtering.

The remainder of the paper is organized as follows: Sect. 2 outlines relevant previous research in anti-spam filtering; Sect. 3 introduces the corpora we used. Sect 4 introduces the two linear classifiers used in our study: the Perceptron and Winnow, Sect. 5 describes our experiments, including details of the test collections and experimental results, and finally our conclusions and future work are given in Sect. 6.

II. Related Work

The Internet is gradually becoming an integral part of everyday life. Internet usage is expected to continue growing and e-mail has become a powerful tool intended for idea and information exchange, as well as for users' commercial and social lives. Along with the growth of the Internet and e-mail, there has been a dramatic growth in spam in recent years [6]. The majority of spam solutions deal with the flood of spam. However, it is amazing that despite the increasing development of anti-spam services and technologies, the number of spam messages continues to increase rapidly.

Spam filtering is able to control the problem in a variety of ways. Identification and spam removal from the e-mail delivery system allows end-users to regain a useful means of communication. Anti-spam filtering is a popular problem in academic circles. Several proposals have been applied to filter spam emails, include technical, economic and legal methods [9]. Currently, machine learning for spam classification is an important research issue at present. The success of machine learning techniques in text categorization has led researchers to explore learning algorithms in spam filtering[7][8]. The mainly machine learning methods can be divided into rule-based and statistical-based methods. The former obtains the people understandable explicit rule; the latter often promotes the result through some kind of computation expression. Essentially, the statistical-based method may regard as one kind of exceptional case of rule-based methods, but the rules obtained by the statistical-based method are the implicit rules which are not understood easily by human. No matter rule-based method or statistics method, the whole system will experiences from the training to the filtration process. It concludes the corresponding spam rule (including explicit rule or implicit rule) through training the existing training set, and then applies the rule in the new mail determination, which will possibly add the man-machine interaction process in the actual system.

In fact the rule-based method's study process is the process of summarizing and it summaries regular information to format rules through examining each training sample. The Representative methods are Ripper [10], PART [11], Decision tree [12][13], and Rough Sets[14]. The rule-based method's principal advantage is it may produce the rules which are understandable. The shortcoming is that pure rule-based methods have not achieved high performance because spam emails cannot easily be covered by rules.

Statistical-based methods have proven more successful, and are generally adopted in mainstream work [15]. Bayesian classifiers are the most widely used method in this field. The representative figure is Sahami et al [16] and Greek scholar Androutsopoulos [17]. Sahami et al trained a Naïve Bayesian classifier on manually categorized legitimate and spare messages, reporting impressive performance on unseen messages. Following this work, Androutsopoulos et al. extended the Naïve Bayes (NB) filter proposed by Sahami et al. 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 abovementioned methods clearly outperform a typical keyword-based filter. In addition to the Naı¨ve Bayes model, other Bayesian models have also been considered for anti-spam filtering. Androutsopoulos et al. [18]introduced a flexible Bayes model which is one kind of model for continuous valued features. Their experiments showed that the flexible Bayes model outperforms Naı¨ve Bayes.

Other machine learning methods, including Support Vector Machine (SVM) [19], Rocchio, kNN and Boosting [20], have also been applied in anti-spam filtering. To the best of our knowledge, few online linear classifiers have been explored in the area of spam filtering, and few experiments are carried on Chinese corpora. These reasons motivate us to explore the application of online linear classifiers to the task of Chinese anti-spam filtering.

III. CORPORA

Three publicly available benchmark corpora are used for our anti-spam filtering research in this paper, the 2005-Jun data set, and PU1 and Ling-Spam collections. The Chinese corpora we used is called 2005-Jun data set from China Education and Research Network Computer Emergency Response Team (CCERT), which is a non-profit organization who provides computer security related incident response service for people and organizations all over China. The 2005-Jun data set is the widest used corpora in China.

The 2005-Jun data set from CCERT contains 25088 spams and 9272 ham. A ham message is composed by a post (from forum) and a raw header of a ham messages. The raw data set removed all html tags in the body part and kept only the plain text part of each message, but it remained the 'Content-type' header unchanged (it may be useful),but in our experiment, we removed the 'Content-type' header also. In another word, the Chinese corpora we used in our experiment are pure composed by Chinese characters.

