A Comparative Study of Classification Based Personal E-mail Filtering

Yanlei Diao, Hongjun Lu[*] and Dekai Wu

Department of Computer Science
The Hong Kong University of Science and Technology
Clear Water Bay, Kowloon, Hong Kong
{diaoyl,luhj,dekai}@cs.ust.hk

Abstract. This paper addresses personal E-mail filtering by casting it in the framework of text classification. Modeled as semi-structured documents, E-mail messages consist of a set of fields with predefined semantics and a number of variable length free-text fields. While most work on classification either concentrates on structured data or free text, the work in this paper deals with both of them. To perform classification, a naive Bayesian classifier was designed and implemented, and a decision tree based classifier was implemented. The design considerations and implementation issues are discussed. Using a relatively large amount of real personal E-mail data, a comprehensive comparative study was conducted using the two classifiers. The importance of different features is reported. Results of other issues related to building an effective personal E-mail classifier are presented and discussed. It is shown that both classifiers can perform filtering with reasonable accuracy. While the decision tree based classifier outperforms the Bayesian classifier when features and training size are selected optimally for both, a carefully designed naive Bayesian classifier is more robust.

1 Introduction

As the Internet grows at a phenomenal rate, electronic mail (abbreviated as E-mail) has become a widely used electronic form of communication on the Internet. Everyday, a huge number of people exchange messages in this fast and inexpensive way. With the excitement on electronic commerce growing, the usage of E-mail will increase more dramatically. However, the advantages of E-mail also make it overused by companies, organizations or people to promote products and spread information, which serves their own purposes. The mailbox of a user may often be crammed with E-mail messages some or even a large portion of which are not of interest to her/him. Searching for interesting messages everyday is becoming tedious and annoying. As a consequence, a personal E-mail filter is indeed needed.

The work on building an E-mail filter can be cast into the framework of text classification: An E-mail message is viewed as a document, and a judgement of

[*] The second author's work is partially supported by a grant from the National 973 project of China (No. G1998030414).

interesting or not is viewed as a class label given to the E-mail document. While text classification has been well explored and various techniques have been reported [2, 3, 7], empirical study on the document type of E-mail and the features of building an effective personal E-mail filter in the framework of text classification is only modest.

Along the line of empirical study on E-mail classification, Fredrik Kilander summarized real users' suggestions and opinions on what should be the important properties in classifying electronic texts and messages [4]. A preliminary study claimed that threading electronic mail [6] could only gain partial success based on structured information. A significant level of effectiveness could be achieved by applying standard text matching methods to the textual portions. A prototype, *Smokey* [12], combined natural language processing and sociolinguistic observations to identify insulting messages. This work differed from general electronic text classification, focusing mainly on language processing. A Bayesian approach to filtering junk E-mail was presented in [11]. It considered domain specific features in addition to raw text of E-mail messages. Elaborating on commercial junk E-mail, it enhanced the performance of a Bayesian classifier by handcrafting and incorporating many features indicative of junk E-mail. William Cohen compared a "traditional IR" method based on *TF-IDF* (*Term Frequency – Inverse Document Frequency*) weighting and a new method for learning sets of "keyword-spotting rules" based on the *RIPPER* rule learning algorithm [1]. The experiments, however, were only conducted with a relatively small number of data sets of real users. The issues related to building an effective E-mail classifier were not fully considered either.

The work reported in this paper was motivated by our belief that to realize an effective personal E-mail filter in the framework of text classification, the following issues should be fully taken into account.

- An E-mail filter is personalized and the knowledge used by each personal filter is subjective. Therefore, classifying personal E-mail messages is more challenging than using a priori knowledge to filter commercial junk messages that are often characterized by symbols and words like '$', "free", "saving", etc.
- An in depth study on the distinct type of E-mail documents is needed to make full use of the information embedded in them. Feature selection is the key issue.
- Typical text classification techniques should be examined and compared to enable better understanding of the capabilities and characteristics of these techniques to perform the task of a personal E-mail filter.
- A relatively large amount of real E-mail data from individuals with different interests should be used in experiments.

