Computing and Informatics, Vol. 33, 2014, 1400–1427

ACTIVE MULTI-FIELD LEARNING FOR SPAM FILTERING

Wuying Liu, Lin Wang*, Mianzhu Yi, Nan Xie

School of Foreign Language, Linyi University
276005 Linyi, Shandong, China
e-mail: wyliu@lyu.edu.cn, wanglin@nudt.edu.cn,
mianzhuyi@gmail.com, xienan@lyu.edu.cn

Abstract. Ubiquitous spam messages cause a serious waste of time and resources. This paper addresses the practical spam filtering problem, and proposes a universal approach to fight with various spam messages. The proposed active multi-field learning approach is based on: 1) It is cost-sensitive to obtain a label for a real-world spam filter, which suggests an active learning idea; and 2) Different messages often have a similar multi-field text structure, which suggests a multi-field learning idea. The multi-field learning framework combines multiple results predicted from field classifiers by a novel compound weight, and each field classifier calculates the arithmetical average of multiple conditional probabilities predicted from feature strings according to a data structure of string-frequency index. Comparing the current variance of field classifying results with the historical variance, the active learner evaluates the classifying confidence and regards the more uncertain message as the more informative sample for which to request a label. The experimental results show that the proposed approach can achieve the state-of-the-art performance at greatly reduced label requirements both in email spam filtering and short text spam filtering. Our active multi-field learning performance, the standard (1-ROCA) % measurement, even exceeds the full feedback performance of some advanced individual classifying algorithm.

Keywords: Spam filtering, active multi-field learning, email spam, short message service spam, TREC spam track

Mathematics Subject Classification 2010: 68T50, 68Q32, 62H30, 68T30

* corresponding Author

1 INTRODUCTION

Electronic junk [1], predicted 30 years ago, is becoming spam infoglut [2] in recent years. Spam is the bulk, promotional and unsolicited message. The rapid development of network communicating and mobile computing has made the spam ubiquitous, such as email spam, short message service (SMS) spam, instant messaging (IM) spam, microblogging spam, etc. Spam explosion has made it critical to develop a practical spam filter that facilitates various spam messages filtering.

Messages may be divided into two kinds by the text length: length-unlimited text and short text. Email is a typical length-unlimited text, and the short text normally includes IM message, SMS message, microblogging post, etc. Previous researches tended to investigate email spam filtering [3] and short text spam filtering [4] separately. We find that different messages have a similar multi-field text structure, for instance: 1) Email includes five natural text fields: Header, From, ToCcBcc, Subject, and Body; and 2) SMS message includes three natural text fields: FromNumber, ToNumbers, and Body. This common feature of multi-field text structure brings an opportunity to develop a universal online text classification (TC) approach to deal with various spam messages.

Statistical TC algorithms have been widely used to defeat spam messages, which have been largely successful when given large-scale, fully labeled data sets. However, in practice it is cost-sensitive to obtain a label for a real-world spam filter. Especially, it is unreasonable to require a user labeling every message in time: such a requirement defeats the full feedback filter. Active learning approaches have been developed to reduce labeling cost by identifying informative samples for which to request labels. Previous researches paid more attention to sampling methods of active learning [5]. This paper tries to add some historical information to improve sampling methods, and further defines the practical spam filtering as an active multi-field learning problem for online binary TC.

In the rest of this paper, we review related work in email spam filtering and short text spam filtering. We describe supervised learning, multi-field learning, and active learning for online binary TC. We then investigate the multi-field text structure of various messages, and propose an active multi-field learning approach, including a historical-variance-based active sampling method, a compound-weight-based linear combining method, and a light field TC algorithm. We find strong results with our approach, greatly reducing the number of labels needed to achieve strong classification performance both in email and short text spam filtering. These results even exceed the full feedback performance of some advanced individual classifying algorithms. We conclude with a discussion on the implication of these results for various real-world spam messages filtering.

2 RELATED WORK

Email spam filtering has been widely investigated, and many robust filtering approaches have been proposed [6, 7, 8, 9]. Short text spam is now prevalent in the world, and its filtering technique has also been focused on [10, 11].

Spam filtering was ideally defined as a supervised learning problem for online binary TC, which was simulated as a full feedback task in the TREC spam track [12]. At the beginning of the task, the filter has no labeled message. Messages are processed in their chronological sequence and the user feedback (category label) for each message is communicated to the filter immediately following classification.

Many online supervised binary TC algorithms have been proposed for spam filtering till now. For instance:

1. Based on the vector space model (VSM), the online Bayesian algorithm [6] uses the joint probabilities of words and categories to estimate the probabilities of categories for a given message;
2. The relaxed online support vector machines (SVMs) algorithm [7] relaxes the maximum margin requirement and produces nearly equivalent results, which has gained several best results at the TREC 2007 spam track; and
3. The online fusion of dynamic Markov compression (DMC) and logistic regression on character 4-grams algorithm [8] is the winner on the full feedback task of trec07p data set.

Except individual algorithms, ensemble approaches are also effective. Our researches [9, 10, 14, 15] have proved that the multi-field structural feature of messages is very useful, and our multi-field learning framework, an ensemble learning structure, can improve the overall performance (1-ROCA) % [16] of many individual TC algorithms, such as the online Bayesian algorithm and the relaxed online SVMs algorithm. Within the multi-field learning framework, the improvement of the overall performance can be explained in two main reasons:

1. The multi-field learning framework can reduce the disturbances among text features from different fields; and
2. The multi-field ensemble learning has statistical, computational and representational advantages [17].

The effectiveness of previous spam filtering approaches mainly depends on large-scale, fully labeled data sets. In practice it may be costly to acquire user feedbacks. Active learning approaches can reduce labeling cost by identifying informative samples. It has been proved that only a small portion of a large unlabeled data set may need to be labeled for training an active learning classifier that can achieve high classification performance [18, 19, 20]. Thus, practical spam filtering is reasonably defined as an active learning problem for online binary TC, which is simulated as an online active learning task in the TREC spam track [21]. The difference between active filter and full feedback filter is acquiring methods of user feedback. The active

filter has a preset quota parameter of messages for which feedback can be requested immediately until the quota is exhausted. When each message is classified, the active filter must decide to request or decline feedback for the message. Declining feedback for some uninformative messages is in order to preserve quota so as to be able to request feedback for later informative messages.

Though the short text spam is relatively new electronic junk, it has already spread over the world. Previous research showed that it was difficult to extract effective text features of short messages [10]. Another challenge is the lack of large-scale actual labeled short messages data sets [11]. We find that most short messages have multi-field text structure, similar with multi-field email structure, which can be used in multi-field learning to reduce the effect from the lack of text features.

This paper tries to integrate the multi-field learning and the active learning to solve the practical spam filtering problem. The proposed active multi-field learning approach is based on the common feature of multi-field text structure, and universal to various spam messages filtering. Our main contributions are the historical-variance-based active learning method and the compound-weight-based linear combining method within the active multi-field learning framework.

3 ACTIVE MULTI-FIELD LEARNING

Email, SMS message, and all the other messages include at least three natural text fields: information source field, information target field, and body text field. Applying the divide-and-conquer strategy, active multi-field learning breaks a complex TC problem into multiple simple sub-problems according to the structural feature of messages. Each sub-problem may have its own suitable text features, and the combined multiple field classifying results will be expected to improve the final classification accuracy. The multiple field classifying results may also help the active learner to make a decision.

3.1 Framework

Figure 1 shows the active multi-field learning (AMFL) framework for the binary TC of messages, including a splitter, several field classifiers, a combiner, and an active learner. The learning process of the AMFL framework includes:

1. The splitter analyzes a message and splits it into several text fields.
2. Each field classifier is obligated to process its corresponding text field, and outputs a result. The extracting of text features, the training or updating of field TC model, and the predicting of field classifying result are only localized in the related text fields for each field classifier.
3. The combiner combines multiple results from the field classifiers to form the final result.
4. The active learner evaluates the classifying confidence according to the results from the field classifiers and makes a decision to request a label or not.

5. If the active learner decides to request a label, then it will send the feedback to the field classifiers, and each field classifier will incrementally update itself immediately.