The PU1 corpus consists of 481 spam messages and 618 legitimate messages, which are all real private personal emails. Because of privacy problems, these messages have been transformed to an ''encrypted'' version prior to distribution, in which the words are encoded with integers. Unfortunately, this transformation also limits the potential to explore language based filtering techniques.

The Ling-Spam corpus was built up in a different way. The 481 spam messages are the same as the spam messages of PU1, but the 2,412 legitimate messages were collected from a freely accessible newsgroup, Linguist list. Thus Ling-Spam has no privacy problem and is not encrypted. Because the legitimate messages of Ling-Spam are more domain topic-specific than those of PU1, Ling-Spam is more appropriate for spam filtering of a topic-specific newsgroup.[17]

IV. THE PERCEPTRON[21] AND WINNOW [22]

According to whether the decision surface in the feature space is a hyperplane or not, classifiers can be divided into two categories: linear classifiers and nonlinear classifiers. The most widely used classifiers are linear models. A linear classifier represents each class c_i with a class weight vector w_i and a threshold θ_i (i = 1, 2, 3... |C|), here |C| equals 2 to represent the example is spam or not. If a document d makes internal product with w_i satisfies $w_i \bullet d > \theta_i$, then it belongs to class c_1 else belongs to c_2, So linear classifiers try to find suitable values for w and θ that correctly separate all the training examples. Obviously, if they exist, the separate hyper plane is $w \bullet x_i - \theta = 0$. From Fig. 1, we can also see that there is not a single discrimination hyper plane.

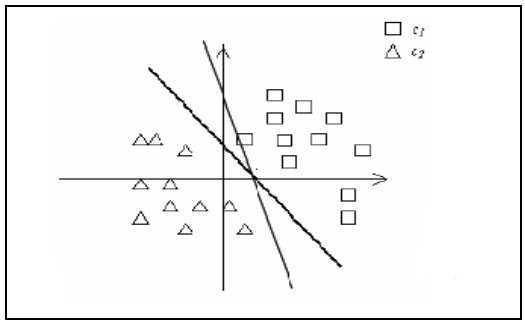


Figure 1. Linearly separable examples (the solid line represents the discrimination hyper plane) of two classes: c1 and c2

The Perceptron algorithm is an example of a weight-update linear classifier which is proposed by Rosenblatt [21] in the late 1950s and early 1960s. Winnow is an alternate algorithm introduced by Littlestone[23] in the late 1980s. The basic idea of both of these classifiers is to update the vector w driven by classification errors and both of them are the kind of linear classifier, which is the most widely used classifiers, follows are the detailed description of two algorithms.

A. The Perceptron Algorithm

One of the oldest algorithms used in machine learning is an online algorithm for learning a linear threshold function called the Perceptron Algorithm. For simplicity, we'll use a threshold of 0, so we're looking at learning functions like:

$$w_1 x_1 + w_2 x_2 + \ldots w_n x_n > 0 .$$

We can simulate a nonzero threshold with a "dummy" input x_0 that is always 1, so this can be done without loss of generality. The guarantee we'll show for the Perceptron Algorithm is the following:

Theorem 1: Let S be a sequence of labeled examples consistent with a linear threshold function $w^* \cdot x > 0$, where w^* a unit-length vector. Then the number of mistakes M on S made by the online Perceptron algorithm is at most $(1/\gamma)^2$, where

$$\gamma = \min_{x \in S} \frac{\left| w^* \cdot x \right|}{\left\| x \right\|}$$

(I.e., if we scale examples to have Euclidean length 1, then γ is the minimum distance of any example to the plane $w^{*.} x = 0$.)

The parameter "γ" is often called the "margin" of $w^{*.}$ (or more formally, the L2 margin. Because we are scaling by the L2 lengths of the target and examples). Another way to view the quantity $w^{*.} \cdot x / \left\| x \right\|$ is that it is the cosine of the angle between x and $w^{*.}$, so we will also use $\cos((w^{*.}, x))$ for it.

The Perceptron Algorithm:

1. Start with the all-zeroes weight vector $w_1 = 0$, and initialize t to 1. Also let's auto-matically scale all examples **x** to have (Euclidean) length 1, since this doesn't affect which side of the plane they are on.

2. Given example x, predict positive if $w_t \cdot x > 0$.

3. On a mistake, update as follows:

• Mistake on positive: $w_{t+1} \leftarrow w_t + x$.