For the problem of classifying E-mail documents, the objects to be classified are semi-structured textual documents consisting of two portions. One portion is a set of structured fields with well-defined semantics and the other portion is a number of variable length sections of free text. We would like to emphasize this feature in our study because information from both portions is important. In the case of E-mail messages, the fields in the mail header such as the sender and the recipient are very informative when we determine how interesting the message part is. On the other hand, the interestingness of an E-mail message from the same sender also depends on the content of the body message. However, not many text classifiers take both portions into consideration. For example, the classic document clustering techniques in information retrieval seldom consider the contents of structured fields. On the other hand, conventional classification techniques may not be effective when dealing with variable length free text.

There have been a number of approaches developed for classification. We selected two most popular approaches, naïve Bayesian classification [5, 8] and decision trees [9] to classify personal E-mail messages. The naïve Bayesian approach was chosen because it is widely used in text processing. Decision tree was chosen because of its effectiveness in classifying relational data. For the naïve Bayesian approach, a classifier based on previous work with some extensions was designed and implemented. For the decision tree approach, we implemented a classifier based on the widely used C4.5 system [10].

A series of experiments were conducted on a relatively large amount of real personal E-mail data. The behaviors of the two classification approaches were compared and discussed in detail. We find that both approaches provide reasonable performance in terms of recall rate and classification accuracy. Decision tree outperforms Bayesian a little when features and training size are selected optimally for both. However, the naïve Bayesian classifier is more robust with respect to the size and class disparity of training data.

The remainder of the paper is organized as follows. Section 2 discusses the modeling and features of E-mail messages. Section 3 presents our design and implementation of a naïve Bayesian classifier and a decision tree based classifier for E-mail filtering. The experiments and results are presented in Section 4. Finally Section 5 concludes the paper with discussions on future work.

2 Document Model

In this section, we describe how E-mail documents are modeled and how features are selected to perform personal E-mail filtering.

2.1 Modeling Semi-structured Documents

In a broad sense, E-mail messages are semi-structured documents that possess a set of structured fields with predefined semantics and a number of variable length free-text fields. In a formal way, such a document can be represented as Fig.1.

Field 1 to *Field s* are structured fields and usually contain information pertaining to the document, such as authorship, date, organization, layout of the text body, etc. As the major contents of the document, *Field s+1* to *Field s+t* are variable length free-text fields, such as subject area, abstract, the body and references. While most classification work focuses on either the structured part or the text part, we argue that both the structured fields and the free-text portion could contain important information for determining the class to which a document belongs. Therefore, a classifier to serve the purpose should be able to include features from both the structured fields and the free text.

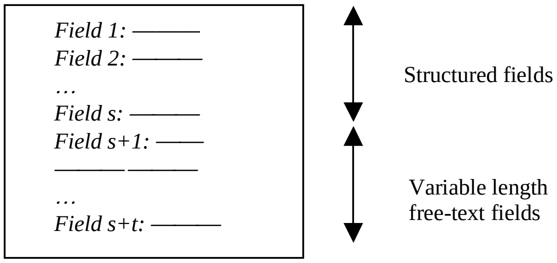

Fig. 1. Modeling Semi-structured Documents

2.2 Modeling Electronic Mail

In general, E-mail messages belong to the board class of semi-structured documents. Therefore they inherit the characteristics of possessing two portions of fields. In particular, they have some unique features. In addition to the structured fields and free text, there is evidence showing that domain specific information implicit in text fields is useful to improve the classification accuracy in certain applications. For example, Sahami et. al. reported that, there are many particular features of E-mail that help determine if a message is junk or not [11]. Such features include phases like "Free Money", and over-emphasized punctuation, such as "!!!". Since a feature in the free-text part normally refers to a single word, these particular features are treated as the third type, handcrafted features. To make full use of the information in an E-mail message, we generated all three types of features for each document.

- *Structured features*: features represented by structured fields in the header part of an E-mail document. In this work, six structured features were generated. They are *SenderDomain* (the domain of a sender, such as .com and .edu), *SenderAddress* (the E-mail address of a sender), *Recipient* (single recipient, in a group with the name mentioned, or via a mailing list), *Date*, *MailType* (replied, forwarded or sent directly), and *ContentType* (having attachment or not).
- *Textual features*: features explicitly contained in a free text section. In this work, only words consisting of alphabetic letters (no numbers or symbols) were counted into the vocabulary. Furthermore, a standard stop list was used to remove those words insignificant to classification. Simple stemming was also applied to reduce the vocabulary size.
- *Handcrafted features*: features obtained by preprocessing the documents. Heuristically six features were handcrafted. They are (1) the number of exclamation marks, (2) the number of dollar signs, (3) the number of http links, (4) the length of the message, (5) the length of the subject line, and (6) the occurrence of words indicative of not interesting E-mail in the subject area (a list of such words was collected in advance).