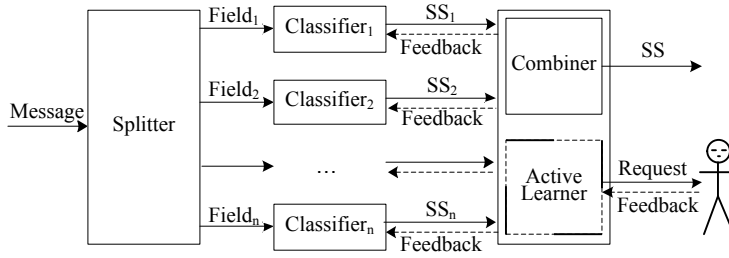


Figure 1. Active multi-field learning framework

Within the AMFL framework, each field classifying result is not the traditional binary output but a spamminess score (SS), which is a real number reflecting the likelihood that the classified document is spam. The classical Bayesian conditional probability $P(spam|docx)$, showed in Equation (1), reflects this likelihood.

$$P(spam|doc) = \frac{P(doc|spam)P(spam)}{P(doc|spam)P(spam) + P(doc|ham)P(ham)}. \tag{1}$$

If the $P(spam|doc)$ is applied to estimate the SS, then the SS threshold T, showed in Equation (2), can be used to make a binary judgment; but the value of SS and threshold is affected by the number distribution of labeled spams and hams, and the number of two categories labeled data is not fixed during the time of online filtering.

$$T = \frac{P(spam)}{P(spam) + P(ham)}. \tag{2}$$

In order to eliminate this number distribution influence and make the same SS value has the equivalent likelihood during the whole online filtering, this paper scales up the number of two categories labeled data to make $P(spam) = P(ham)$, and uses the scaled Bayesian conditional probability $P(spam|doc)$, showed in Equation (3), to represent the SS, then the SS threshold $T = 0.5$ will be a fixed point.

$$P(spam|doc) = \frac{P(doc|spam)}{P(doc|spam) + P(doc|ham)}. \tag{3}$$

The effectiveness of the AMFL framework depends on the splitting strategy, the combining strategy, and the active learning strategy; and the space-time complexity of the AMFL framework is mostly determined by the field TC algorithm.

3.2 Splitting Strategy

The explicit splitting strategy is based on the natural multi-field text structure of messages. For instance, the splitter can easily extract five natural fields (Header, From, ToCcBcc, Subject, and Body) for email documents and three natural fields (FromNumber, ToNumbers, and Body) for SMS message documents according to their natural multi-field text structures.

Except the explicit splitting strategy, this paper also proposes a novel artificial splitting strategy, which is motivated by that there are some classifiable texts hard to be pretended by spammers. For instance, spammers try to confuse the spam body text, but they dare not misspell their email addresses and phone numbers expected to be called back by spam receivers. We can extract these specific texts by some regular expression rules to form artificial fields which do not really exist in actual message documents. The artificial splitting strategy is equivalent to increasing the statistical weight for some specific texts.

This paper extracts two artificial fields (H.IP, H.EmailBox) for email documents and two artificial fields (BodyNumbers, Punctuation) for SMS message documents. The H.IP field contains IP address texts and the H.EmailBox field contains email address texts in the Header field of email documents. This paper also extracts all phone numbers in SMS Body field to form the artificial field BodyNumbers. The Punctuation field is made up of 2-character before each punctuation, each punctuation, and 2-character after each punctuation in SMS Body field. Considering both natural fields and artificial fields within the AMFL framework of this paper, the splitter implements a 7-field framework for email documents and a 5-field framework for SMS message documents.

3.3 Combining Strategy

The combining strategy used in the combiner is the key-point to guarantee good effectiveness within the AMFL framework. The linear combination of spamminess scores from field classifiers is an effective method, which is defined as Equation (4), where SS denotes the final spamminess score, n denotes the number of field classifiers, and SS_i denotes the spamminess score predicted by the i^{th} field classifier. The coefficient α_i (a real number) can be set by different weighted strategies. The straightforward weighted strategy is arithmetical average calculating method: $\alpha_i = 1/n$, abbreviated as $cs1$ combining strategy.

$$SS = \sum_{i=1}^{n} \alpha_i SS_i. \tag{4}$$

Spam filtering is an online active learning process, so the normalized historical classification accuracy rates of field classifiers can be used to estimate the linear combination coefficients. Within the AMFL framework, each field classifier's historical SS values can be plotted to a receiver operating characteristic (ROC) curve.

The percentage of the area below the ROC curve, abbreviated as ROCA, indicates the historical classifying ability. So each ROCA value is reasonable to estimate the classification accuracy rate of each field classifier. This historical performance weighted strategy was used in our previous research [9], abbreviated as $cs2$ combining strategy, where the normalized current n values of ROCA were used to set the coefficient α_i before a document is classified. Our research has also proved that the overall performance of $cs2$ precedes that of $cs1$.

Furthermore, the information amount of the current classified document will also influence the classification accuracy at the time of online predicting. The character number of each text field is the information amount measurement of each field, which can be used as the current classifying contribution weighted strategy, abbreviated as $cs3$ combining strategy, where the normalized character numbers of text fields are used to set the coefficient α_i.

In fact, the $cs2$ and $cs3$ strategies are two sides of the same coin. The both strategies, the historical performance weighted strategy and the current classifying contribution weighted strategy, affect the classification accuracy together. This paper presents a compound weight considering the both strategies on the assumption that the influences of $cs2$ and $cs3$ are equivalent. Let α_i^{cs2} and α_i^{cs3} denote separately the coefficient of the $cs2$ and $cs3$, then a compound weight, showed in Equation (5), is used as the coefficient α_i. This compound weight strategy is abbreviated as $cs4$.

$$\alpha_i = \frac{{\alpha_i}^{cs2} + {\alpha_i}^{cs3}}{2}. \tag{5}$$

The four linear combination strategies are refined step by step from $cs1$ to $cs4$, especially the $cs4$ strategy considers the two influences thoroughly, and can ensure the high classification accuracy. The combiner can combine the scores of multiple field classifiers to form the final SS by one of above four strategies. If the final $SS \in [0, T]$, then the document will be predicted as a ham; otherwise, if the final $SS \in (T, 1]$, it will be predicted as a spam, where T denotes the SS threshold and $T = 0.5$ in this paper.

After the binary classification decision, the active learning process is triggered to evaluate the current classifying confidence and make a decision whether a user feedback for the current document is requested. While the full feedback filter lacks this active learning process, and it will unconditionally get a user feedback immediately after the binary judgment. The user feedback will be sent to each field classifier for its TC model updating.

3.4 Active Learning Strategy

It is a key-point to active learning how to choose more informative training samples. The widely used uncertainty sampling [22, 23] is an effective method, which selects those easily wrong classified samples for training. The reason is that the more uncertain sample can highly improve the training.

In this paper, the active learner has a preset quota parameter of messages for which feedback can be requested immediately until the quota is exhausted. This quota is much less than the total number of messages. There are several strategies to exhaust this quota. The simplest strategy is the first coming priority strategy, abbreviated as $as1$, which requests a label for each coming message until the quota is exhausted. The $as1$ strategy is not a real active learning strategy, but a baseline to compare with other active learning strategies.

We can also set a heuristic uncertain range $(0.4, 0.6)$ of spamminess scores, and estimate whether the current SS, output from the combiner, belongs to the uncertain range. This explicit uncertainty sampling method is abbreviated as $as2$.

For each message, we can get several spamminess scores from field classifiers within the AMFL framework. These spamminess scores, the same inputs of the combiner, can also be used by the active learner. The difference among the spamminess scores indicates uncertainty of the current classification, which can be measured by their variance. We propose a historical-variance-based active learning (HVAL) strategy, abbreviated as $as3$, and the detail HVAL algorithm is showed in Figure 2. Here, the variable SS means an array of spamminess scores from field TC Results, the variable D means the historical average variance, the variable C means the number of training samples, and the variable Q means the preset quota.

```
// HVAL: Historical-Variance-based Active Learning algorithm.
// SS: Field TC Results; D: Historical Average Variance; C: Training Sample Count; Q: Quota.
// b: Sampling Rate; This paper sets (b = 1).
HVAL (ArrayList<Float> SS; Float D; Integer C; Integer Q)
(1) Float V := ComputeVariance (SS);
(2) If (V > b*D) And (Q > 0) Then:
   (2.1) Q := Q-1;
   (2.2) Request a Label L;
   (2.3) D := D*C+V;
   (2.4) C := C+1;
   (2.5) D := D/C;
(3) Output: L and D; C; Q.

ComputeVariance (SS) //For n real numbers SS_i ∈ SS, compute their variance.
```

Figure 2. Pseudo-code for the HVAL algorithm

From Figure 2, we find that the space-time complexity of the HVAL algorithm is very low. The main space cost is the locations of several real numbers, and the main time cost is also several multiplicative time. This space-time-efficient active learning algorithm will improve our AMFL framework's performance.

From the perspective of machine learning, the AMFL framework adds a document-level category label to each field document. Each field classifier can develop more sophisticated features and train a TC model in its own feature space, which reduces the feature disturbance between the several fields and makes the TC model

more precise; and the $as3$ active learning strategy makes use of multiple decision-makers to estimate the classifying confidence, and trends to regard that the huge divergence of opinions indicates the more uncertainty. The AMFL framework is a general structure, easily applied to integrate previous TC algorithms, because previous TC algorithms can be used to implement the field classifier by changing a binary result output to a continuous SS output.

