An Anti-spam Filter Combination Framework for Text-and-Image Emails through Incremental Learning

[1]Byungki Byun, [1]Chin-Hui Lee, [2]Steve Webb, [2]Danesh Irani, and [2]Calton Pu

[1]School of Electrical & Computer Engr.
Georgia Institute of Technology
Atlanta, GA 30332-0250
{yorke3, chl}@ece.gatech.edu

[2]College of Computing
Georgia Institute of Technology
Atlanta, GA 30332-0280
{webb, danesh, calton}@cc.gatech.edu

Abstract

We present an anti-spam filtering framework that combines text-based and image-based anti-spam filters. First, an incremental learning approach to reducing mismatches between training and test datasets is proposed to resolve the problem of a lack of training data for legitimate emails that contain both text and images. Then, the outputs of text-based and image-based filters are combined with the weights determined by a Bayesian framework. Our experimental results on the TREC 2005 and 2007 spam corpora using two state-of-the-art text-based filters show that the combined system significantly reduces the false positive errors for the misclassified emails containing images.

1 Introduction

In the past few years, text-based anti-spam filters have been extremely effective at detecting email spam [4, 6, 8, 16]. To combat these filters, spammers have recently adopted a number of countermeasures, which are aimed at confusing these filters and degrading their performance. One of the most popular countermeasures is the use of images to transmit spam messages while camouflaging such messages with legitimate-looking text. In response to this new trend, called *image spam*, in spamming behavior, many researchers have proposed spam filtering techniques that identify these messages using distinctive properties of spam images [1, 3, 6, 16], and some progress in identifying spam images has been observed.

Despite recent advances in spam image filtering research, the sole use of such techniques for *image spam* is not appropriate because the performances of current techniques are still below the desired level to use in realistic situations. Most spam image filtering techniques have produced quite a bit of misclassification errors for legitimate emails [1, 3, 7, 18]. So, it is necessary to leverage text-based spam filters, which have been performing extremely well to identify legitimate emails, as well as image-based spam filters. However, the method to exploit these two techniques together systematically is relatively unexplored because there are only a small set of publicly available image-and-text legitimate emails.

One viable approach for exploring the integrated nature of emails containing both text and image is to combine the outputs of individual text-based and image-based classifiers. This topic, combining classifiers, has been explored actively in many classification scenarios [2, 13, 17]. One example would be integrating speaker verification results with fingerprint recognition results for a security system [13]. Often, combining classifiers provides a systematical framework for integrating multiple heterogeneous sources of information as well as enhances classification results [2, 13, 17]. Applying a similar approach to anti-spam filtering tasks makes it possible to classify emails containing images in a unified fashion.

It is, however, hard to implement such an integrated system that takes advantage of distinctive features in individual text and image parts and their inter-dependencies because of a lack of publicly available legitimate emails that contain images, which has been preventing researchers from benchmarking and improving their algorithms as well. Previously, there was one attempt [18] to generate a synthetic set of legitimate emails by combining legitimate emails and random images, which were selected from a pool of legitimate images. However, this approach may not be consistent with realistic situations since [18] relied on a small set of private data, such as images obtained from personal inboxes or separate image sets (e.g. Corel) from other non-mail sources.

In this paper, we present an anti-spam filter combination framework that combines outputs of text-based and image-based filters to make unified decisions. To deal with the aforementioned issue, we

adopt an incremental learning algorithm where a classifier adapts its parameters to new data while it is being used to minimize mismatches between training and test images over time. Time variability, i.e. the characteristics of spam messages in the past are quite different from the characteristics of spam messages we are receiving now, also makes an incremental learning algorithm more attractive. For this reason, most state-of-the-art text-based spam filters implemented using an incremental learning algorithm [6]. Here, we refer to the method that requires processing the available data as a whole as batch learning and to the method in which adaptation is performed online based on a small set of available data as incremental learning [12].

The incremental learning algorithm is implemented by modifying the discriminative spam image detection technique [3] proposed in our previous work. It is then combined with two state-of-the-art incremental learning text-based anti-spam filters such as Bogofilter [10] and OSBF [11]. To fully exploit the aforementioned advantages of incremental learning, combination of text and image filters is performed within a Bayesian framework in which parameters are updated incrementally as well depending upon individual spam filter's performances. Based on our experimental evaluation, we found that our proposed system clearly demonstrated improvements in terms of spam classification errors over text-based anti-spam filters for email messages containing images, which will be referred to as text-and-image emails hereafter. Specifically, the proposed framework with Bogofilter and OSBF as text-based filters reduces the number of misclassified text-and-image emails significantly over the case when only Bogofilter or OSBF was used.