The usefulness of each type of features in different classifiers will be discussed in detail in experiments.

3 Two Classifiers

3.1 A Naïve Bayesian Classifier

The Bayesian learning approach views a text document as a bag of words. For a naive Bayesian classifier to work effectively, two conditions should be satisfied. First any word inside a document occurs independently. Second, there is no linear ordering of the word occurrences. Although these two assumptions may not hold in real cases, naïve Bayesian classifiers do provide good performance in a lot of applications [3, 8].

Two different generative models, the *multi-variate Bernoulli* model and the *multinomial* model under the Bayesian framework were reported in [8]. Experimental results show the multinomial model usually outperforms the multi-variate Bernoulli model when vocabulary size is large or chosen optimally for both. Thus, we adopted the multinomial model with a little simplification as shown from formula (1) to (4).

The following symbols are used in the rest part of this paper. $C_1 \ldots C_K$ are a set of class labels of a class variable C. $D_1 \ldots D_m$ are a set of training documents. $F_1 \ldots F_n$ represent a set of features in a given document. The class label of a document D' is determined as follows:

$$C = \arg\max_k P(C_k \mid D') = \arg\max_k P(D' \mid C_k)P(C_k)\,. \tag{1}$$

Since a document is represented by a set of features $\{F_1 \ldots F_n\}$, with the naïve Bayes assumption that each feature in a document occurs independently, we have:

$$C = \arg\max_k P(F_1 \mid C_k)P(F_2 \mid C_k)\ldots P(F_n \mid C_k)P(C_k)\,. \tag{2}$$

With a given set of labeled samples (the training data), the training process calculates *Bayes-optimal* estimates for all the parameters. Here the estimation of the probability of feature F_j on condition of class k and each class prior are obtained as follows:

$$P(F_j \mid C_k) = \frac{1 + \sum_{i=1}^{|D|} N(F_j, D_i)P(C_k \mid D_i)}{|V| + \sum_{t=1}^{|V|} \sum_{i=1}^{|D|} N(F_t, D_i)P(C_k \mid D_i)}\,, \tag{3}$$

$$P(C_k) = \frac{\sum_{t=1}^{|V|} \sum_{i=1}^{|D|} N(F_t, D_i)P(C_k \mid D_i)}{\sum_{k=1}^{K} \sum_{t=1}^{|V|} \sum_{i=1}^{|D|} N(F_t, D_i)P(C_k \mid D_i)}\,. \tag{4}$$

Here $N(F_j, D_i)$ is the number of occurrences of feature F_j in document D_i, $P(C_k \mid D_i)=\{0,1\}$ is given by the class label of that document, $|D|$ denotes the number of training documents and $\sum_{t=1}^{|V|} P(F_t \mid C_k) = 1$. To handle the probability of non-occurring features in the training data, add-by-one smoothing is used. $|V|$ is the vocabulary size.

Note that we are classifying E-mail messages that are distinct in document type. A feature involved in classification could be either a word in the text portion or a certain property (structured feature or handcrafted feature) associated to the document.

A Bayesian classifier has the advantage of being able to handle a large number of features. It simply models a document as "a bag of words" and all the words together form the vocabulary of the classifier. Naturally each word consisting of alphabetic letters in the main text portion is one feature in the vocabulary. To accommodate other two types of features in classification, a simple way is to treat such features as certain artificially created words and extend the vocabulary to include those features. The advantage of this approach is no need to modify the classifier. The importance of a feature is reflected uniformly by the probability of F_j on condition of class C_k no matter what type the feature belongs to.