The total space-time cost within the AMFL framework depends on the space-time complexity of each field classifier. Unfortunately, previous TC algorithms, normally using the VSM, have to align vector dimensions, select features, and often lead to high dimensional sparse and time-consuming problems. The online TC algorithm also faces an open incremental problem of the text feature space, and cannot foreknow the vector space dimension. The problems make previous TC algorithms unsuitable to be implemented as the field classifier for the practical spam filtering application. So this paper also explores a light space-time-efficient TC algorithm to implement the field classifiers.

4 FIELD TEXT CLASSIFICATION

We address the efficient online binary field TC problem, design a data structure of string-frequency index (SFI), and propose a SFI-based text classification (SFITC) algorithm, which is a general light field TC algorithm no binding with any concrete field and is suitable to implement the field classifiers owing to the space-time-efficient SFI data structure.

4.1 String-Frequency Index

The feature string frequency of historical labeled field texts implies rich classification information and must be stored effectively for online training. This paper applies the overlapping word-level 4-grams model, which can achieve promising results [24], to define feature strings, and lets a field text T be represented as a sequence of feature strings in the form $T = S_j$, $(j = 1, 2, \ldots, N)$. The string-frequency index is a data structure to store the feature string information of labeled field texts, from which we can conveniently calculate spamminess score of each feature string according to the scaled Bayesian conditional probability $P(spam|S_j)$, and straightforwardly combine the scores to form the field's final score.

Figure 3 shows the SFI structure for a field classifier including two integers and a hash table. The integers F_{spam} and F_{ham} denote separately the total number of historical labeled spam and ham field texts, which are used to scale up the number of two categories labeled field texts to make $P(spam) = P(ham)$. Each table entry is a key-value pair $\langle \text{Key}, \text{Value} \rangle$, where each key is a feature string and each value consists of two integers. The integers $F_{spam}(S_j)$ and $F_{ham}(S_j)$ denote separately the occurrence times of feature string S_j in historical labeled spam and ham field texts, and the S_j denotes the j^{th} feature string in the field text. The hash function maps the feature string S_j to the address of two integers $F_{spam}(S_j)$ and $F_{ham}(S_j)$.

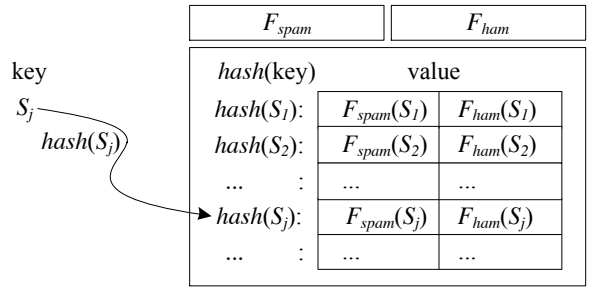


Figure 3. String-frequency index

4.2 SFITC Algorithm

Supported by the SFI, the SFITC algorithm takes the online classifying process of a field text as an index retrieving process, and also takes the online training process as an incremental updating process of index. Figure 4 gives the pseudo-code for the SFITC algorithm consisting of two main procedures: PREDICT and UPDATE.

When a new (Label = null) field text arrives, the PREDICT procedure is triggered:

1. It extracts the feature string sequence from the field text based on the overlapping word-level 4-grams model;
2. It retrieves the current SFI and calculates each feature string's SS according to the scaled Bayesian conditional probability described in Equation (6); and
3. It assumes that each feature string's contribution to the final SS is equivalent and uses the arithmetical average to calculate the final SS described in Equation (7).

$$SS_j = P(spam|S_j) = \frac{F_{spam}(S_j)/F_{spam}}{F_{spam}(S_j)/F_{spam} + F_{ham}(S_j)/F_{ham}} \tag{6}$$

$$SS_i = \frac{1}{N}\sum_{j=1}^{N} SS_j. \tag{7}$$

When a new labeled field text arrives, it is only required that the field text's feature strings are put into the SFI. The UPDATE procedure extracts the feature string sequence, and updates the frequency or adds a new index entry to the SFI according to the feature strings within the sequence.

4.3 Space-Time Complexity

The SFITC algorithm uses Bayesian conditional probability to estimate the probability of categories for a given field text, and belongs to an online Bayesian algorithm.

Based on the SFI, the SFITC algorithm overcomes some disadvantages caused by traditional VSM, and smoothly solves the online open problem of text feature space. Some time-consuming operations [25], such as vector alignment and feature selection, are avoided in the SFITC algorithm. The SFITC algorithm, independent of any concrete field, is a general robust field TC algorithm, whose space-time complexity depends on the SFI storage space and the loops in the PREDICT and the UPDATE procedures.

The SFI storage space is efficient owing to the native compressible property of index files. The SFI is an improved version of traditional inverted files [26], which simplifies the position and document ID information to two integers, only reflecting the occurrence frequency of feature strings. This hash list structure, prevailingly employed in Information Retrieval, has a lower compression ratio of raw texts. Though the training field texts will mount in the wake of the increasing of online feedbacks, the SFI storage space will increase slowly. Theoretically, the native compressible property of index files ensures that the SFI storage space is proportional to the total number of feature strings, and not limited to the total number of training field texts.

The incremental updating or retrieving of SFI has constant time complexity according to a hash function. The major time cost of the online classifying procedure is the $3N + 1$ divisions' time in loop (see 6.1 of Figure 4). The online training procedure is lazy, requiring no retraining when a new labeled field text added. From Figure 4, it is found that the time cost of per updating is only proportional to the total number of feature strings in the field text. Except the loop (see 2.2 and 3.2 of Figure 4) according to the number of feature strings, there is no time-consuming operation. The above time complexity is acceptable in the practical spam filtering application.

5 EXPERIMENT

In this section, we report results from experiments testing the effectiveness of the active multi-field learning approach from Section 3 with the field TC algorithm described in Section 4 for spam filtering. The experimental results from email and SMS spam filtering show strong support for the use of active multi-field learning in ubiquitous spam filtering.

5.1 Data Sets

For privacy protection, the public email corpus and SMS corpus are less. Email corpora mainly include trec05p-1 (39 399 hams and 52 790 spams)[1], trec06p (12 910 hams and 24 912 spams)[2], trec06c (21 766 hams and 42 854 spams)[3] and trec07p[4].

[1] http://plg.uwaterloo.ca/cgi-bin/cgiwrap/gvcormac/trec05p-1.tgz
[2] http://plg.uwaterloo.ca/cgi-bin/cgiwrap/gvcormac/trec06p.tgz
[3] http://plg.uwaterloo.ca/cgi-bin/cgiwrap/gvcormac/trec06c.tgz
[4] http://plg.uwaterloo.ca/cgi-bin/cgiwrap/gvcormac/trec07p.tgz

SMS corpora mainly include NUS SMS Corpus[5], SMS Han Message Corpus (120 040 messages)[6] and our csms data set.

We test the proposed approach universal to various spam messages by email spam filtering and SMS spam filtering. In email spam filtering, we use a large-scale, publicly available benchmark data set first developed for the TREC spam filtering competition: trec07p [21] containing 75 419 total email messages (25 220 hams and 50 199 spams). Previous researches of SMS spam filtering lacked the large-scale actual labeled SMS data set. For instance, the experiments only used the corpus with a thousand short messages [11]. In SMS spam filtering of this paper, we provide a large-scale actual labeled Chinese SMS data set: csms containing 85 870 total short messages (64 771 hams and 21 099 spams). Each short message in csms collection involves three text fields (FromNumber, ToNumbers, Body).

5.2 Evaluation Methodology

The TREC spam filter evaluation toolkit and the associated evaluation methodology are applied. We report the overall performance measurement (1-ROCA) %, the area above the ROC curve percentage, where 0 is optimal, and the total running time to evaluate the filter's performance. We also report two measurements: the ham misclassification percentage (hm %) is the fraction of all ham classified as spam; the spam misclassification percentage (sm %) is the fraction of all spam classified as ham. All above measurements are automatically computed by the TREC spam filter evaluation toolkit. This toolkit can also plot a ROC curve and a ROC learning curve for ROC analysis. The ROC curve is the graphical representation of spam misclassification percentage and ham misclassification percentage. The area under the ROC curve is a cumulative measure of the effectiveness of the filter over all possible values. The ROC learning curve is the graphical representation of the filter's behavior and the user's expectation evolve during filter use. The ROC learning curve is that cumulative (1-ROCA) % is given as a function of the number of messages processed, which indicates that the filter has reached steady-state performance.

