# Filtering Junk Mail with A Maximum Entropy Model

ZHANG Le and YAO Tian-shun
Institute of Computer Software & Theory.
School of Information Science & Engineering, Northeastern University
Shenyang, 110004 China
Email: ejoy@xinhuanet.com, tsyao@mail.neu.edu.cn

**Abstract**

The task of junk mail filtering is to rule out unsolicited bulk e-mail (junk) automatically from a user's mail stream. Two classes of methods have been shown to be useful for classifying e-mail messages. The rule based method uses a set of heuristic rules to classify e-mail messages while the statistical based approach models the difference of messages statistically, usually under a machine learning framework. Generally speaking, the statistical based methods are found to outperform the rule based method, yet we found, by combining different kinds of evidence used in the two approaches into a single statistical model, further improvement can be obtained.

We present such a hybrid approach, utilizing a Maximum Entropy Model, and show how to use it in a junk mail filtering task. In particular, we present an extensive experimental comparison of our approach with a Naive Bayes classifier, a widely used classifier in e-mail filtering task, and show that this approach performs comparable or better than Naive Bayes method.

**Key Words: Junk Mail Filtering, Maximum Entropy Model, Text Categorization**

ICCPOL2003, pp. 446-453, ShenYang (China)

# 1 Introduction

In this era of rapid information exchange, electrical mail has proved to be an effective means to communicate by virtue of its high speed, reliability and low cost to send and receive. While more and more people are enjoying the convenience brought by e-mail, an increasing volumes of unwanted junk mails (spam) have found their way to users' mailboxes. Junk mail, also called unsolicited bulk e-mail, is Internet mail that is sent to a group of recipients who have not requested it. These unsolicited mails have already caused many problems such as filling mailboxes, engulfing important personal mail, wasting network bandwidth, consuming users' time and energy to sort through it, no to mention all the other problems associated with spam (crashed mail-servers, pornography adverts sent to children, etc). According to the Direct Marketing Association's 1996 Statistical Fact Book the Americans get 21.31 pieces of direct mail per week, 43% wish they got less, 52.2% order something from it, and 46% of it is never read. With $30 billion spent on direct mail, that's a waste of about $15 billion annually. Therefore it is necessary to eliminate these unwanted mails automatically before they enter a user's mailbox.

Different methods have been used for automatically email classification. (Cohen, 1996) uses a system which learns a set of "keyword-spotting rules" based on the RIPPER rule learning algorithm to classify emails into pre-defined categories. The performance is comparable to traditional TF-IDF weighting method. In another approach, differences between legitimate mails and junk mails are modeled statistically, by recasting this problem into a Machine Learning framework. (Sahami et al, 1998) uses a Bayesian classifier and trains their system on a corpus of 1789 email messages among which 1578 messages are pre-classified as "junk" and 211 messages are labeled as "legitimate". Using this method, they are able to achieve a legitimate precision of 96.2% and a 100% junk precision. (Androutsopoulos et al, 2000a) gives a thorough evaluation of Naive Bayesian Anti-Spam filtering method and investigates the effect of attribute-size, training-corpus size, lemmatization, and stop-lists on the filter's performance. (Androutsopoulos et al, 2000b) also investigates the performance of a Memory Based Learner on the junk filtering task. They got an accuracy comparable to a Naive Bayes classifier. (Carreras and Mrquez, 2001) trains a Boosting tree classifier to filter junk mail using Ada Boost algorithm. They report rates of correctly classified messages which are comparable to that presented by (Sahami et al, 1998).

A purely statistical approach expresses the differences among messages in terms of the likelihood of certain events. The probabilities are usually estimated automatically to maximize the likelihood of generating the observations in a training corpus. Hence a statistical model is easy to build and can adapt to new domains quickly. However, the mathematical model used in a purely statistical system often lacks deep understanding of the problem domain, it simply estimates parameters from training corpus. So a model performs well on one corpus may work badly on another one with quite different characteristics. A rule based approach, on the other hand, expresses the domain knowledge in terms of a set of heuristic rules, often constructed by human experts in a compact and comprehensive way. This approach has the advantage of expressing complex domain knowledge usually hard to be obtained in a purely statistical system. For instance, heuristics such as "the message contains some java-scripts for form validation", may be used in filtering our junk messages which contain a HTML register form. However, building a rule based system often involves acquiring and maintaining a huge set of rules with an extremely higher cost compared to the purely statistical approach. And such system is hard to scale up.