• Mistake on negative: $w_{t+1} \leftarrow w_t - x$.

$t \leftarrow t+1$.

So, this seems reasonable. If we make a mistake on a positive x we get $w_{t+1} \cdot x = (w_t + x) \cdot x = w_t \cdot x + 1$, and similarly if we make a mistake on a negative x we have $w_{t+1} \cdot x = (w_t - x) \cdot x = w_t \cdot x - 1$. So, in both cases we move closer (by 1) to the value we wanted.

Proof of Theorem 1. We're going to look at the magic quantities $w_t \cdot w^*$ and $\|w_t\|$.

Claim 1: $w_t \cdot w^* \geq w_t \cdot w^* + \gamma$. That is, every time we make a mistake, the dot-product of our weight vector with the target increases by at least γ .

Proof: if x was a positive example, then we get $w_{t+1} \cdot w^* = (w_t + x) \cdot w^* = w_t \cdot w^* + x \cdot w^* \geq w_t \cdot w^* + \gamma$ (by definition of γ). Similarly, if x was a negative example, we get $(w_t - x) \cdot w^* = w_t \cdot w^* - x \cdot w^* \geq w_t \cdot w^* + \gamma$

Claim 2: $\|w_{t+1}\|^2 \leq \|w_t\|^2 + 1$. That is, every time we make a mistake, the length squared of our weight vector increases by at most 1.

Proof: if x was a positive example, we get $\|w_{t+1}\|^2 = \|w_t\|^2 + 2w_t \cdot x + \|x\|^2$.

This is less than $\|w_t\|^2 + 1$ because $w_t \cdot x$ is negative (remember, we made a mistake on x). Same thing (flipping signs) if x was negative but we predicted positive.

Claim 1 implies that after M mistakes, $w_{M+1} \cdot w^* \geq \gamma M$. On the other hand, Claim 2 implies that after M mistakes, $\|w_{M+1}\| \leq \sqrt{M}$. Now, all we need to do is use the fact that $w_t \cdot w^* \leq \|w_t\|$, since w is a unit vector. So, this means we must have $\gamma M \leq \sqrt{M}$, and thus $M \leq 1/\gamma^2$.

We can see Perceptron algorithm can update the weight by adding a positive value to each of them if the examples which are badly classified.

B. The Winnow Algorithm

Like the Perceptron algorithm, The Winnow Algorithm learns linear threshold hypotheses. The algorithm and its analysis are specialized to inputs in $\{0,1\}^n$, that is, when the features are binary.

We think of w as the hypothesis while $1 < \alpha \in R$ and $0 < \theta \in R$ are parameters of the algorithm.

1 $w \leftarrow 1^n$

2 Repeat

3 get example ($x^{(i)}, y^{(i)}$),
where $x^{(i)} \in \{0,1\}^n, y^{(i)} \in \{-1,1\}$

4 classify $\hat{y} = \oplus \Leftrightarrow w \cdot x^{(i)} \geq \theta$

5 if $\hat{y} \neq y^{(i)}$, update w by

$$\forall k = 1,...,n : w_k \leftarrow \begin{cases} w_k \alpha^{x_k^{(i)}{}_k} & if \ \ y^{(i)} = \oplus \\ w_k / \alpha^{x_k^{(i)}} & if \ \ y^{(i)} = \otimes \end{cases}$$

Notice that if a mistake occurs on $\oplus$ -example, the algorithm considers every attribute $x_k^{(i)}$ and the corresponding weight w_k :

$$if \ \ x_k^{(i)} = 1 \Rightarrow w_k \leftarrow w_k \alpha$$

$$if \ \ x_k^{(i)} = 1 \Rightarrow w_k \leftarrow w_k$$

A similar property holds for negative examples.

C. The comparison of two classifiers

The difference between the winnow and Perceptron is in how they increment the weights. The Perceptron algorithm adds a positive constant to each of them, whereas Winnow multiplies each of them by a constant that is larger than one. Similarly, if the prediction was right, the parameters will not be changed, else the weights of w_i with are decremented either by subtracting a constant or dividing by a constant. We call choosing these parameters tuning to adjust the performance of the algorithms.

Because the two classifiers can both update the vector w driven by classification errors, which is why they are called error-driven [24] methods.