5.3 Implementations

We implement the total of 17 spam filters (10 filters for email spam filtering and 7 filters for SMS spam filtering). Each email spam filter has a 7-field splitter (five natural fields, and two artificial fields, abbreviated as n5a2), and each SMS spam filter has a 5-field splitter (three natural fields, and two artificial fields, abbreviated as n3a2) described in Section 3.2. All field classifiers apply the SFITC algorithm described in Section 4, and the feature strings are based on overlapping word-level 4-grams model. There are the total of four combining strategies described in Section 3.3, and three active learning strategies described in Section 3.4 in the spam

[5] `http://wing.comp.nus.edu.sg:8080/SMSCorpus`

[6] `http://cbd.nichesite.org/CBD2013D001.htm`

filters. For each active learning strategy, we separately run in two kinds of quota parameters ($Quota = 10\,000$ and $Quota = 1\,000$). The detail configuration of the spam filters is shown in Table 1.

	Message	Splitter	Combiner	Feedback
ndtF1	email	n5a2	$cs1$	full
ndtF2	email	n5a2	$cs2$	full
ndtF3	email	n5a2	$cs3$	full
ndtF4	email	n5a2	$cs4$	full
ndtF5	email	n5a2	$cs4$	$as1$ ($Quota = 10\,000$)
ndtF6	email	n5a2	$cs4$	$as2$ ($Quota = 10\,000$)
ndtF7	email	n5a2	$cs4$	$as3$ ($Quota = 10\,000$)
ndtF8	email	n5a2	$cs4$	$as1$ ($Quota = 1\,000$)
ndtF9	email	n5a2	$cs4$	$as2$ ($Quota = 1\,000$)
ndtF10	email	n5a2	$cs4$	$as3$ ($Quota = 1\,000$)
ndtF14	SMS	n3a2	$cs4$	full
ndtF15	SMS	n3a2	$cs4$	$as1$ ($Quota = 10\,000$)
ndtF16	SMS	n3a2	$cs4$	$as2$ ($Quota = 10\,000$)
ndtF17	SMS	n3a2	$cs4$	$as3$ ($Quota = 10\,000$)
ndtF18	SMS	n3a2	$cs4$	$as1$ ($Quota = 1\,000$)
ndtF19	SMS	n3a2	$cs4$	$as2$ ($Quota = 1\,000$)
ndtF20	SMS	n3a2	$cs4$	$as3$ ($Quota = 1\,000$)

Table 1. Spam filter configuration

Three other filters are chosen as baselines:

1. the bogo filter (bogo-0.93.4) [27, 28] is a classical implementation of VSM-based online Bayesian algorithm;
2. the tftS3F filter [29] is based on the relaxed online SVMs algorithm and has gained several best results at the TREC2007 spam track; and
3. the wat3 filter [8], the winner at the trec07p full feedback spam track, is based on the online fusion of DMC and logistic regression, and whose overall performance (1-ROCA) % is the best one (0.0055).

The bogo filter[7] is offered under the GNU General Public License, and the tftS3F filter's source code is obtained from its author. So the two filters can be run in the same environment with our filters, and we can compare their running time to evaluate time complexity. The hardware environment for running experiments is a PC with 1 GB memory and 2.80 GHz Pentium D CPU.

[7] `http://bogofilter.sourceforge.net/`

5.4 Results and Discussion

We verify the effectiveness of the active multi-field learning for spam filtering through three experiments. The first is the full feedback experiment, in which we try to compare the performance of the four combining strategies in email spam filtering. We also evaluate the performance between our filter with the best combining strategy and other three full feedback filters (bogo, tftS3F, wat3) both in email spam filtering and SMS spam filtering. The second is the active learning experiment with the 10 000 quota, in which we try to compare the performance of the three active learning strategies both in email and SMS spam filtering. The last is similar to the second experiment except only with the 1 000 quota. Under this few quotas, we also evaluate the performance between our active learning filter and the top three filters at the TREC 2007 active learning track.

	Time (sec)	(1-ROCA) %	sm %	hm %	TREC2007 Full Feedback Rank
ndtF4	2 834	0.0055	0.21	0.11	
wat3		0.0055			1
ndtF2	2 776	0.0067	0.16	0.15	
ndtF3	1 910	0.0070	0.40	0.08	
ndtF1	1 863	0.0074			
tftS3F	62 554	0.0093			2
bogo	25 100	0.1558			

Table 2. Experimental results of email full feedback

In the first experiment, the bogo, tftS3F, and our four filters run in full feedback task on the trec07p corpus separately. The detailed experimental results are shown in Table 2. The results show that the ndtF4 filter can perform complete filtering task in high speed (2 834 sec), whose overall performance (1-ROCA) % is comparable to the best wat3 filter's (0.0055) among the participators at the trec07p evaluation. The time and (1-ROCA) % performance of our four filters exceed those of the bogo and the tftS3F more. Comparing the ndtF2 and the ndtF3 in the percentage of misclassified spams and hams, we find that the $cs2$ strategy optimizes spam's decision ($0.16 < 0.40$) and the $cs3$ strategy optimizes ham's decision ($0.08 < 0.15$). The sm % and hm % of ndtF4 shows that compound weight can consider both aspects.

Figure 5 shows the ROC curve and the ROC learning curve of the bogo, tftS3F, wat3, and our best ndtF4 filter, respectively. In the ROC curve, the area surrounded by the left border, the top border and the ndtF4 curve is relatively small, which means that the overall filtering performance of ndtF4 filter is promising. The ROC curve also shows that the overall performance is comparable among the tftS3F, wat3, and ndtF4 filters. In the ROC learning curve, around 7 000 training samples, the ndtF4 curve achieves the ideal (1-ROCA) % performance (0.01). Comparing the ndtF4, tftS3F, wat3 learning curves, we find that the performances are all dropping near 20 000 training samples. However, when close to 40 000 training samples, the ndtF4 can quickly return to the ideal steady-state, and the average overall perfor-

mance (1-ROCA) % can reach 0.0055. This indicates that the SFITC algorithm applying *cs*4 strategy of the AMFL framework has strong online learning ability.

	Time (sec)	(1-ROCA) %
ndtF14	264	0.0005
tftS3F	20 837	0.0010
bogo	6 631	0.1067

Table 3. Experimental results of SMS full feedback

In order to verify the effectiveness of the *cs*4 strategy in short text spam filtering, we also run the bogo, tftS3F, and our ndtF14 filters in full feedback task on the csms corpus separately. The detailed experimental results are shown in Table 3. The results show that the ndtF14 filter can complete filtering task in high speed (264 sec), whose overall performance (1-ROCA) % exceeds that of the other filters. The high overall performance of the ndtF14 filter can be explained by two main reasons:

1. The same to the email spam filtering, the *cs*4 strategy will cover the historical classifying ability of each field classifier and the classifying contribution of each text field in the current classified message; and
2. Artificial text field reduplicates some classifiable texts and partly solves the lack problem of text features for the short text.

Figure 6 shows the ROC curve and the ROC learning curve of the bogo, tftS3F, and our ndtF14 filter, respectively. The ROC curve shows that the performance of tftS3F and ndtF14 is comparable. The ROC learning curve shows that the ndtF14 filter has strong online learning ability.

	Time (sec)	(1-ROCA) %
ndtF5	1 422	0.0465
ndtF6	1 560	0.0200
ndtF7	1 976	0.0071

Table 4. Experimental results of email ($Quota = 10\,000$) active learning

From the first experiment, we find that the *cs*4 strategy is effective to both email spam filtering and short text spam filtering; and we will evaluate the active learning strategies through the second experiment.

In the second experiment, the ndtF5, ndtF6, ndtF7 filters run in active learning task ($Quota = 10\,000$) on the trec07p corpus separately. The detailed experimental results are shown in Table 4. The overall performance (1-ROCA) % of the *as*3 active learning strategy (0.0071) exceeds that of the *as*1 strategy (0.0465) and the *as*2 strategy (0.0200). This indicates that the *as*3 active learning strategy can indeed choose more informative samples.