A better way may be combining the advantages of both approaches into a single statistical model. In this paper, we present a Maximum Entropy (ME) hybrid approach to junk mail filtering task, utilizing a Maximum Entropy probabilistic model. Maximum Entropy Modeling (Berger et al, 1996) has been used in the area of Natural Language Processing for several years. The performance of Maximum Entropy Model has been shown to be comparable to that of other statistical modeling methods or even better (Ratnaparkhi, 1998). We start by building a simple ME filter to the task which uses only word feature, serving as a baseline for comparison with our hybrid method. We then improve the baseline model by adding domain specific evidence into it. In addition, we also compare the performance of the ME model with a Naive Bayes classifier.

In the rest of this paper, ME model is introduced in section 2. Section 3 describes the feature selection method used in our model. Section 4 gives an evaluation of our approach. Finally, we draw our conclusion in section 5.

## 2 Maximum Entropy Model

Maximum Entropy (ME) models have been successfully applied to various Natural Language Processing tasks including sentence boundary detection, part-of-speech tagging, prepositional phrase attachment and adaptive statistical language modeling with the state-of-the-art accuracies (Ratnaparkhi, 1998; Rosenfeld, 1996). The goal of the ME principle is that, given a set of features, a set of functions $f_1 \dots f_m$ (measuring the contribution of each feature to the model) and a set of constrains, we have to find the probability distribution that satisfies the constrains and minimizes the relative entropy (Divergence of Kullback-Leibler) $D(p||p_0)$, with respect to the distribution $p_0$. In general, a conditional Maximum Entropy model is an exponential (log-linear) model has the form:

$$p(a|b) = \frac{1}{Z(b)} \prod_{j=1}^{k} \alpha_j^{f_j(a,b)} \tag{1}$$

where $p(a|b)$ denotes the probability of predicting an *outcome* $a$ in the given *context* $b$ with constraint or "feature" functions $f_j(a, b)$. Here $k$ is the number of features and $Z(b) = \sum_a \prod_{j=1}^{k} \alpha_j^{f_j(a,b)}$ is a normalization factor to ensure that $\sum_a p(a|b) = 1$, the parameters $\alpha_j$ can be derived from an iterative algorithm called Generalized Iterative Scaling (Darroch and Ratcliff, 1972). ME model represents evidence with binary functions known as *contextual predicates* in the form:

$$f_{cp,a'}(a, b) = \begin{cases} 1 & \text{if } a = a' \text{ and } cp(b) = true \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

where *cp* is the *contextual predicate* which maps a pair of *outcome* $a$ and *context* $b$ to $\{true, false\}$.

By pooling evidence into a "bag of features", ME framework allows a virtually unrestricted ability to represent problem-specific knowledge in the form of *contextual predicates*. Compared to Naive Bayes classifier, a commonly used classifier in Text Categorization (Mitchell, 1997) , which imposes strong conditional independence assumptions about the observed evidence, the features in a Maximum Entropy model need not to be statistically independent, therefore is is easy to incorporate overlapping and interdependent features. Thus ME models often yield better probability estimates for the observed distribution when statistical evidence from many sources must be combined freely.

While the use of ME model is computational straightforward, the main computational burden is the GIS parameter estimation procedure which involves computation of each observed expectation $E_p f_j$ and re-computation of the model's expectation $E_p f_j$ on each iteration. Though how many iterations are needed for the model to converge to an optimal solution $p^*$ is theoretically unknown, we found 100 iterations are good enough in experiments conducted in this paper. So all experiments reported here will use a 100 iterations for GIS algorithm.

# 3 Feature Selection

Two kinds of features are used in our model: one is single word (term) feature contained in messages, the other is domain specific feature. The former captures the characteristics of the messages to be classified, which is commonly used in Text Categorization, while the later reflects some domain specific properties of the task in hand.