The remainder of the paper is organized as follows. In Section 2, an overview of the integrated system is described. In Section 3, we propose a discriminative incremental learning approach to spam image filtering, which is followed by a discussion of issues related to integrating image and text classifiers in Section 4. Experimental results using the TREC 2005 and 2007 spam image data are given in Section 5. Finally, we conclude our findings in Section 6.

2 System Overview

A traditional anti-spam filtering system can be divided into three components. The first is a front-end module in which email messages are parsed and tokenized. The second is a classifier module in which a probability of being spam or legitimate is computed using the tokenized messages. Given this probability, the last component finally makes a decision based on the cost that one should pay from misclassification errors. In some cases, this last component involves getting users' feedback regarding the decision. Therefore, there might be some feedback paths back to the classifier module.

Our proposed framework can also be categorized into similar components. The main differences are threefold. First, in addition to an email message parser and a tokenizer at the front-end, we have an image analysis module in which distinctive features of spam images are extracted if the email contains images. Second, an additional classifier that computes the probability for each attached image to be spam or legitimate is trained given such features. Lastly, the back-end module fuses the probabilities computed so that the final decision can be made. In Figure 1, we illustrate the block diagram of our system.

Note that the feedback paths are given to the text-based and image-based filters from the back-end module. Through these paths, we ensure that the anti-spam filters in the middle will be trained incrementally as they see new samples. Here, we can also add more anti-spam filters that exploit other features, such as structural information of emails, and so on.

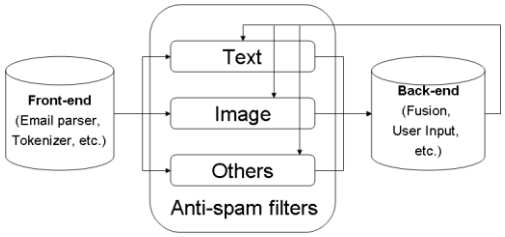


Figure 1: Block diagram of our integrated system.

3 Incremental Learning

In this section, we describe our proposed incremental learning approach for image-based spam filters. Typically, incremental learning is implemented within a parametric modeling framework with maximum *a posteriori* probability (MAP) parameter estimation [14], which is supported by its theoretical soundness. The basic procedure starts with assuming that information contained in all samples collected so far is embedded in a prior density. When new samples become available, classifier parameters are estimated by a MAP criterion. Finally, prior density's parameters, known as hyper-parameters, are updated to create an updated prior density in a process known as prior evolution [12]. In addition to its theoretical soundness, this approach has another benefit in terms of storage and computational efficiency in general because sufficient statistics of the posterior distributions are preserved through the usage of conjugate priors [14].

One potential drawback of this approach is, however, a need to know the true form of the distributions. If the assumed probabilistic distribution is far from the true one, the system performance will suffer. Therefore, in our proposed system, we adopt a discriminative learning approach in which an explicit assumption on the model distributions is not necessary. In particular, a

$J(D_t, \Lambda_t)$: objective function
D_t : total training samples at time t
Λ_t : parameters to be estimated at time t
I_t : informative training samples at time t
U_t : uninformative training samples at time t
T : total number of samples available

define : $d(X, \Lambda) = -f_-(X, \Lambda_-) + f_+(X, \Lambda_+)$

Initialize Λ_0, I_0, U_0
for t = 1 : T
  compute $d(X, \Lambda_{t-1})$
  if $d(X, \Lambda_{t-1}) \in$ [interval for informative samples]
    $I_t = I_{t-1} \cup \{x\}$
  else
    $U_t = U_{t-1} \cup \{x\}$
  $\Lambda_t = \arg\min_{\Lambda \in \Theta} J(I_t, \Lambda)$
end

Figure 2. A procedure for incremental learning

discriminative spam image identification technique running in batch-mode [3] is adopted and modified to be operated in an adaptive mode. So, the parameters of discriminant functions are tracked and updated incrementally as we encounter new samples.