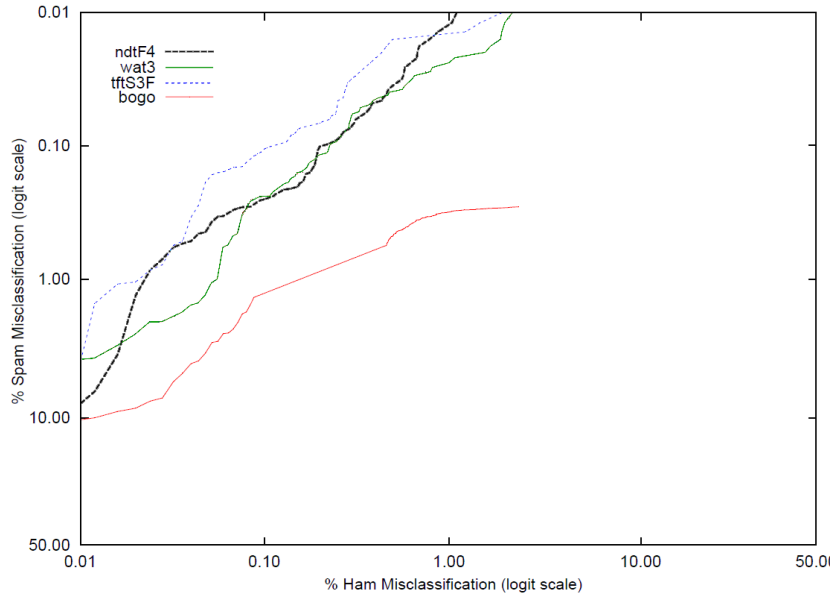
Moreover, the time (1 976) and the overall performance (0.0071) of the ndtF7 filter both outgo the full feedback filtering time (62 554) and the overall performance

```
// SFITC: String-Frequency-Index-based Text Classification algorithm.
// T: Field Text; L: Binary Category Label; SFI: String-Frequency Index.
SFITC (T; L; SFI)
(1) If (L = null) Then: PREDICT (T; SFI);
(2) Else: UPDATE (T; L; SFI).

// PREDICT: Online classifying procedure.
PREDICT (T; SFI)
(1) String[] S := FEATURE(T);
(2) Integer I_s := SFI.F_spam;
(3) Integer I_h := SFI.F_ham;
(4) New ArrayList<Float> F;
(5) If (I_s = 0) Or (I_h = 0) Then: Float SS_i := 0.5;
(6) Else:
   (6.1) Loop: For Each S_j ∈ S Do:
      (6.1.1) If (SFI.containKey(S_j)) Then:
         (6.1.1.1) Integer I_sj := SFI.F_spam(S_j);
         (6.1.1.2) Integer I_hj := SFI.F_ham(S_j);
         (6.1.1.3) Float SS_j := (I_sj/I_s)/(I_sj/I_s+I_hj/I_h);
         (6.1.1.4) F.add(SS_j);
   (6.2) Integer N := F.length;
   (6.3) If (N = 0) Then: Float SS_i := 0.5;
   (6.4) Else: Float SS_i := (1/N)∑SS_j; // SS_j ∈ F
(7) If (SS_i > 0.5) Then: Label L := spam;
(8) Else: Label L := ham;
(9) Output: SS_i and L.

// UPDATE: Online training procedure.
UPDATE (T; L; SFI)
(1) String[] S := FEATURE(T);
(2) If (L = spam) Then:
   (2.1) SFI.F_spam := SFI.F_spam + 1;
   (2.2) Loop: For Each S_j ∈ S Do:
      (2.2.1) If SFI.containKey(S_j) Then: SFI.F_spam(S_j) := SFI.F_spam(S_j) + 1;
      (2.2.2) Else: SFI.putKey(S_j), And SFI.F_spam(S_j) := 1, SFI.F_ham(S_j) := 0;
(3) Else If (L = ham) Then:
   (3.1) SFI.F_ham := SFI.F_ham + 1;
   (3.2) Loop: For Each S_j ∈ S Do:
      (3.2.1) If (SFI.containKey(S_j)) Then: SFI.F_ham(S_j) := SFI.F_ham(S_j) + 1;
      (3.2.2) Else: SFI.putKey(S_j), And SFI.F_spam(S_j) := 0, SFI.F_ham(S_j) := 1.

FEATURE (T) //Extract the feature string sequence from T based on overlapping word-level 4-grams model.
```

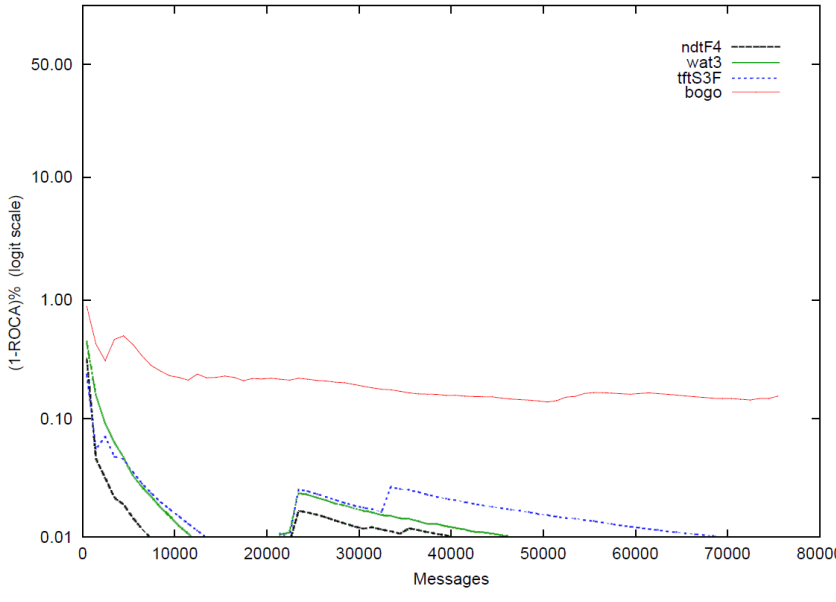
Figure 4. Pseudo-code for the SFITC algorithm

	Time (sec)	(1-ROCA) %
ndtF15	105	0.0126
ndtF16	107	0.0033
ndtF17	120	0.0009

Table 5. Experimental results of SMS ($Quota = 10\,000$) active learning

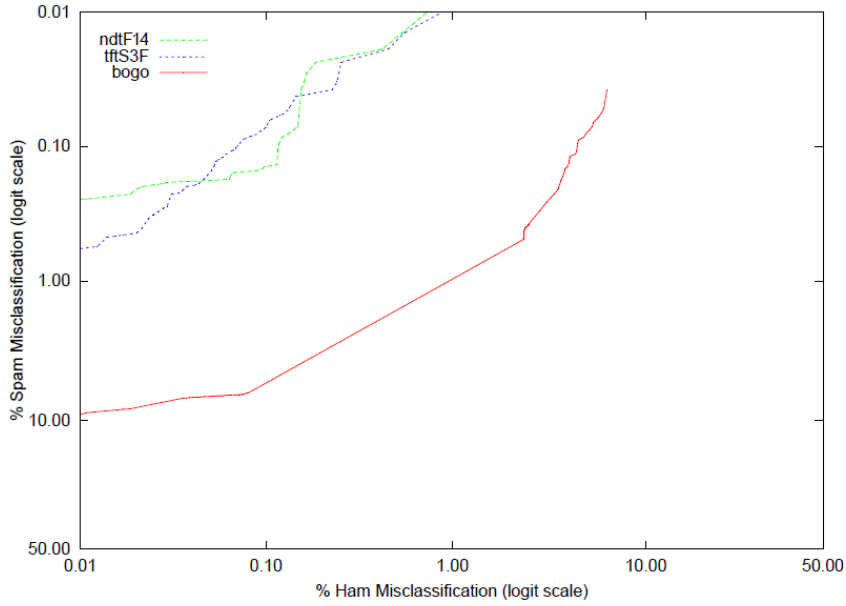


a)

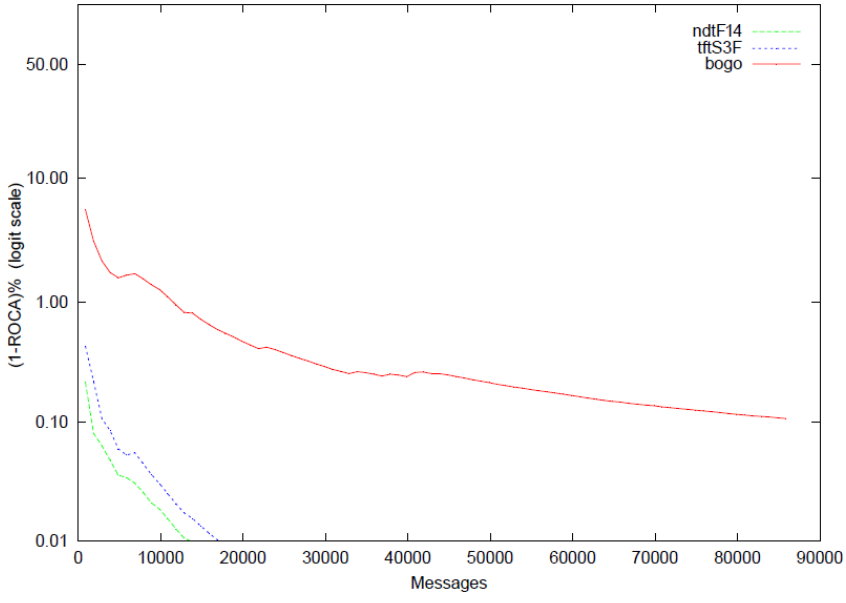


b)