V. EXPERIMENTAL RESULTS

A. Feature selection

Training set contains a large vocabulary, and if each of it is considered as a feature then there will be a series of problems: fist of all, the dimension of the vectors will be enormous, which would add a huge burden to the calculation, while occupying a huge storage room and slowing down the processing speed. Secondly, a large part of the vocabulary actually has nothing to do with the classes, thus it is not helpful for the classification. We, therefore, have to decrease the dimensions of the vectors, and choose only the representative ones as the feature.

So firstly we pre-process the text by eliminating those "stop words" that are not essential to the classification.

Then we sequence the other words with feature selection, and the words listed on the top will be the features. Here are the feature selection methods we use:

a) Mutual Information

Mutual Information, the abbreviation is MI. It is a useful measure of information content in Information Theory. Its definition is:

$$MI(t) = \sum_{i=1}^{|c|} P(c_i) \log \frac{P(t \mid |c_i)}{p(t)}$$

$P(c_i)$ represents the probability of class i document appearing in the training document set. $P(t)$ represents the probability of term t appearing in the training document set. The larger the MI, the more likely that word and class will co-occur.

b) χ^2 statistic

$$\chi^2(t, c_i) = \frac{N \times (AD - BC)^2}{(A+C) \times (B+D) \times (A+B) \times (C+D)}$$

$$\chi^2_{avg}(t) = \sum_{i=1}^{|c|} P(c_i) \chi^2(t, c_i)$$

A、B、C、D represent quantity of document,showing in the following table, $N = A + B + C + D$

	type c_i	not type c_i
t included	A	B
t not include	C	D

χ^2 statistic evaluates the absence degree of measurement word in regard to the genre independence, the bigger the χ^2 statistic, the less independent but more relevant is it. χ^2_{avg} stands for the χ^2 statistic on average of all the genres.

c) Relative Entropy

$$CE(t) = \sum_{i=1}^{|c|} P(c_i \mid t) \log \frac{P(c_i \mid t)}{P(c_i)}$$

It is also known as the Kullback-Leibler divergence, which reflects the divergence between the probability distribution of text genres and that of which with the appearance of a certain word, the bigger the divergence, the more influential such a word is towards the text classification.

In our experiments each document d is represented with a binary vector($x_1\ x_2 \ldots x_k$), $x_i = 1$ or 0 $(0 \leqslant i \leqslant k)$which respectively means the ith feature is present or absent in this document, k is the number of retained features after feature selection. Those features that actually occur in the document ($x_i = 1$) are called active features of this document.

B. Evaluation measures

Four traditional measures in text classification research are used for evaluation: recall, precision, F1 measure and accuracy. For the following definitions let there be: a total of a messages, including a_1 spam and a_2 legitimate emails; b messages of which are judged as spam by an anti-spam filtering system, and among them, b_1 messages are really spam; and a total of a – b messages judged as legitimate, among which b_2 messages are really legitimate. The evaluation measures of this system are defined as follows:

$$recall = \frac{b_1}{a_1}$$

$$percision = \frac{b_1}{b}$$

$$F1 = \frac{2 \times recall \times precision}{recall + precision}$$

$$accuracy = \frac{b_1 + b_2}{a}$$

F1 combines recall and precision into a single measure and is frequently used in TC. Accuracy is widely used in many previous anti-spam filtering studies. However, we believe that it is greatly affected by the class distribution. For example, if the number of legitimate messages is much bigger than that of spam messages, the accuracy will not be greatly affected by spam messages. In other words, the accuracy may be very high even though the spam filtering performance is very poor due to the skew distribution. So the F1 measure is mainly used in our experiments, but accuracy results are included for comparison with existing work.

C. Experiment result

We performed four sets of experiments using 2005-Jun data set (“Chinese” in the Figures), the PU1 and Ling-Spam test corpora. The parameters of the classifiers were set to the same values for each test corpora for each classifier. All of the experiments were performed using tenfold cross validation, where each corpora was equally divided into ten parts, nine of them were used for training, while the other part was used for testing.