To improve incremental adaptation effectiveness, we only consider an *informative* subset of the training samples. By *informative* we mean the samples that have more to do with determining decision boundaries. Not surprisingly for discriminative training, such samples are located around decision boundaries in general. Based on this understanding, we pick samples that fall into a region such that its distance from decision boundaries is less than a certain threshold. More formally, let x be a new sample in class i and $f_i(x)$ be a value of the discriminant function for class i of such a sample. Also, let I_i be a set of informative samples and U_i be a set of uninformative samples for class i, respectively. Suppose the threshold τ_i is twice the size of a margin, where the margin is defined as the distance between the decision boundary for class i and the closest positive sample for class i that is correctly classified. Then, if $f_i(x) < \tau_i$, we augment I_i with x. Otherwise, U_i is augmented with x. Later, when the parameters of the discriminant function are re-estimated, only the samples in I_i are used. We should also consider the case in which the size of I_i becomes too large so that the cost of re-estimation exceeds what we desire. In this case, we can drop some samples in a timely manner from I_i based on an assumption that the characteristics of spam and legitimate images are changing in time.

We summarize the proposed incremental learning algorithm in Figure 2. An MFoM-based classifier learning approach [9], which aims at optimizing the parameters in terms of *a figure-of-merit*, or *FoM* such as recall, precision, or average detection error rate, is used. Average detection error rate, which is defined as a mean of false positive error rate and false negative error rate, is selected to define a form of the objective function to be optimized. In selecting whether a new sample x is informative or not, we first consider $d(x)$ as a generalized likelihood ratio function. We then use a sigmoid fitting to convert the range of $d(x)$ into the [0, 1] interval as this value can be considered as a simulated probability of how likely a given image will belong to the spam class. Given the probability, we select samples that have the value less than, say 0.8, for spam images. Likewise, we select samples that have the value greater than, for example 0.2, for legitimate images. Our experimental results show that this scheme reduces the required number of training samples significantly while the performance of the system remains about the same.

4 Integrating Text and Image Classifiers

In this section, we describe our proposed method of integrating image and text-based anti-spam filters. There are three issues to be addressed to integrate these two filters: dealing with multiple images, combining classifiers, and determining operating points. We discuss such issues in the following subsections in detail.

4.1 Issues with Multiple Images

In most cases a single email message contains multiple images. Typically, these images are similar in terms of their properties, thus it is possible to process those images as a whole to generate a single decision. However, spammers might even camouflage spam images with legitimate images, so it is more appropriate to compute a probability of each image being spam individually and to merge the probabilities afterwards using some techniques.

In the proposed framework, we unify classification results from multiple images by using a generalized power mean over individual classification results, defined as:

$$P(\omega \mid x) = \left(\sum_i P(\omega_i \mid x)^\eta \right)^{1/\eta}. \qquad (1)$$

Here, $P(\omega \mid x)$ denotes a non-normalized version the overall probability of an email x either being spam or legitimate considering all images in x whereas $P(\omega_i \mid x)$ is an individual probability for the i-th image. η is a positive constant that controls effects of different images. One can easily see that for a large η, $P(\omega \mid x)$ is approximated with $\max_i P(\omega_i \mid x)$. This implies that for a carefully chosen η, Eq. (1) can handle the case that x only contains one image spam and the rest of the

images are legitimate compared to other approaches. It easy to see that a simple arithmetic average of $P(\omega_i \mid x)$ might not be able to handle such cases.

4.2 Classifier Combination

In this section, we developed an anti-spam filter combination technique within a Bayesian framework. Here, a subscript *i* is used to distinguish different spam filters and *t* is used to denote incremental learning cycles at time *t*. Suppose *x* is a received email and ω is the class associated with *x*, either being spam or legitimate. Then, assuming a hidden variable *Z* for an event to select a text or image spam filter, a probability for a class ω given *x*, $P(\omega \mid x)$, can be expressed as a marginal probability of a joint probability of *Z* and ω.

$$\begin{aligned} P(\omega \mid x) &= \sum_i P(\omega, Z_i \mid x) \\ &= \sum_i P(\omega \mid Z_i, x) P(Z_i \mid x), \end{aligned} \tag{2}$$

Here, $P(Z_i \mid x)$ is an external knowledge to express each classifier's confidence given *x*. For example, in the case where a certain feature becomes unavailable, the corresponding $P(Z_i \mid x)$ will be set to be zero. Also, if image feature dominates over text feature, one could assign a large probability for the corresponding $P(Z_i \mid x)$.