Another issue of building a classifier in the context of E-mail messages is cost sensitiveness. When we assign a class label with the maximum class probability among all to a document, we are making an implicit assumption that the cost of misclassification is the same to all classes. In this application, the assumption is not true. Let C_1 denote the class label of "not interesting" and C_2 the class label of "interesting" (this notation will be used in the rest of the paper). The cost of misclassifying an interesting message to be not interesting is obviously much higher than that of misclassifying a not interesting message to be interesting. To make the naïve Bayesian classifier cost sensitive, we introduce to (2) one design parameter, threshold α_k for each class label k with $\sum_k \alpha_k = 1$:

$$C = \arg\max_k \left(\frac{P(F_1 \mid C_k) P(F_2 \mid C_k) \cdots P(F_n \mid C_k) P(C_k)}{\alpha_k} \right). \quad \textbf{(5)}$$

In this application with two class labels, the intuitive meaning of the threshold is as follows: In the case where misclassifying C_2 (interesting) into C_1 (not interesting) is more costly, we only make a prediction of class label C_1 if the final probability for decision making, $P(C_1|D')$, is greater than the threshold α_1, otherwise class label C_2 is chosen. In the rest part of the paper, for simplicity we use α to represent α_1. The classifier is cost sensitive with $\alpha > 0.5$. If we set $\alpha = 0.5$, we will have a normal cost-insensitive classifier.

3.2 A Decision Tree Based Classifier

Decision tree is a widely used data mining technique for classification and prediction, which is intensively studied in data mining applications in databases. C4.5, a typical and effective method of building decision trees, was used in our work to build a classifier of E-mail documents.

For a decision tree based approach, the situation is different from a Bayesian classifier. There is no problem for it to cope with the structured features and the handcrafted features since the number of these features (or attributes) is relatively small. However, it is not easy to handle a large number of textual features if every feature in the text is used in classification. A straightforward way is to limit the number of textual features that are considered by the classifier when a tree is built. In order to select textual features from the vocabulary, mutual information [8] is computed for each textual word F_t:

$$I(C; f_t) = \sum_{C_k \in C} \sum_{f_t \in \{0,1\}} P(C_k, f_t) \log(\frac{P(C_k, f_t)}{P(C_k)P(f_t)}) . \qquad \textbf{(6)}$$

Here f_t =1 indicates the presence of feature F_t in a document. $P(C_k)$ is the number of feature occurrences in documents with class label C_k divided by the total number of feature occurrences; $P(f_t)$ is the number of occurrences of feature F_t divided by the total number of feature occurrences; and $P(C_k, f_t)$ is the number of feature occurrences of F_t in documents with class label C_k divided by the total number of feature occurrences. Based on the $I(C; f_t)$ value a certain number of textual features are selected from the vocabulary as attributes that will be used in classification. For each document, the number of occurrences of a selected textual feature is the attribute value.

4 Experiments and Results

To have better understanding of the issues related to building a personal E-mail filter and the behavior of such filters, a series of experiments were conducted using both the naïve Bayesian classifier and the decision tree based classifier.

4.1 Data Sets and Performance Metrics

In the experiments, E-mail messages were used as document samples. The characteristics of collected data sets are shown in Table 1.

Table 1. Data Samples Used in Experiments

Source of data sets	5 (2 professors, 3 graduate students)
Number of data sets	11 (one set consists of E-mail messages in a month)
Size of each data set	250-700
Number of classes	2 (“not interesting”, “interesting”)

Every user who provided personal E-mail messages labeled all her/his messages as either interesting or not interesting. Since we did not give classification criteria to the person who provided the E-mail data, the classification was rather subjective. Unlike some other reported work, “not interesting” E-mail does not necessarily refer to commercial advertisements. For example, given two call-for-paper messages from international conferences in computer science, one may be classified as “not interesting” and the other as “interesting” depending on the theme of the conferences and the personal interest. Therefore, classifying an E-mail message as interesting or not is more challenging than pure commercial spam filtering.