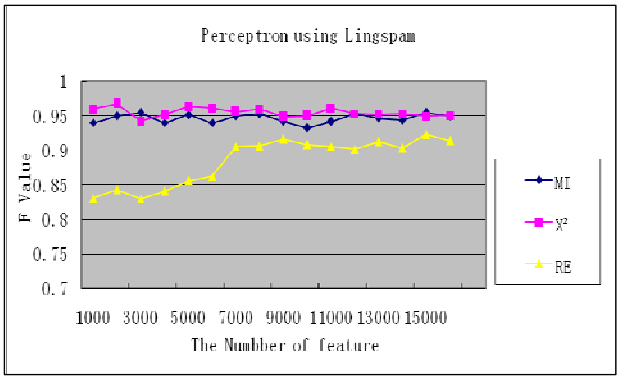
In the first set different feature selection methods were compared. In the second set different corpora were compared under the same classifier, here is Winnow and Perceptron respectively. The relationship between the number of features and the performance was also investigated. The third set of experiments focused on the performance in the three classifier in the same corpora. Three methods were tested in 2005-Jun data set,

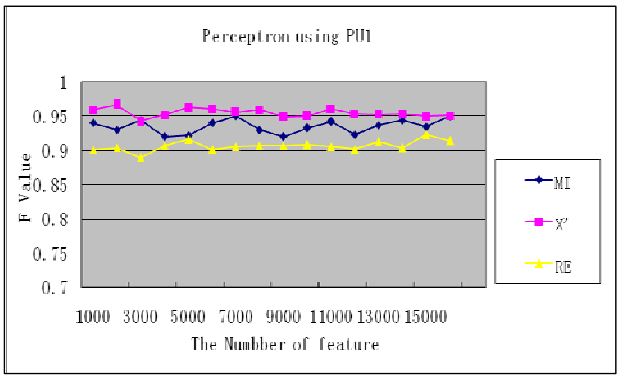
Lingspam and PU1 respectively. In the last set of experiments, we list all composite indicators in the Table1 to indicate the recalled rate, accuracy and precision in 3 corpora with 3 methods.

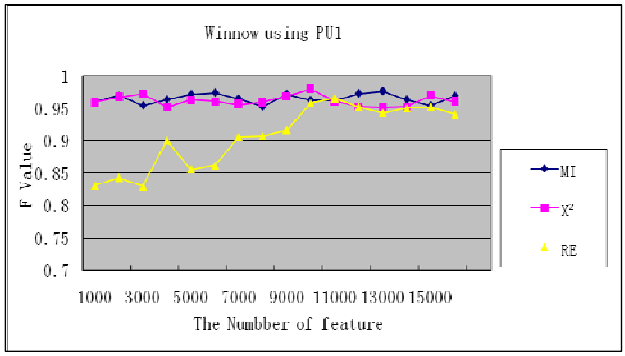
1) Experiments with different feature selection

Our first experiments used different feature selection approaches for both the Perceptron and Winnow in three corpora. Three feature selection approaches were applied: MI, χ^2 statistic and RE. The top-k words with the highest MI, χ^2 statistic and RE were selected as the final features. We then changed the number of features k. The range variations were divided into two parts. In the first part, k varied from 10 to 200 by 10, and then from 200 to 800 by 50; in the second part, it varied from 500 to 15,000 by 500.

From Figure 2. we can see that for both the Perceptron and Winnow, MI and χ^2 statistic perform similarly well, and clearly much better than RE. This finding is similar to Yang's observation that MI and χ^2 statistic are two top-performing feature selection methods when using kNN and LLSF classifiers [25]. A reasonable explanation for this observation that MI and χ^2 statistic outperform RE, is that RE biases toward favoring those features which are good at representing spam class, whereas features good for representing the legitimate class are also very useful in the Perceptron and Winnow classifiers because they allow negative weights.







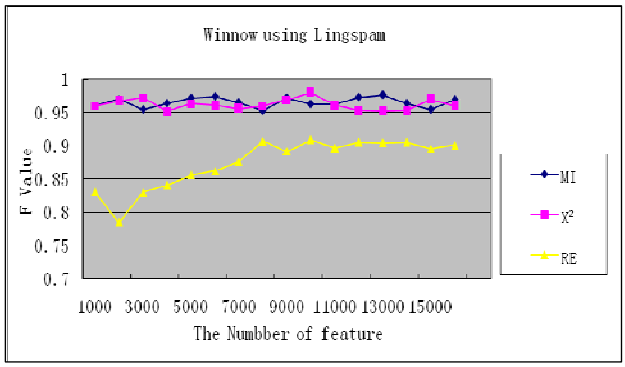


Figure 2. Results of different feature selection

2) Different performance of corpora in the same method

In the second set of experiments we investigated the performance of the three corpora (Chinese, Lingspam and PU1) when using Perceptron and Winnow respectively where the number of feature varied from 0 to 200 by 50.