## 3.1 Term Feature

When extracting term feature, one of the standard convention of the Text Categorization literature is to stem words to their morphological base forms, e.g. both "clicking" and "clicked" are reduced to root word "click". However, our preliminary experiments showed that non-stemming version of our classifier performs slightly better than stemmed one. So we do not stem words in our experiments. Another factor that we found helps improve classification accuracy is tokenizing scheme. We consider certain punctuations like exclamation points as part of a term. Since spammers tend to use phrases like "click here!","FREE!!!". Also special terms such as url, ip address are not splitted and treated as single terms. For we found things like "click http://xxx.xxx.com" have a high frequency in spam messages. Certain HTML tags are preserved too. For example, HREF and COLOR attributes are preserved since many HTML spam messages contain links to spammer's sites in conspicuous color in order to catch the reader's attention.

Another way to enhance performance is to extract terms not only from message body but from message headers as well. Message headers carry some important information such as the sender's ip address, the server used for relaying etc. which is found to be helpful in identifying junk mails, though normal users do not pay much attention to them.

Words occur in the To, From, Subject lines are treated specially. For instance, "free" occurs in Subject lines like "FREE PC!" is considered a different term from "free" appears in message body. Because messages contain "free" in Subject line like the above example tend to have a higher chance to be spams than messages only have "free" in body text.

## 3.2 Domain Specific Feature

On the task of filtering junk mail, domain special features are also precious and should not be treated with ignorance. Junk mails differ from legitimate mails not only in the boast (or offending) words they use, but also in their strange behaviors in various stages of dispatching.

When filtering junk mail, it is important to consider some particular features in the header field of a mail which give strong evidence whether a mail is junk or not. For instance, junk mails are seldom sent through normal email client such as Outlook Express or Mutt. Instead, spammers prefer to use some group mail sending software specially designed for sending mails to a large amount of recipients. This can be detected by looking at the X-Mailer field in the header. If a mail has an X-Mailer field indicating some group sending software or does not have X-Mailer field at all, it is very likely to be a spam. In addition, a good many non-textual features commonly found in spam messages can serve as good indicators to rule out spams. For example, junk mails often do not have user name in their From, To fields or simply use "Dear User" as user name. This can be identified by matching header fields with a pre-defined rule set. Also if all the words in Subject field are capitals, it probably indicates a spam. Entire HTML message containing only one relaying command trying to load an outside url (often spam site) when reading is a new kind of trick played by spammers.

We use SpamAssassin [1] , a rule based anti-spam software, to extract these non-textual, domain specific features. SpamAssassin attempts to identify spam using a wide range of heuristic rules and tests constructed by human experts. Table 1 lists top 10 domain specific features[2] and their frequencies found in legitimate and spam messages of our corpus respectively. (Sahami et al, 1998) also uses domain specific features in their Naive Bayes classifier. Our approach differs from theirs in the number of domain specific features used, SpamAssassin has a rich rule set of more than 1000 hand-crafted rules while Sahami's system uses only 20 rules generated during a short brainstorming meeting.

## 3.3 Feature Selection Method

For Text Categorization task, the performance rests heavily on the features selected. A $\chi^2$-test is used for feature selection. $\chi^2$-test is found to be an effective feature selection method for Text Categorization task (Yang and Pedersen, 1997). The $\chi^2$-test measures is defined to be:

$$\chi^2(f, c) = \frac{N \times (AD - CB)^2}{(A + C) \times (B + D) \times (A + B) \times (C + D)}$$

[1]See http://spamassassin.org. At the time of writing this paper, a Bayesian filter is also included in the latest version of this software.

[2]Please refer to the document of SpamAssassin for the meaning of these features.

| features of spam | frequency | features of legitimate | frequency |
|---|---|---|---|
| CLICK_BELOW | 1067 | SPAM_PHRASE_00_01 | 4329 |
| NO_REAL_NAME | 1029 | KNOWN_MAILING_LIST | 3753 |
| CTYPE_JUST_HTML | 929 | IN_REP_TO | 2792 |
| BIG_FONT | 849 | QUOTED_EMAIL_TEXT | 2390 |
| FROM_ENDS_IN_NUMS | 641 | REFERENCES | 2257 |
| HTML_FONT_COLOR_RED | 595 | MSG_ID_ADDED_BY_MTA_3 | 1329 |
| SPAM_PHRASE_08_13 | 579 | INVALID_DATE | 1289 |
| CLICK_HERE_LINK | 519 | SPAM_PHRASE_01_02 | 1114 |
| HTML_FONT_COLOR_BLUE | 515 | USER_AGENT | 1045 |
| EXCUSE_3 | 508 | EMAIL_ATTRIBUTION | 865 |

Table 1: Top 10 domain specific features of spam and legitimate messages

where A is the number of times feature $f$ and category $c$ co-occur. B is the number of times of $f$ occurs without $c$. C is the number of times $c$ occurs without $f$. D is the number of times neither $c$ or $f$ occurs and N is the total number of documents (messages).