Figure 5. Email full feedback: a) ROC curve and b) ROC learning curve

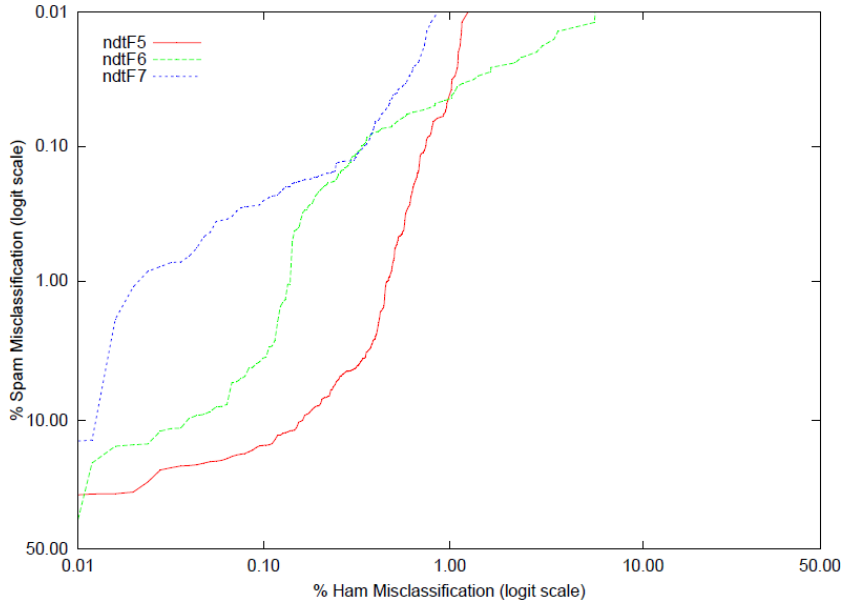


a)

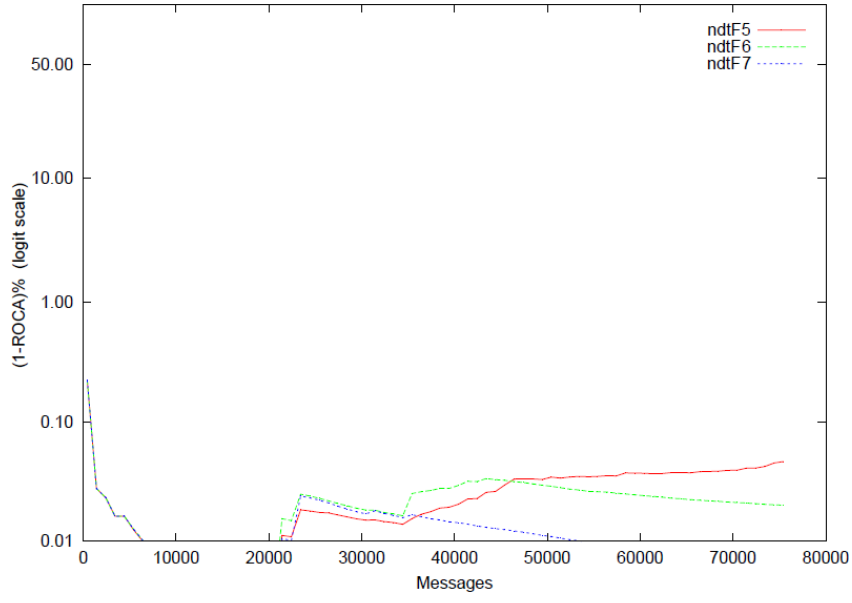


b)

Figure 6. SMS full feedback: a) ROC curve and b) ROC learning curve



a)



b)

Figure 7. Email ($Quota = 10\,000$) active learning: a) ROC curve and b) ROC learning curve

(0.0093) of the tftS3F filter, which is the second rank of the TREC 2007 full feedback track. This indicates that our proposed approach can achieve the state-of-the-art performance at greatly reduced label requirements in email spam filtering. Figure 7 shows the ROC curve and the ROC learning curve of the ndtF5, ndtF6, ndtF7 filters, respectively, which is also indicated by above results.

We also run the ndtF15, ndtF16, ndtF17 filters in active learning task ($Quota = 10\,000$) on the csms corpus separately to validate that our proposed approach is effective in short text spam filtering. The detailed experimental results are shown in Table 5, and the ROC curve and the ROC learning curve are shown in Figure 8. The time (120) and overall performance (0.0009) of the ndtF17 filter both outgo the full feedback filtering time (20 837) and performance (0.0010) of the tftS3F filter in SMS spam filtering. The experimental results and the curves shown in Figure 8 validate the effectiveness of our proposed approach.

	Time (sec)	(1-ROCA) %	TREC2007 Active Learning Rank
tftS2F		0.0144	1
wat4		0.0145	2
ndtF10	1 518	0.0380	
crm1		0.0401	3
ndtF8	1 392	0.0530	
ndtF9	1 486	0.0997	

Table 6. Experimental results of email ($Quota = 1\,000$) active learning

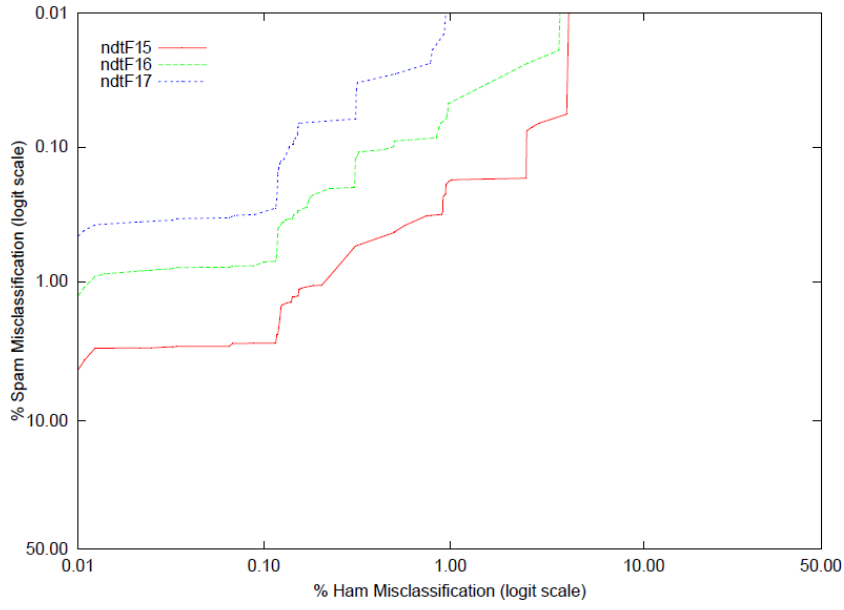
In the last experiment, we run the ndtF8, ndtF9, ndtF10 filters in active learning task ($Quota = 1\,000$) on the trec07p corpus separately. The detailed experimental results are shown in Table 6. The overall performance (1-ROCA) % of the ndtF10 filter (0.0380) can exceed that of the crm1 filter (0.0401), which is the third rank of the TREC 2007 active learning track. This indicates that our proposed approach is robust even in very small feedbacks.

Figure 9 shows the ROC curve and the ROC learning curve of the ndtF8, ndtF9, ndtF10 filter respectively. Figure 9 indicates that the ndtF10 even precedes the ndtF8, ndtF9 in very small feedbacks.