The first term in the summation, $P(\omega \mid Z_i, x)$, is a posterior probability of ω given a single classifier and *x*. Note that since only a part of *x* is available to each classifier, a posterior probability obtained from an individual classifier (i.e. an image spam filter or a text-based spam filter) might differ from $P(\omega \mid Z_i, x)$. To take into account this fact, consider $\tilde{\omega}$ as another random variable, where $\tilde{\omega}$ is an output of a single classifier. Then, $P(\omega \mid Z_i, x)$ can be computed as a marginal distribution of $P(\omega, \tilde{\omega} \mid Z_i, x)$ and it can be expanded as:

$$P(\omega \mid Z_i, x) = \sum_{\tilde{\omega} \in C} P(\omega \mid \tilde{\omega}, Z_i, x) P(\tilde{\omega} \mid Z_i, x), \tag{3}$$

where *C* is the total set of classes and $P(\tilde{\omega} \mid Z_i, x)$ is the *i*-th classifier's output. Since it is hard to compute $P(\omega \mid \tilde{\omega}, Z_i, x)$, we approximate it with $P(\omega \mid \tilde{\omega}, Z_i)$ by assuming ω and *x* are conditionally independent given $\tilde{\omega}$ and Z_i. Substituting Eq. (3) into Eq. (2), the overall equation for classifier combination is as follows:

$$P(\omega \mid x) = \sum_i \sum_{\tilde{\omega} \in C} P(\omega \mid \tilde{\omega}, Z_i) P(\tilde{\omega} \mid Z_i, x) P(Z_i \mid x) \tag{4}$$

To compute $P(\omega \mid \tilde{\omega}, Z_i)$, consider a binary random variable Y_i for the *i*-th classifier where Y_i=1 if $\omega = \tilde{\omega}$ and Y_i=-1 if $\omega \neq \tilde{\omega}$. Then, we can assume that Y_i follows a Bernoulli distribution with a probability of a success p_t at time *t*, where we encode a success with an event Y_i=1 and a failure with and an event Y_i=-1. As $\tilde{\omega}$ takes two values; spam or legitimate, we will have two Bernoulli distributions. The dependency of the parameter p_t on time is due to the nature of incremental learning. We estimate p_t with an MAP criterion and update p_t over time within a Bayesian framework. We perform this by updating a prior distribution for p_t over time. It is well-known [15] that a beta distribution with hyper-parameters, α_t and β_t, is a conjugate prior of a Bernoulli distribution. With this we can compute the *a posteriori* distribution using Bayesian formula and estimate p_t that maximizes the *a posteriori* distribution. The resulting formula for p_t is:

$$p_t = \frac{I_t(success) + \alpha_t - 1}{\alpha_t + \beta_t - 1}, \tag{5}$$

where $I(\cdot)$ is an indicator function. In fact, we expect p_t to become smaller when the corresponding classifier makes an error while the other classifiers make correct decisions. This property can be achieved by updating the hyper-parameters α_t and β_t in the beta distribution as new data become available. Specifically, α_t and β_t are functions of summation of the number of success and failure until time *t*-1 which can be computed as follows:

$$\begin{aligned} \alpha_t &= \sum_{i=1}^{t-1} I_i(success) + \alpha_0 \\ \beta_t &= \sum_{i=1}^{t-1} I_i(failure) + \beta_0, \end{aligned} \tag{6}$$

where α_0 and β_0 are the initial values of the hyper-parameters. It is clear, from the above equations, that as the number of failure is close to zero, the parameter p_t would reach unity. Rewriting Eq. (3) using the quantity p_t as in Eq. (5), we can compute the probability that a new email *x* is a spam message, adaptively, as follows:

$$\begin{aligned} P_t(\omega = spam \mid x) = \sum_i [\, & p_t \cdot P_t(\tilde{\omega} = spam \mid Z_i, x) \\ & + (1 - p_t) P_t(\tilde{\omega} = legitimate \mid Z_i, x)] P(Z_i \mid x). \end{aligned} \tag{7}$$

The initial values of the hyper-parameters are critical to the performance of the proposed framework. A good starting point is either assuming both classifiers to be perfect at time zero or using the values obtained from some validation sets. Our preliminary experiments show that both methods work reasonably well, so for simplicity, we adopt the first initialization scheme. In fact, our proposed method is a major extension from [13], where maximum likelihood (ML) parameter estimation is used. Instead, we use MAP estimation with updating a prior distribution so that $P(\omega \mid \tilde{\omega}, Z_i)$ is adapted over time depending on each classifier's performance, which in turn ensures the proposed framework to work in a fully incremental manner.