During the experiments, each data set was divided into two portions: training data and test data in the chronicle order. The training data were used to train a classifier and the obtained classifier then classified the test data. Metrics used to measure the classification performance are defined as follows:

$$Error - rate = \frac{\#\ \mathit{false\ classification}}{\#\ \mathit{classified\ messages}}, \tag{7}$$

$$"\mathit{Interesting}"\ \mathit{recall} = \frac{\#"\mathit{in\ teresting}"\ \mathit{messages\ \ classified\ as}\ "\mathit{in\ teresting}"}{\#\ \mathit{total}\ "\mathit{in\ teresting}"\ \mathit{messages}}, \tag{8}$$

$$"\mathit{Interesting}"\ \mathit{precision} = \frac{\#"\mathit{in\ teresting}"\ \mathit{messages\ classified\ as}\ "\mathit{in\ teresting}"}{\#\ \mathit{total\ messages\ classified\ as}\ "\mathit{in\ teresting}"}. \tag{9}$$

"Not interesting" recall and precision are defined likewise. In the application of a personal E-mail filter, considering the information loss by misclassifying an "interesting" message as "not interesting", we emphasize the "interesting" recall and the error rate in the following tests.

4.2 Precision-Recall Graph of the Naïve Bayesian Classifier

The implemented Bayesian classifier classifies an E-mail message as "not interesting" only if the probability of "not interesting" is greater than threshold α ($\alpha \geq 0.5$). The value of the threshold in fact reflects the relative cost of misclassifying an "interesting" message as "not interesting". High α means high cost of such misclassification. Therefore, α is an important design parameter for a naïve Bayesian classifier. The first experiment aimed at the general behavior of the classifier when different threshold values were used. All three types of features were generated. By varying the threshold from 0.5 to 0.999, different recall and precision pairs were obtained for both "not interesting" and "interesting" classes. The average of 11 data sets was used to draw the recall-precision graph as shown in Fig. 2.

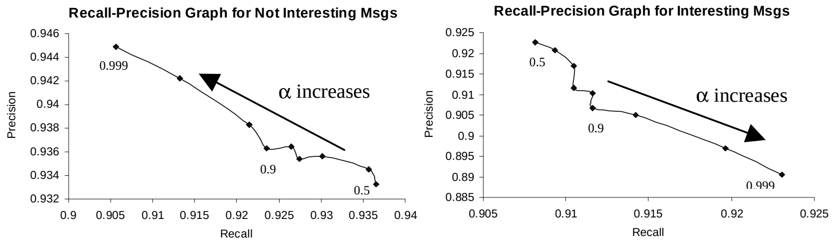

Fig. 2. Recall-Precision Graphs

A few observations can be made. First, within a wide range of the threshold value, both recall and precision values are around or above 90%. Second, the recall-precision curve of "not interesting" E-mail is better than that of "interesting" E-mail. Seemingly it is easier for a Bayesian classifier to identify "not interesting" E-mail messages because they are often obviously characterized by some features. Finally, the rate at which recall and precision change is different from the rate at which threshold α changes. For "interesting" E-mail, when α increases from 0.5 to 0.9, the recall increases slowly by 0.35%. However when α increases from 0.9 to 0.999, the recall increases by 1.2%. Likewise "not interesting" recall decreases slowly as α changes from 0.5 to 0.9 but much faster when α changes from 0.9 to 0.999. Therefore, in order to obtain high recall of "interesting" E-mail α should be set a relatively high value, say higher than 0.9. In the following experiments, we used 0.99 as the default setting for α.

4.3 Experiments on Feature Selection

As we mentioned earlier, three types of features are generated for both classifiers. One question under exploration is how important these features are in E-mail classification. The second set of experiments was conducted to study the performance of the classifiers when different types of features were used. Fig. 3 and Fig. 4 depict the results of 11 data sets using the Bayesian classifier and the decision tree based classifier, respectively. In the figures, H stands for header features only, T for textual features only, HT for header features plus textual features, HH for header features plus handcrafted features, TH for textual features plus handcrafted, and HTH for header, textual and handcrafted features, namely all features. H, T, HT were three baselines. HH, TH, HTH were tested to detect the change of performance by adding handcrafted features to those baselines. The average of 11 groups was used for evaluation in terms of three accuracy metrics, error rate, "not interesting" recall and "interesting" recall.