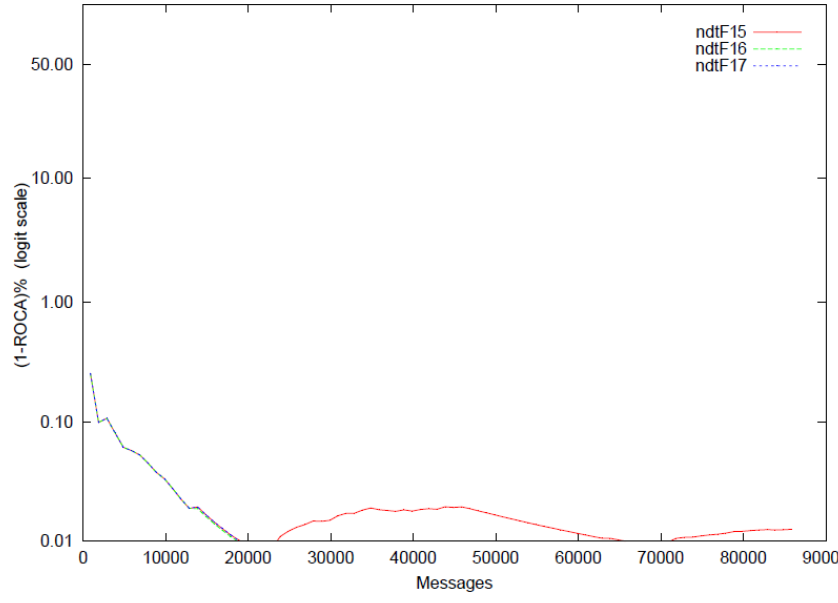
	Time (sec)	(1-ROCA) %
ndtF18	100	0.1175
ndtF19	104	0.1676
ndtF20	115	0.0575

Table 7. Experimental results of SMS ($Quota = 1\,000$) active learning

We also run the ndtF18, ndtF19, ndtF20 filters in active learning task ($Quota = 1\,000$) on the csms corpus separately. The detailed experimental results are shown in Table 7, and the ROC curve and the ROC learning curve are shown in Figure 10.

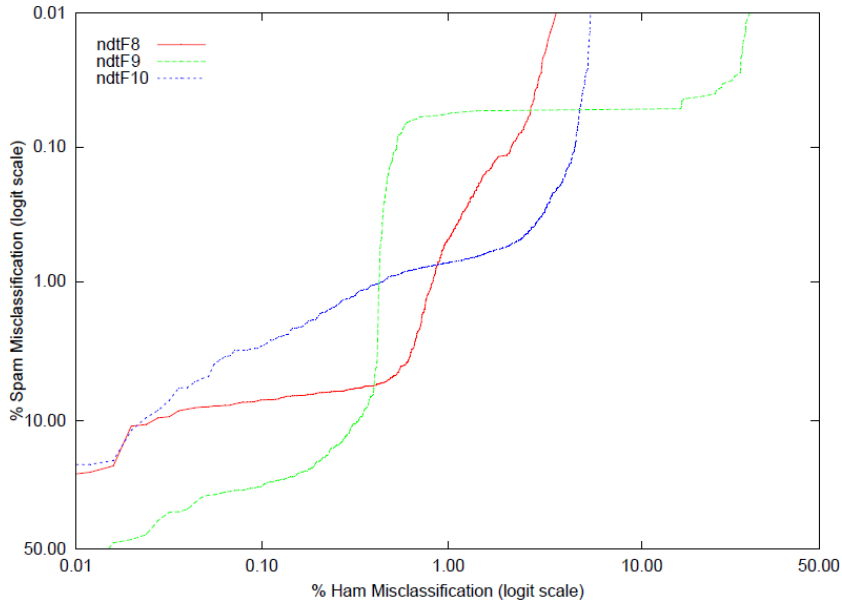


a)

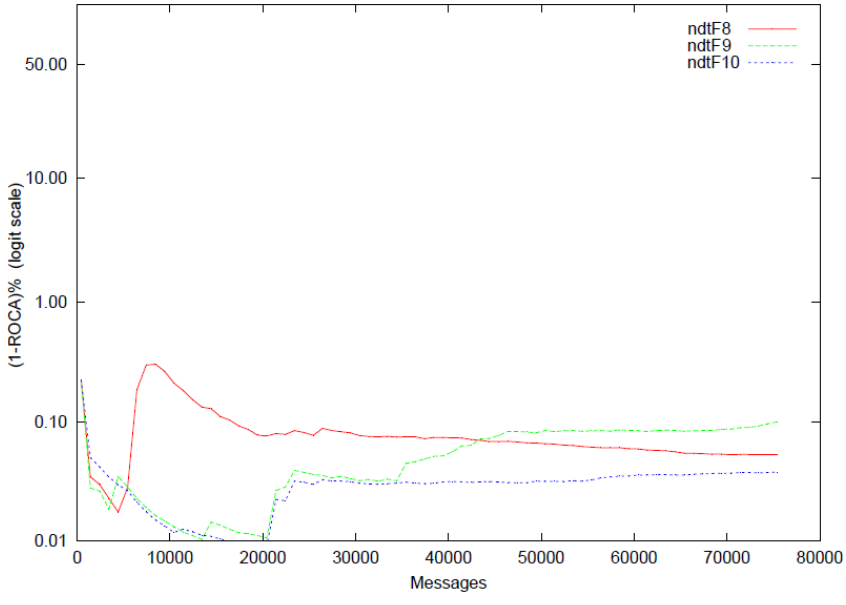


b)

Figure 8. SMS ($Quota = 10\,000$) active learning: a) ROC Curve and b) ROC learning curve



a)



b)

Figure 9. Email ($Quota = 1\,000$) active learning: a) ROC curve and b) ROC learning curve

From the above three experiments, we find:

1. the two splitting strategies are effective;
2. the compound weight combining strategy is the best among the four combining strategies;
3. the historical-variance-based active learning strategy is the best among the three active learning strategies; and
4. our proposed approach is effective in email spam filtering and short text spam filtering.

6 CONCLUSION

Recently, the spam concept has already generalized from email spam to various messages spam. Through the investigation of various messages, this paper proposes a universal approach for the practical application of large-scale spam filtering. The experiments have proved that proposed active multi-field learning approach can satisfactorily solve both email spam and short text spam problems.

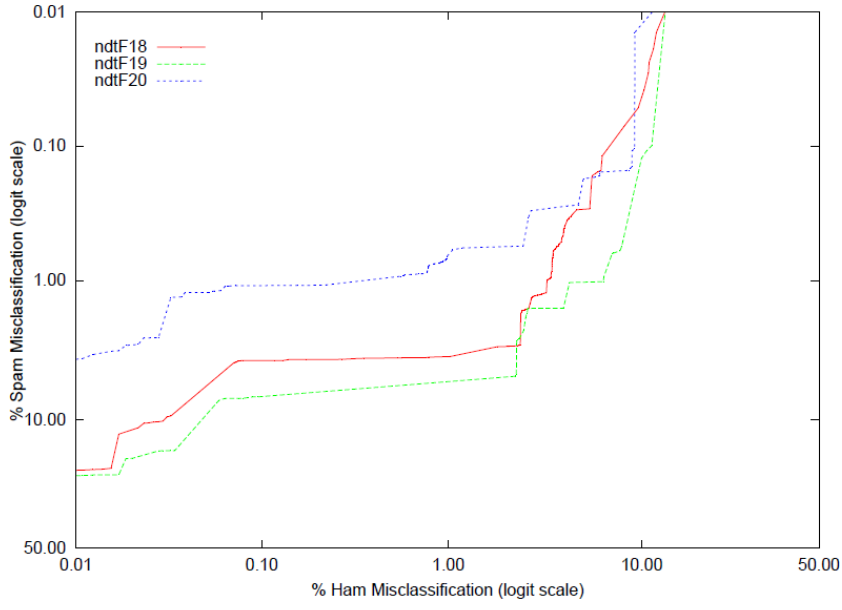
The contributions mainly include:

1. We find that different messages have a similar multi-field text structure, by which we can easily break a complex problem into multiple simple sub-problems. The ensemble of sub-problem results according to the compound weight can achieve the promising performance.
2. The difference among the results of field classifiers suggests a novel uncertainty sampling method, and the historical-variance-based active learning algorithm can choose informative samples and greatly reduce user feedbacks.
3. Using the SFI data structure to store labeled samples, the proposed SFITC algorithm is space-time-efficient, and well satisfies the requirements of large-scale online applications.

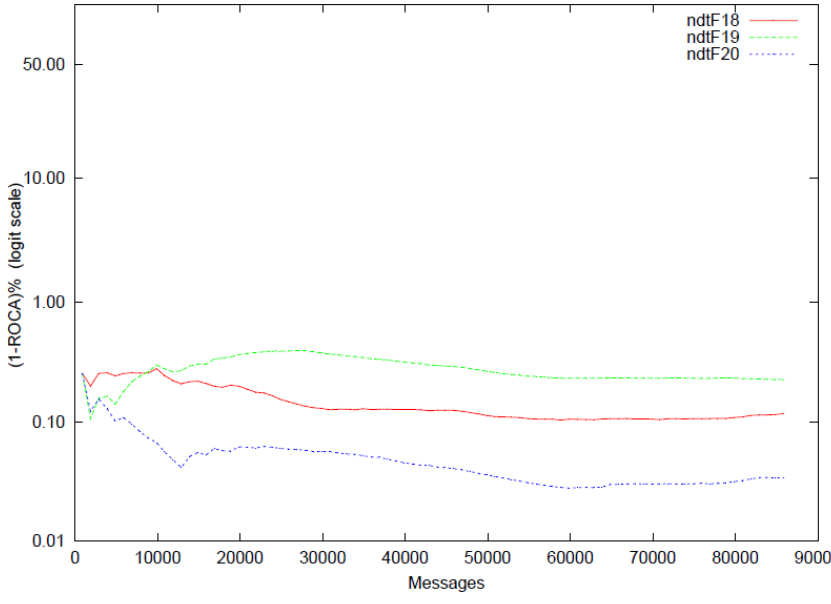
Moreover, the AMFL framework is suitable to parallel running environment, if it is applied on the reduplicate hardware for multiple field classifiers, the theoretical computational time of the AMFL framework to classify a message is nearly equal to the lowest field classifier's running time. In this paper, we also distribute a large-scale actual labeled Chinese SMS data set: csms containing 85 870 total short messages, which may be helpful to the short text research.