Once a feature set is defined, it is straightforward to incorporate the features into the Maximum Entropy model in a "bag of features" manner. All features will have *context predicates* in the form:

$$cp_f(b) = \begin{cases} true & \text{if message } b \text{ contains feature } f \\ false & \text{otherwise} \end{cases} \tag{3}$$

Therefore all features have the form:

$$f(a,b) = \begin{cases} true & \text{if } cp_f(b) = true \\ false & \text{otherwise} \end{cases} \tag{4}$$

here $a$ is the possible category {spam,legitimate} of message $b$.

# 4 Evaluation

In this section, we report empirical results achieved with the Maximum Entropy model on the junk filtering task. We also compare the performance of ME model with a Naive Bayes classifier, a widely used classifier in e-mail filtering task.

All experiments described here are performed on a public spam corpus[3], which contains 9351 messages of which: 2400 are labeled as spam and 6951 are marked as legitimate (ham), with a spam rate 25.7%.

A 10-fold cross-validation approach is employed in our experiments in which the corpus is partitioned ten times into 90% training material, and 10% testing material in order to minimize random variation.

## 4.1 Evaluation Measures

We first introduce the evaluation measures used in this paper here. Let $S$ and $L$ stand for spam and legitimate respectively. $n_{L\to L}, n_{S\to S}$ denote the numbers of legitimate and spam messages correctly classified by the system. $n_{L\to S}$ represents the number of legitimate messages misclassified as spam, $n_{S\to L}$ is the number of spam messages wrongly treated as legitimate. Then precision(p), recall(r), error-rate(e) and $F_\beta$-measure are defined as follows:

$$precision(p) = \frac{n_{S\to S}}{n_{S\to S} + n_{L\to S}} \tag{5}$$

$$recall(r) = \frac{n_{S\to S}}{n_{S\to S} + n_{S\to L}} \tag{6}$$

$$error(r) = \frac{n_{L\to S} + n_{S\to L}}{\text{the number of all messages}} \tag{7}$$

$$F_\beta = \frac{(\beta^2+1)pr}{\beta^2 p + r} \tag{8}$$

[3]http://spamassassin.org/publiccorpus. We removed a few messages which contain very big binary attachments, say more than 200K, from original corpus, since we only consider textual feature so far.

where $\beta$ is the parameter allowing differential weighting of $p$ and $r$. When the value of $\beta$ is set to 1, *recall* and *precision* are weighted equally:

$$F_1(p, r) = \frac{2pr}{p + r} \tag{9}$$

### 4.2 Performance

Our goal here is to measure the performance of our ME model and whether incorporating domain specific features helps improve classification accuracy. We also evaluate the performance of ME model by comparing it with a Naive Bayes classifier. To this end, we first train and test a Naive Bayes classifier with word (term) feature only, serving as the baseline for comparison. On the same corpus, we then train and test a standard Maximum Entropy model (ME-standard) with the same feature set as baseline Naive Bayes model and an enhanced model (ME-enhanced) with additional domain specific features. A message is classified as spam if $p(spam|message) > 0.5$[4]. The results of the three models are given in Table 2. Figure 1 to 4 illustrate the performances of the three models on data sets of different sizes. Training set size is on the X-axis, performance measures (precision, recall, error-rate and $F_1$ measure) are on the Y-axis, respectively.

| model | junk precision | junk recall | error-rate | $F_1$ |
|---|---|---|---|---|
| NB(baseline) | 99.67% | 96.58% | 0.98% | 98.09% |
| ME | 99.83%(0.16%) | 97.37%(0.82%) | 0.73%(-25.51%) | 98.59%(0.51%) |
| ME-enhanced | 99.83%(0.16%) | 97.74%(1.20%) | 0.63%(-35.71%) | 98.77%(0.69%) |