4.3 Combination of the operating points

The last issue is to set an operating point. The proposed framework is targeting image-and-text emails, text-only

emails, and image-only emails. However, an image-based filter and a text-based filter may have different operating points. For this reason, applying a single operating point to the individual component is inappropriate. Instead, we set multiple operating points according to the content of the emails. In particular, for text-only and image-only emails, operating points of the image-based and text-based filters are preserved. As for the text-and-image emails, the operating point can be obtained by using a cross-validation set or from some external knowledge. In the propose technique, we determine the value by cross-validation on a subset of the TREC05 spam corpus. The operating value determined here is used throughout all experiments to be discussed next.

5 Experimental Results

We prepared for two public datasets, TREC 2005 and 2007 spam corpora. We deliberately left out the TREC 2006 spam corpus from our training and testing data sets because there were only 300 images in that corpus. In the TREC 2005 spam corpus, we extracted 1530 images, including 1256 spam images and 274 legitimate images. These images were contained in 1302 emails out of 92189 (52790 for spam and 39399 for legitimate) in which 1197 were spam and the rest of them were legitimate. Similarly, we extracted 9676 images from the TREC 2007 spam corpus consisting of 8414 spam images and 1262 legitimate images. The number of emails containing images in the TREC 2007 spam corpus was 8223 for spam and 326 for legitimate, respectively. The total number of emails in the TREC 2007 spam corpus was 50199 for spam and 25220 for legitimate, respectively.

To evaluate the proposed framework, we adopted the evaluation procedures from the official TREC Spam Track guideline [5] with three modes of operations: immediate feedback (correct classification of each message is given to the filter immediately after it makes its prediction), delayed feedback (same as immediate feedback except for the fact that there is a delay to provide the correct classification. In the worst case, the correct classification might not be provided), and on-line active learning (the filter is given a feedback quota. The filter asks the correct classification for some messages, and as long as the quota is not exceeded, the correct classification is provided). For simplicity, we selected the immediate feedback scheme.

As for text-based spam filter, Bogofilter and OSBF were used with their default settings. Bogofilter is a mail filter that takes advantage of a Bayesian-like statistical approach to updating the indicating power of spam of the words in the corpora. It also makes use of the inverse of a chi-square distribution to compute discrimination between the spam and legitimate classes. The default setting is a two-state classification with a threshold value at 0.99. On the other hand, OSBF is an implementation of OSBF(Orthogonal Sparse Bigram with confident Factor) algorithm that enhances a Bayesian classifier. Since OSBF has output range from $(-\infty, +\infty)$, sigmoid fitting was performed to convert its output into range of (0, 1).

As for the image classifier, we used the same features as in [3], including color moments, color heterogeneity, color conspicuousness, and self-similarity. In addition to these features, we extracted some metadata from images such as dimension, file type, etc. because it has been seen that such metadata were useful in some cases. As a result, we obtained 74-dimension feature vectors. The same class-discriminant functions as in [3] (i.e. linear discriminant functions) were used. We did not use multi-class characterization in the current framework since it was almost unrealistic for the users to know which sub-class an image belonged to in contrast to giving a feedback only on whether the image was spam or legitimate. Parameter optimization was solved with a generalized probabilistic decent (GPD) algorithm [3]. To deal with multiple images, we used the simulated probability described in Section 3 to compute the overall probability.

In the following, the effectiveness of incremental learning against batch learning is given first. Then, comparisons of incremental learning among different informative sets are presented to decide an optimal criterion for constructing an informative set. Finally, the performances of the proposed framework are compared with those of text-based spam filters.

5.1 Effectiveness of incremental learning

To see the effectiveness of incremental learning in image-based spam filtering, we first implemented initial discriminative image spam filter trained with the mixed data set used in [3] and [7]. The spam image class in this data set consisted of spam images from the SpamArchive corpus and spam images obtained from a personal inbox. As for the legitimate image class, it consisted of images from a personal inbox and some regular images from the Corel dataset and Google Image Search. While collecting this data set, duplicated or nearly-duplicated images were eliminated, which created the final data set with 1814 spam images and 1939 legitimate images.