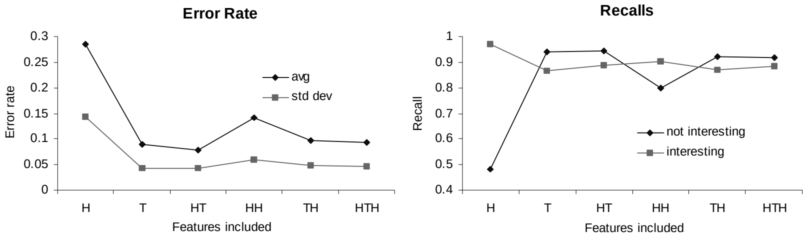

Fig. 3. Effects of Selected Features on the Naive Bayesian Classifier

Fig. 3 shows that, when only the structured features (H) are considered, the error rate is very high. If the structured features and the handcrafted features are selected (HH), the error rate is still quite high. However in these cases, the "interesting" recall is unexpectedly high. The reason lies in the small number of features involved in classification, only six or twelve. When only a small number of features in a document are selected, the difference between $P(D'|C_k)$ with different class label k is outweighed by the denominator α. High α leads to a good "interesting" recall.

However, error rate is high and “not interesting” recall is low, indicating these two feature selection methods are not appropriate to a Bayesian classifier. All other four cases involve the textual features. The best performance is achieved using the structured and the textual features (case HT). Adding header features better performs both case (T) and case (TH). However, comparing cases T and TH, HT and HTH, we find that, adding handcrafted features in fact does not improve the performance of case (T) and worsens that of (HT).

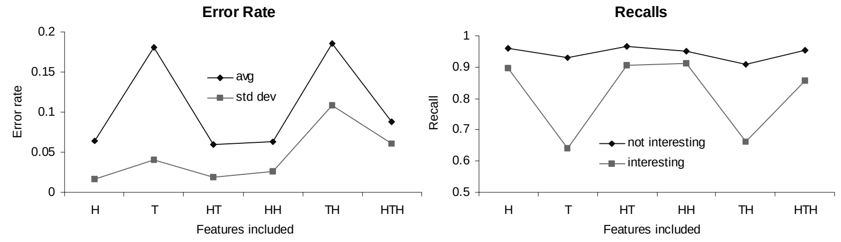

Fig. 4. Effects of Feature Selection on the Decision Tree Based Classifier

Fig. 4 shows the performance of the decision tree based classifier when different features are included in classification. Its performance is bad when the structured (header) features are not included. Therefore these two types of feature selection are not appropriate. On the baseline of H, adding either textual features or handcrafted features enhances the performance. However, when both textual features and handcrafted features are added to the baseline, the performance deteriorates, esp. in “interesting” recall and error rate. With all features included, the database schema consists of 32 attributes: 6 header features, 6 handcrafted features and 20 textual features. Decision tree becomes inaccurate with so many attributes. It works best with the selection method HT or HH.

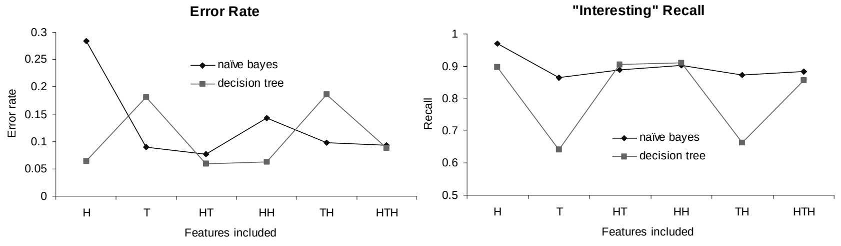

Fig. 5. Comparison in Feature Selection

Fig. 5 presents the average accuracy of the two classifiers in terms of error rate and “interesting” recall. Both the Naïve Bayesian classifier and the decision tree based classifier perform best with header features and textual features. The method of combining these two types of features for classification is useful. Neither of the classifiers works best with all features selected. One lesson we learned is that adding many features does not necessarily enhance the performance. Cautions should be taken in feature selection. In the optimal case, decision tree beats Bayesian based classifier in error rate and "interesting" recall.

4.4 Experiments on Robustness of Classifiers

We also conducted a set of experiments aiming to discover the robustness of both classifiers on different conditions that may happen in the real use of a personal E-mail filter. Limited by space, we just summarize the results without going into details.