Table 2: Filtering performance of different models
(the number in parenthesis indicates improvements over baseline NB model)

The enhanced model (ME-enhanced) clearly outperformed the baseline model (NB) and standard ME model (Table 2). Naive Bayes classifier has a higher precision than the two ME models when training data set is small. As the size of training data set increases all methods achieve very high precision, and ME models perform slightly better than Naive Bayes (Figure 1). In addition, ME models achieve a higher recall, a lower error-rate and a better overall $F_1$ measure on all kinds of data set sizes we tried (Figure 2 to 4). The ME-enhanced model has a 1.20% enhancement in recall compared to baseline Naive Bayes model and 35.71% reduction in error-rate, 0.69% enhancement in $F_1$ measure, respectively. We argue that it is the ME model's ability of freely incorporating overlapping and interdependent features that makes it superior to Naive Bayes on this task.

Figure 1: Junk Precision

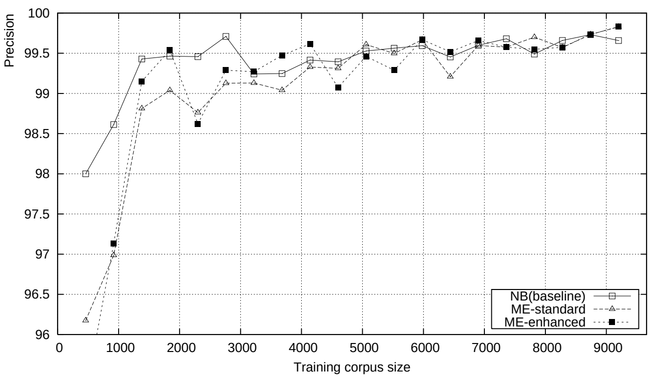
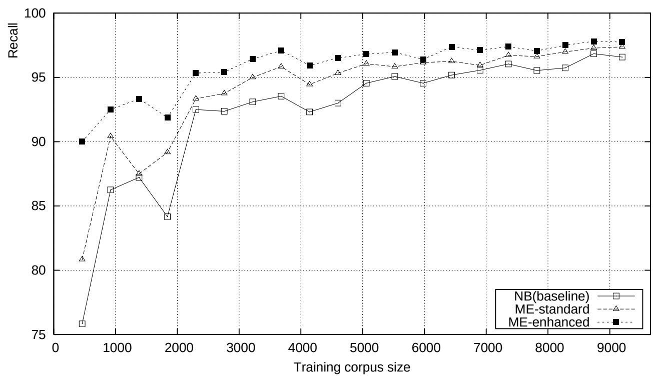



Figure 2: Junk Recall




# 5 Conclusion

In this paper, a Maximum Entropy based method is presented to filter junk mail. A comparison of the classification results with a Naive Bayes classifier showed that the ME method performs comparable or better than NB approach.

[4]When applied to real filtering system this threshold can be adjusted to a higher value so as to reduce the chance a legitimate message being misclassified as spam(false positive).

Figure 3: Error rate

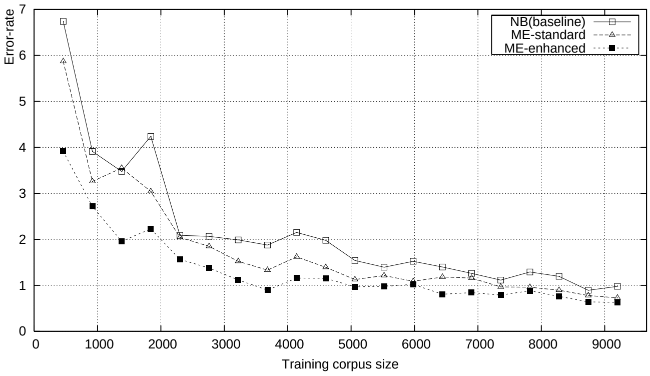
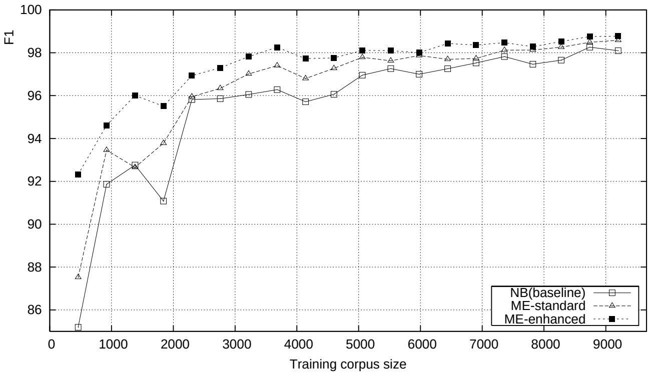