From the Figure 3. and Figure 4. we can see that when the number of the feature selected increases, the performance of three corpora also increases. But that the F1 performance only increases slowly compared to the increase in the size of the feature numbers. When the number of feature is below 200, PU1 outperformed than the other 2 corpora. However, when the feature numbers is above 200, Chinese corpora has the best performance no matter using the Winnow classifier or the Percetron.

3) Different classified methods in the same corpora

In the third set of experiments, we investigated the performance of the three classified methods in the some corpora. we can see that when the number of features is small, Naı¨ve Bayes performs comparably to the Perceptron and Winnow filters, but that when the number of features increases, the performance of Naı¨ve Bayes falls, while the performance of the Perceptron and Winnow increase and then, as shown previously (see Figure 5.), become very stable as the number of features increases. A reasonable explanation for this behavior is that anti-spam filtering is a task with many irrelevant features, but Naı¨ve Bayes cannot perform very well if many irrelevant features are considered to estimate the probabilities. However, from Figure 5. , we can see clearly that the Perceptron and Winnow can handle such conditions very well.

Here we can also conclude that, in the Chinese corpora,

when the feature number is below 300, the Perceptron method outperformed a little better than Winnow, however, with the number of the feature increased, the F1 value remains stable, which means the two methods reaches the same level of effective.