Training size is an important issue that affects the accuracy of classifiers. From the experimental results we find when the training size is less than the test size, the decision tree based classifier has much lower "interesting" recall and higher error rate than the Bayesian classifier. It shows decision tree has a sparse data problem. As the training size grows, both classifiers improve the performance. In the optimal case decision tree outperforms naïve Bayesian. But a Bayes classifier keeps a reasonable performance on most conditions and has better performance when only a small training size is available.

Both classifiers can be affected by class disparity. Naïve Bayes classifier favors the major class by the factor of class prior in the decision rule. Decision tree based classifier chooses the major class at each test. Real users can have any ratio of "not interesting" messages to "interesting" messages. This experiment aimed to find out how these two classifiers perform as the class disparity of training data changes. The results show that the naïve Bayes classifier works well when "interesting" E-mail messages cover from 30% to 80% of the total training messages. The decision tree based classifier has high error rate at both ends of "interesting" coverage and the general performance is not stable.

5 Conclusion

This paper models E-mail messages as a combination of structured fields and free text fields, which motivated the work of classifying such documents deploying both kinds of information. Certain heuristic features obtained from preprocessing the documents were also included for the purpose of classifying E-mail messages. A comprehensive comparative study was conducted using a naïve Bayesian based classifier and a decision tree based classifier. Different ways of feature selection for both models were evaluated. Performance of two classifiers was compared with respect to training size and class disparity. By a series of experiments on real personal data, we find that both classifiers can be used to classify E-mail messages with reasonable accuracy. In the optimal cases, decision tree based classifier outperforms Bayesian classifier, but Bayesian is more robust on various conditions. Careful feature selection from structured fields and free text body enhances performance.

The study reported in this paper can be extended in three directions. First, due to the personalized nature of electronic mail, the test data available is only moderately large. We are trying to collect more data from different types of users. It will deepen our study and enable more findings about how to achieve an effective personal E-mail filter. Second, we are exploring the ways of combining these two types of classifiers in feature selection and decision making, which might lead to a more accurate classification method in this problem domain. Last, we plan to expand the classification from two classes to multiple classes and further to a hierarchy of classes, which will better serve the need of E-mail users.

References

1. William W. Cohen: Learning Rules that Classify E-mail. In *Proceedings of the 1996 AAAI Spring Symposium on Machine Learning in Information Access*
2. W. W. Cohen, Y. Singer: Context-Sensitive Learning Methods for Text Categorization. In *Proceedings of SIGIR-1996*
3. M. Craven, D. DiPasquo, D. Freitag, A. McCallum, T. Mitchell, K. Nigam and S. Slattery: Learning to Extract Symbolic Knowledge from the World Wide Web. In *Proceedings of the 15th National Conference on Artificial Intelligence* (AAAI-98)
4. Fredrik Kilander: Properties of Electronic Texts for Classification Purposes as Suggested by Users. http://www.dsv.su.se/~fk/if_Doc/F25/essays.ps.Z
5. D. D. Lewis: Naïve (Bayes) at Forty: The Independent Assumption in Information Retrieval. In *European Conference on Machine Learning*, 1998
6. D. D. Lewis, K. A. Knowles: Threading Electronic Mail: A Preliminary Study. In *Information Processing and Management*, 33(2): 209-217, 1997
7. D. D. Lewis, M. Ringuette: A Comparison of Two Learning Algorithms for Text Categorization. In *Third Annual Symposium on Document Analysis and Information Retrieval*, pp. 81-93, Las Vegas, NV
8. Andrew McCallum and Kamal Nigam: A Comparison of Event Models for Naïve Bayes Text Classification. *Working notes of the 1998 AAAI/ICML workshop on Learning for Text Categorization*
9. J. R. Quinlan: Induction of Decision Trees. *Machine Learning*, 1: 81-106, 1986
10. J. R. Quinlan: *C4.5: Programs for Machine Learning*. San Mateo, Calif.: Morgan Kaufmann Publishers, 1993
11. M. Sahami, S. Dumais, D. Heckerman, E. Horvitz: A Bayesian Approach to Filtering Junk E-mail. In *Learning for Text Categorization: Papers from the 1998 workshop*. AAAI Technical Report WS-98-05
12. Ellen Spertus: Smokey: Automatic Recognition of Hostile Messages. In *Proceedings of Innovative Applications of Artificial Intelligence* (IAAI) 1997