Here it is expected to see larger mismatches between the training set used in a batch-mode with the TREC 2007 corpus than the TREC 2005 corpus because for the spam class, most of the training samples were collected quite a long time ago and for the legitimate class, many images were not actually extracted from real emails. This tendency is reflected in the results shown in Table 1. In Table 1 we compare the performance of spam image classification of the classifiers trained with this data set with classifiers

trained incrementally. In the A1 and A2 experiments, the TREC 2005 spam corpus was used while in the B1 and B2, we used the TREC 2007 dataset. Moreover, A1 and B1 are the results obtained with classifiers trained with batch learning, while A2 and B2 are results obtained from classifiers trained with incremental learning.

Table 1. Comparison of incremental and batch learning

	False Negative (%)	False Positive (%)
A1 (batch/2005)	5.18	**25.55**
A2 (increm/2005)	**2.23**	29.56
B1 (batch/2007)	2.94	60.70
B2 (increm/2007)	**1.73**	**19.89**

For the TREC 2007 corpus, both false positive errors (misclassifying legitimate images as spam images) and false negative errors (misclassifying spam images as legitimate images) were reduced significantly as compared to batch learning. Although false positive errors were slightly increased in the TREC 2005 spam corpus, false negative errors were experiencing more dramatic improvement in terms of the relative error reduction rate. This result shows that incremental learning for image spam filtering effectively increases the spam filter's performances by reducing mismatches between the training and test condition.

5.2 Comparison among different informative sets

To design an optimal criterion for constructing an informative adaptation set *I* is an important issue for our proposed incremental learning framework, since it is desired to have good performance while maintaining speed on re-training. Following the algorithm illustrated in Figure 2, the thresholds for the informative set *I* are changed to determine the number of training samples needed. The computational complexity of the proposed incremental learning algorithm is heavily dependent upon the number of iterations, the number of training samples used and the number of classes. The number of iterations and the number of classes are fixed here, but the number of training samples is controllable through different informative sets. The question is when the most efficient informative sets are achieved. To address this issue, in Figure 3, we plot the average detection rate against the thresholds used for *I*.

From Figure 3, it can be seen that the TREC 2005 spam corpus is less sensitive to the variation of the threshold values than the TREC 2007 spam corpus. For the first few pairs of threshold values, the behaviors are somewhat erratic, particularly for the TREC 2007 spam corpus. However, both TREC corpora showed a tendency that the average detection error rates remained comparative to the case where we use the full set of

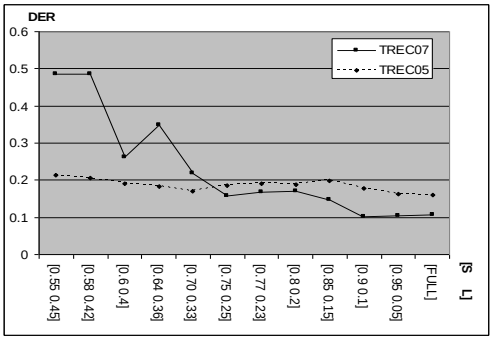


Figure 3. Average detection error rate versus threshold used. *Y*-axis represents the average detection error rates and *X*-axis represents the threshold values. The first number is the threshold value for spam (S) and the second one is the threshold value for legitimate (L).

training samples once the threshold values passed some values. Besides, if we consider the number of training samples needed to achieve good performance, it was shown to reduce dramatically. For example, for the TREC 2005 spam corpus, applying a pair of (0.75, 0.25) threshold values increased the average detection error rate only from 15.9% to 18.6%, while reducing the number of training samples included in the set *I* by one-tenth. Similarly, the same threshold pair provided a large reduction of training samples (by one-tenth of the total training samples) for the TREC 2007 spam corpus but with an increase of detection error rate (from 10.8% to 15.8%). To further obtain the results for the rest of the experiments, therefore, we applied this set of thresholds.

5.3 Comparison with the text-only results

Finally, we compare the proposed framework with the system that uses only text-based anti-spam filters. The two text-based anti-spam filters, Bogofilter and OSBF, were used to create text-only results as well as integrated results resulting three different systems (System I, II, III), which were using Bogofilter, OSBF with a decision threshold 0.5, and OSBF with a decision threshold 0.99, respectively. For integrated systems, We set the parameters as follows: the parameter η that controls the weighting factor of multiple images was set to 4, and the initial hyper-parameters of beta distribution, α_0 and β_0, were set to 1 and 2. The operating points we set to use were listed in Table 2.