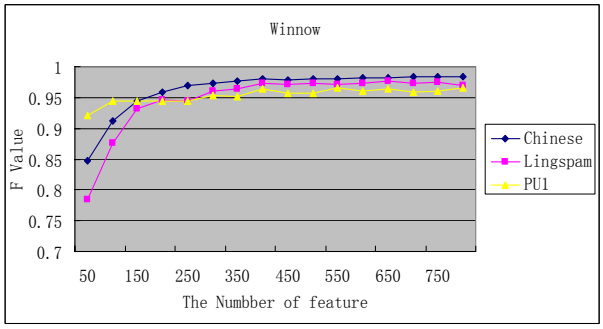


Figure 3. Winnow Method on three corpora

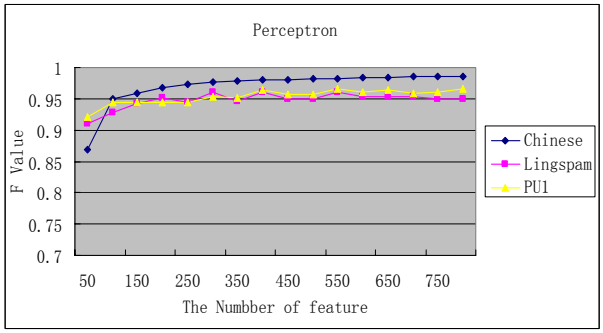


Figure 4. Perceptron Method on three corpora

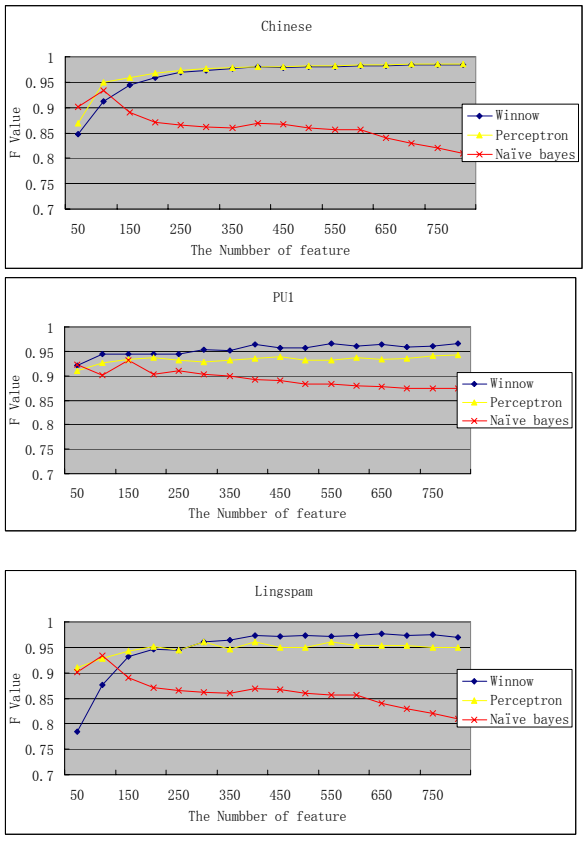


Figure 5. Three different classifier methods on the same corpora

TABLE I. TABLE1 THE RESULTS OF THE PERCEPTRON, WINNOW AND NAÏVE BAYES IN THREE CORPORA.

corpora	Classifier	Index			
		Recall	*precision*	*F1*	*Accuracy*
2005-Jun	Winnow	98.60	97.56	98.07	97.33
	Perceptron	97.52	98.93	98.22	97.56
	Naïve bayes	88.8	97.7	94.88	94.2
PU1	Winnow	97.09	98.17	97.57	97.91
	Perceptron	97.29	97.55	97.40	97.72
	Naïve bayes	87.5	98.3	92.98	93.9
Lingspam	Winnow	97.72	98.14	97.92	99.31
	Perceptron	94.60	98.75	96.58	98.89
	Naïve bayes	87.9	97.9	92.53	93.9

4) All the composite indicators of the three methods in three corpora

From TABLE I, we can see the composite indicators clearly, the Winnow algorithm shows the best performance in Accuracy. No matter in English corpora or Chinese corpora, two linear classifiers show a better efficiency and effectiveness than classic Naïve bayes algorithm. Finally, every index in Chinese corpora all outperforms than that in English. So we can see the advantage of the two linear classifier than other classifier, they outperformance than the classic classifiers and are more suitable to finish Chinese spam filtering task.

VI. CONCLUSION AND FUTURE WORK

Though our experiment data above, we can see that the two classifiers outperformed than other previous filtering approaches in filtering performance, computation complexity and some other indexes. So we concluded as follows:

1. The experimental results show that both the Perceptron and Winnow are more suitable to handle the Chinese corpora. Obviously, when compared with the two set of English corpora, the result on Chinese corpora is much better.
2. It was also shown that theoretical and implementation computational complexities of these two classifiers are very low, and they can very easily be adaptively updated. Especially on the Chinese corpora.
3. Experimental results show that these two classifiers outperform existing methods for spam filtering and they are very fast and suitable for real use.

Certainly, we also see much future work to do, for example, though we got a great performance in Chinese corpora, we can't give a reasonable for this phenomena, maybe there is some special characteristics in Chinese language itself we didn't find yet, and another problem is the contents of both the legitimate and spam email classes may change dynamically over time in a real spam filtering system, so the anti-spam filtering profile should be easily updatable to reflect this change in class

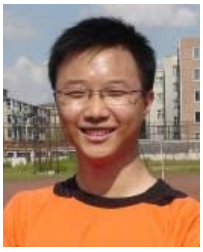
definition. We may consider building dynamical corpora for filtering task. Other work is currently focused on developing a benchmark corpora for Chinese spam email filtering.

ACKNOWLEDGMENT

This paper is supported by Ministry of Education of the People's Republic of China No: 109028, National Natural Science Foundation of China. No: 60873166 we would like to thank the reviewers for providing valuable comments and advices and Education science Foundation of Beijing. No: AHA09110