Figure 4: $F_1$ measure




We also showed the classification accuracy can be further improved by considering domain specific features to the task. By combining some of the best features of domain specific property (usually used in a rule based system) and statistical term feature, ME model performs fairly well even on small training corpora in terms of $F_1$ measure. We believe the ME model's ability of freely incorporating evidence from different sources makes it perform better than Naive Bayes classifier, which suffers from conditional independence assumptions.

A disadvantage of the presented method may be its lower training speed compared to Naive Bayes method. With the rapid development of computational power, however, this does not seem to be a serious drawback. Another downside of ME model is lacking the ability of incremental learning. When applied to a real mail filtering system, it is necessary to adjust the model's parameters in order to reflect the characteristics of newly arrived e-mail messages, since the characteristics of junk messages can change over time. Periodically training the model on newly collected messages may be an solution, though not as flexible as methods which can perform incremental learning such as Naive Bayes and Memory Based Learning.

As a future research line, we would like to explore better feature selection techniques. For instance, considering word n-gram feature or even some grammatical feature which may further enhance the filtering performance.

# Acknowledgments

This research was partially funded by the National Natural Science Foundation of China under Grant No. 60083006 and the National Grand Foundational Research 973 Program of China under Grant No. G19980305011.

# References

I. Androutsopoulos, J. Koutsias, K. V. Chandrinos, G. Paliouras, and C. D. Spyropoulos. An Evaluation of Naive Bayesian Anti-Spam Filtering. In Proc. of the workshop on Machine Learning in the New Information Age, 2000.

I. Androutsopoulos and G. Paliouras and V. Karkaletsis and G. Sakkis and C. Spyropoulos and P. Stamatopoulos. Learning to Filter Spam E-Mail: A Comparison of a Naive Bayesian and a Memory-Based Approach. In Workshop on Machine Learning and Textual Information Access, 4th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD).

Berger, Adam L. and Della Pietra, Stephen A. and Della Pietra, Vincent J. A Maximum Entropy Approach to Natural Language Processing. Computational Linguistics,Vol. 22,No. 1,1996,pp. 39–71

X. Carreras and L. Mrquez, Boosting trees for anti-spam email filtering. In Proceedings of RANLP-01, 3rd International Conference on Recent Advances in Natural Language Processing, Tzigov Chark, BG, 2001.

Cohen, W. W. Learning Rules that Classify E-mail. In Proceedings of the 1996 AAAI Spring Symposium on Machine Learning in Information Access, Stanford, California.

Darroch, J.N. and Ratcliff, D. Generalized Iterative Scaling for Log-Linear Models. The Annals of Mathematical Statistics, Vol. 43, pp 1470–1480, 1972.

A. Ratnaparkhi. Maximum Entropy Models for Natural Language Ambiguity Resolution. Ph.D. thesis, University of Pennsylvania, Philadelphia, PA, 1998.

R. Rosenfeld. A Maximum Entropy Approach to Adaptive Statistical Language Modeling. Computer, Speech and Language, vol. 10, pp. 187– 228, 1996. Longer version: Carnegie Mellon Tech. Rep. CMU-CS-94-138.

Mitchell, T. M. Machine Learning. McGraw-Hall, 1997.

M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz. A Bayesian Approach to Filtering Junk E-Mail. Learning for Text Categorizaton - Papers from the AAAI Workshop,1998, pages 55-62, Madison Wisconsin. AAAI Technical Report WS-98-05.

Yiming Yang, Jan O. Pedersen. A Comparative Study on Feature Selection in Text Categorization. Proceedings of ICML-97, 14th International Conference on Machine Learning, 1997.