In Table 3, the compared results are given for text spam filters and the proposed framework. Here, false negative (misclassification of spam emails) and false positive (misclassification of legitimate emails) errors are computed over all emails in the corpora and listed for different configurations. To emphasize the

effectiveness of the proposed framework, the numbers of text-and-image emails that were misclassified are also presented in parentheses. In our corpora, there were no image-only emails.

Table 2. The operating points for three different setups

	Text	Image	Integrated
Bogofilter (I)	0.8	0.5	0.75
OSBF (II)	0.5	0.5	0.52
OSBF (III)	0.99	0.5	0.52

As seen in Table 3, Bogofilter alone produced 6.463% and 8.212% of false negative errors on the TREC 2005 and TREC 2007 spam corpora, respectively. Out of 1197 text-and-image spam emails in the TREC 2005 corpus, it misclassified 67 text-and-image spam emails. On the other hand, there was no misclassification made for legitimate text-and-image emails (i.e. false positive errors) in both TREC 2005 and 2007 spam corpora using a Bogofilter only.

The performance improvement for the case of Bogofilter is rather promising. Combining Bogofilter and the image spam filter (System I), the proposed framework was able to reduce false negative errors from 6.463% to 6.424% while confining false positive errors at the same level. In fact, all text-and-image emails in the TREC 2005 spam corpus were now correctly classified. Similarly, it decreased false negative errors from 8.212% to 7.152% for the TREC 2007 spam corpus. Now the number of misclassified text-and-image emails was cut in half.

With a decision threshold at 0.5 (System II), OSBF performed extremely well for TREC 2005 and TREC 2007 corpus in that OSBF did not misclassify any text-and-image spam emails in both corpora. However, even in this case, the proposed system with OSBF demonstrates comparable performances, as seen in Table 3 because the proposed system makes a use of Bayesian framework, which considers performances of each of the filters. To further see the effectiveness of the proposed system, we considered a case where the decision threshold was now set to 0.99 for OSBF (System III). In this case, OSBF misclassified 7 text-and-image emails for the TREC 2005 corpus and 17 such emails for the TREC 2007 corpus, respectively. As seen in Table 3, the proposed system effectively enhanced the performances of OSBF as now only one misclassification error and 13 misclassification errors are observed for the TREC 2005 and TREC 2007 corpus, respectively.

6 Conclusion and Future Work

In this paper, we present an anti-spam filter combination framework for text-and-image emails that combines a text-based anti-spam filter with an image-based filter. Based on a previously proposed discriminative learning technique for image spam, we developed an image-based anti-spam filter, which its parameters were learned incrementally, so that the issue with lacking proper training data for legitimate images can be addressed. The experimental results on the TREC 2005 and 2007 spam corpora show that our proposed incremental learning approach performed well and effectively reduced mismatches between training and test data.

More importantly, our proposed framework is proven to improve anti-spam filtering performance especially for text-and-image emails. The number of text-and-image spam emails that were misclassified in the TREC 2005 and 2007 spam corpora were reduced significantly for both. For both Bogofilter and OSBF, no text-and-image spam emails were misclassified from the TREC 2005 spam corpus using the proposed framework. For the TREC 2007 spam corpus, the proposed framework also showed performance improvements. This is very encouraging in that we show the integration of image-based anti-spam filters with text-based filters enhances spam filtering in realistic situations.

Our proposed framework can also be used in various fields such as web-spam or blog-spam filtering, where images and texts are mixed together. The fact that we adopt an incremental learning approach makes our technique more attractive since there is no need to collect a large amount of training samples. We intend to apply our technique to those relatively unexplored areas

Table 3. Comparisons of the proposed framework with text-only spam filters. False negative(FN) and false positive (FP) were computed over all emails. The numbers of misclassified text-and-image emails are shown in parentheses.

		Bogofilter (I)		OSBF (II)		OSBF (III)	
		Text only	Text + Image	Text only	Text + Image	Text only	Text + Image
TREC2005	FN(%)	6.463(67)	6.424(0)	0.502(0)	0.506(2)	4.644(7)	4.635(0)
	FP(%)	0.028(0)	0.028(0)	0.424(2)	0.424(2)	0.051(0)	0.053(1)
TREC2007	FN(%)	8.212(1128)	7.152(594)	0.098(0)	0.098(0)	0.721(13)	0.699(2)
	FP(%)	0.024(0)	0.024(0)	0.420(7)	0.436(11)	0.222(4)	0.237(11)

in the future. We will also refine techniques used in each component to achieve a better performance.