REFERENCES

[1]. F. Cranor and B.A. LaMacchia. Spare! Communications of the ACM, 41(8):74--83, 1998.
[2]. Vaughan Nichols SJ (2003) Saving private e-mail. IEEE Spectr 40(8):40–44
[3]. China's anti-spam alliance http://anti-spam.org.cn/ 2009
[4]. Symantec's report http://www.symantec.com/business/theme.jsp?themeid=threatreport,2009
[5]. Anti-Spam Committee is an affiliated organization of ISC http://www.anti-spam.cn/ ,2009
[6]. Salem,E. 2004, Interview with Enrique Salem (CEO of Brightmail) conducted on 23rd Feb, 2004
[7]. Joachims, T. 1998, Text categorization with Support Vector Machines: Learning with many relevant features. *In Proceedings of the 10th European Conference on Machine Learning, Lecture Notes in Computer Science*, n. 1398 Springer Verlag, Heidelberg, Germany, 137–142..
[8]. Sebastiani, F. 2002 Machine learning in automated text categorization. *ACM Computing Surveys* 34, 1, pp.1–47.
[9]. Hoffman P, Crocker D (1998) Unsolicited bulk email: mechanisms for control. Technical Report Report UBESOL, IMCR-008, Internet Mail Consortium
[10]. Cohen W (1995) Fast effective rule induction. In: Machine learning: Proceedings of the 12th international conference, pp 115–123
[11]. Drucker H, Wu D, Vapnik VN (1999) Support vector machines for spam categorization. IEEE Trans Neural Netw20(5):1048–105
[12]. Hidalgo JMG (2002) Evaluating cost-sensitive unsolicited bulk email categorization. In: Proceedings of ACM symposium on applied computing, pp 615–620
[13]. M. Desouza, J. Fit Rald, C. Kemp and G. Truong, A DecisionTree basedspam FilteringAgent[EB]. from http://www.cs.mu.oz.au/481/2001-projects/gntr/index.html, 2001
[14]. Yang L, Xiaoping D, Ping L, Zhihui H, Chen G, Huanlin L (2002) Intelligently analyzing and filtering spam emails based on rough set. In: Proceedings of 12th Chinese computer society conference on network and data communication, pp211–215
[15]. Bin Wang, Gareth Jones and Wenfeng Pan, Using Online Linear Classifiers to Filter Spam Emails, Pattern Analysis and Application, 2006, Vol. 9, No. 4:339-351.
[16]. Sahami M, Dumais S, Heckerman D, Horvitz E (1998) A Bayesian approach to filtering junk e-mail. In: Proceedings of AAAI workshop on learning for text categorization, pp 55–62
[17]. Androutsopoulos I, Koutsias J, Chandrinos KV, Spyropoulos CD (2000) An experimental comparison of Naive Bayesian and keyword-based anti-spam filtering with encrypted personal e-mail messages. In: Proceedings of the 23rd annual international ACM SIGIR conference on research and development in information retrieval, pp 160–167
[18]. Androutsopoulos I, Paliouras G, Michelakis E (2004) Learning to filter unsolicited commercial e-mail. Technical report 2004/2, NCSR ''Demokritos''
[19]. Drucker H, Wu D, Vapnik VN (1999) Support vector machines for spam categorization. IEEE Trans Neural Netw 20(5):1048–1054
[20]. Carreras X, Marquez L (2001) Boosting trees for anti-spam email filtering. In: Proceedings of European conference on recent advances in NLP, pp 58–64
[21]. Rosenblatt E (1988) The perceptron: a probabilistic model for information storage and organization in the brain. Psych Rev 65(1958):386–407; reprinted in: Neurocomputing (MIT Press, Cambridge, 1988)
[22]. Kivinen J, Warmuth MK, Auer P (1997) The Perceptron algorithm versus Winnow: linear versus logarithmic mistake bounds when few input variables are relevant. Artif Intell 97(1–2):325–343
[23]. Littlestone N (1988) Learning quickly when irrelevant attributes abound: a new linear-threshold algorithm. Mach Learn 2(4):285–318
[24]. Dagan I, Karov Y, Roth D (1997) Mistake-driven learning in text categorization. In: Proceedings of the 2nd conference on empirical methods in natural language processing, pp 55–63
[25]. Yang Y, Pedersen JP (1997) A comparative study on feature selection in text categorization. In: Proceedings of the 14th international conference on machine learning, pp 412–420



Yongqin Qiu computer science and technology application, 1st year graduate student, Beijing Language and Culture University, majored in Computer Engineering Department, Jilin University for undergraduate study.

Yan Xu received the M.S. degree in computer science (1996) and the Ph.D. degree in computer science (2004) from Beijing University of Aeronautics & Astronautics, Beijing, China. She is an associate professor in Institute of Computing Technology, Chinese Academy of Sciences now. Her research interests include date mining, information retrieval.

Dan Li computer science and technology application, 1st year graduate student, Beijing Language and Culture University, majored in information management and information system, Beijing Language and Culture University, for undergraduate study.