Based on the above researches, we can draw following conclusions:

- The multi-field structural feature can support the divide-and-conquer strategy. Using an optimal linear combination strategy of the compound weight, the straightforward occurrence counting of string features may obtain promising classification performance, even beyond that of some advanced individual algorithms. This straightforward counting will also bring time reduction.



a)



b)

Figure 10. SMS ($Quota = 1\,000$) active learning: a) ROC curve and b) ROC learning curve

- Uncertainty sampling is an effective active learning method. Within the AMFL framework, the multiple field classifiers bring an opportunity to estimate the uncertainty by the variance of multi-result. The historical-variance-based active learning algorithm is space-time-efficient, according to which the active learner regards the more uncertain message as the more informative sample.
- The index data structure has the native compressible property of raw texts, by which the information retrieval approach can be used to solve the information classification problem. Each incremental updating or retrieving of index has constant time complexity, which may satisfy the space-limited and real-time requirements of online applications.

In recent years the amount of spam messages has been dramatically increasing on the network. This paper proposed general, space-time-efficient approach that can be easily transferred to other spam filtering in ubiquitous environment. Further research will concern personal learning for spam filtering. We will improve the SFI structure for both global and personal labeled text storage.

REFERENCES

[1] Denning, P. J.: Electronic Junk. Communications of the ACM, Vol. 25, 1982, No. 3, pp. 163–165.

[2] Denning, P. J.: Infoglut. Communications of the ACM, Vol. 49, 2006, No. 7, pp. 15–19.

[3] Cormack, G. V. : Email Spam Filtering: A Systematic Review. Foundations and Trends in Information Retrieval, Vol. 1, 2007, No. 4, pp. 335–455.

[4] Gómez Hidalgo, J. M.—Cajigas Bringas, G.—Puertas Sánz, E.—Carrero García, F.: Content Based SMS Spam Filtering. Proceedings of the 2006 ACM Symposium on Document Engineering (DocEng '06), ACM Press 2006, pp. 107–114.

[5] Sculley, D.: Online Active Learning Methods for Fast Label-Efficient Spam Filtering. Proceedings of the 4th Conference on Email and Anti-Spam (CEAS 2007), 2007, 8 pp.

[6] Chai, K. M. A.—Ng, H. T.—Chieu, H. L.: Bayesian Online Classifiers for Text Classification and Filtering. Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '02), Tampere, Finland 2002, pp. 97–104.

[7] Sculley, D.—Wachman, G. M.: Relaxed Online SVMs for Spam Filtering. Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '07), 2007, pp. 415–422.

[8] Cormack, G. V.: University of Waterloo Participation in the TREC 2007 Spam Track. Notebook of the 16th Text REtrieval Conference (TREC 2007), National Institute of Standards and Technology, 2007.

[9] Liu, W.—Wang, T.: Multi-Field Learning for Email Spam Filtering. Proceedings of the 33rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '10), 2010, pp. 745–746.

[10] Cormack, G. V.—Gómez Hidalgo, J. M.—Puertas Sánz, E.: Spam Filtering for Short Messages. Proceedings of the 16th ACM Conference on Information and Knowledge Management (CIKM '07), ACM, New York 2007, pp. 313–320.

[11] Cormack, G. V.—Gómez Hidalgo, J. M.—Puertas Sánz, E.: Feature Engineering for Mobile (SMS) Spam Filtering. Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '07), 2007, pp. 871–872.

[12] Cormack, G. V.—Lynam, Th.: TREC 2005 Spam Track Overview. Proceedings of the 14th Text REtrieval Conference (TREC 2005), National Institute of Standards and Technology, Special Publication 500-266, 2005.

[13] Liu, W.—Wang, T.: Utilizing Multi-Field Text Features for Efficient Email Spam Filtering. International Journal of Computational Intelligence Systems, Vol. 5, 2012, No. 3, pp. 505–518.

[14] Liu, W.—Wang, L.—Yi, M.: Power Law for Text Categorization. Proceedings of the 12th China National Conference on Computational Linguistics (CCL 2013), 2013, pp. 131–143.

[15] Liu, W.—Wang, L.—Yi, M.: Simple-Random-Sampling-Based Multiclass Text Classification Algorithm. The Scientific World Journal, Vol. 2014, 2014, Article ID 517498, DOI: 10.1155/2014/51749.

[16] Cormack, G. V.: TREC 2006 Spam Track Overview. Proceedings of the 15th Text REtrieval Conference (TREC 2006), National Institute of Standards and Technology, Special Publication 500-272, 2006.

[17] Dietterich, T. G.: Ensemble Methods in Machine Learning. Proceedings of the Multiple Classifier Systems (MCS 2000), 2000, pp. 1–15.

[18] Roy, N.—McCallum, A.: Toward Optimal Active Learning through Sampling Estimation of Error Reduction. Proceedings of the 18th International Conference on Machine Learning (ICML '01), 2001, pp. 441–448.

[19] Tong, S.—Koller, D.: Support Vector Machine Active Learning with Applications to Text Classification. Journal of Machine Learning Research, Vol. 2, 2002, pp. 45–66.

[20] Cesa-Bianchi, N.—Gentile, C.—Zaniboni, L.: Worst-Case Analysis of Selective Sampling for Linear Classification. Journal of Machine Learning Research, Vol. 7, 2006, pp. 1205–1230.

[21] Cormack, G. V.: TREC 2007 Spam Track Overview. Proceedings of the 16th Text REtrieval Conference (TREC 2007), National Institute of Standards and Technology, Special Publication 500-274, 2007.

[22] Lewis, D. D.—Gale, W. A.: A Sequential Algorithm for Training Text Classifiers. Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '94), 1994, pp. 3–12.

[23] Lewis, D. D.—Catlett, J.: Heterogeneous Uncertainty Sampling for Supervised Learning. In Proceedings of the 11th International Conference on Machine Learning, 1994, pp. 48–156.

[24] Drucker, H.—Wu, D.—Vapnik, V. N.: Support Vector Machines for Spam Categorization. IEEE Transactions on Neural Networks, Vol. 10, 1999, No. 5, pp. 1048–1054.

[25] Radovanovic M.—Ivanovic M.—Budimac Z.: Text Categorization and Sorting of Web Search Results. Computing and Informatics, Vol. 28, 2009, pp. 861–893.

[26] Zobel, J.—Moffat, A.: Inverted Files for Text Search Engines. ACM Computing Surveys, Vol. 38, 2006, No. 2, Article 6, 56 pp.

[27] Graham, P.: A Plan for Spam. August 2002. Availaible on: `http://www.paulgraham.com/spam.html`.

[28] Graham, P.: Better Bayesian Filtering. In the 2003 Spam Conference, January 2003. Availaible on: `http://www.paulgraham.com/better.html`.

[29] Sculley, D.—Wachman, G. M.: Relaxed Online SVMs in the TREC Spam Filtering Track. Proceedings of the 16th Text REtrieval Conference (TREC 2007), National Institute of Standards and Technology, Special Publication 500-274, 2007.

Wuying Liu received his Ph. D. degree in computer science and technology from National University of Defense Technology, and currently is an instructor. His research interests include natural language processing, information retrieval and machine learning. He has published more than 40 research papers and 1 monograph.

Lin Wang received her Master of Arts degree in foreign linguistics and application linguistics from National University of Defense Technology, and currently is an instructor. Her research interests include computational linguistics, corpus linguistics and discourse analysis. She has published over 10 research papers.

Mianzhu Yi received his Doctor of Philological Sciences degree from Pushkin State Russian Language Institute, and currently is a Professor and doctoral supervisor. His research interests include computational linguistics, psycholinguistics, and ontological semantics. He has published over 60 research papers and 5 monographs.

Nan Xie is currently a Professor. Her research interests include theoretical linguistics, child language acquisition and philosophy of language. She has published over 10 research papers and 3 monographs.