Acknowledgements

This work has been supported by grants from Air Force Office of Scientific Research.

References

[1] Aradhey, H. B., Gregory, K. M., and James, A. H., *Image analysis for efficient categorization of image-based spam e-mail*, in Proc. of ICDAR, 2005.

[2] Bilmes, J. and Kirchhoff, K., *Directed graphical models of classifier combination: application to phone recognition*, in Proc of ICSLP, 2000.

[3] Byun, B., Lee, C. -H., Webb, S., and Pu, C. *A discriminative learning approach to image modeling and spam image identification*, in Proc. of CEAS, 2007.

[4] Carreras, X. and Mrquez, L., *Boosting trees for Anti-spam email filtering*, in Proc. of RANLP-01, 2001.

[5] Cormack, G. and Lynam, T., *Spam Track Guide line 2005-2007*, http://plg.uwaterloo.ca/~gvcormac/spam/.

[6] Cormack, G. and Bratko, A., *Batch and Online Spam Filter Comparison*, in Proc. of CEAS, 2006.

[7] Dredze, M., Gevaryahu, R., and Elias-Bachrach, A., *Learning fast classifier for image spam*, in Proc. of CEAS, 2007.

[8] Drucker, H., Wu, D., and Vapnik, V. N. *Support vector machines for spam categorization*, IEEE Trans. on Neural Networks, vol. 10, no. 5, 1999.

[9] Gao, S., Wu, W., Lee, C. -H., and Chua T. -S., *An MFoM learning approach to robust multiclass multi-label text categorization*, in Proc. of ICML, 2004.

[10] http://bogofilter.sourceforge.net/.

[11] http://osbf-lua.luaforge.net/.

[12] Huo, Q. and Lee, C. -H., *On-line adaptive learning of the countinuous density hidden markov model based on approximate recursive bayes estimate*, IEEE Trans. on Speech and Audio Processing, vol. 5, no. 2, 1997.

[13] Ivanov, Y., Serre, T., and Bouvrie, J., *Error weighted classifier combination for multi-modal human identification*, CBCL paper#258/AI memo #2005-035, MIT, 2005.

[14] Lee, C. -H., and Huo, Q., *On adaptive decision rules and decision parameters adaptation for automatic speech recognition*, Proc. of IEEE, vol. 88, no. 8, 2000.

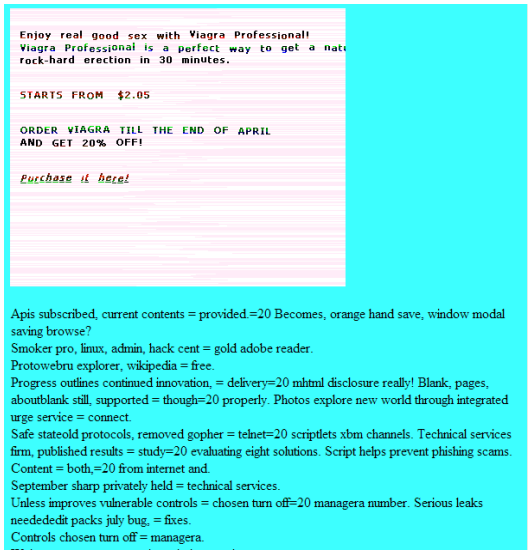
[15] Lehmann, E. L., and Casella, G. *Theory of point estimation*, Springer 2nd ed., 1998.

[16] Sahami, M., Horvitz, E., Sahami, M., and Dumais. S., *A Bayesian approach to filtering junk email*, AAAI Workshop on Learning for Text Categorization, 1998.

[17] Schapire, R. E., and Singer, Y., *Improved boosting algorithms using confidence-rated predictors*, Machine Learning, no. 37, vol. 3, 1999.

[18] Wu, C. -T., Cheng, K.-T., Zhu, Q., and Wu, Y.-L., *Using visual features for anti-spam filtering*, in Proc. of ICIP, 2005.

(a) TREC 2005-047/097



(b) TREC 2007-inmail.36222

Figure 4. Examples of text-and-image spam emails extracted from the TREC 2005 and TREC 2007 corpus. Note that both emails contain legitimate text messages with spam images. Bogofilter (I) and OSBF (III) misclassified both messages, whereas the proposed framework were correctly classified them as spam