Optimising Naïve Bayesian Networks for Spam Detection

Henry Stern

Faculty of Computer Science, Dalhousie University
6050 University Avenue, Halifax, Nova Scotia, Canada, B3H 1W5
stern@cs.dal.ca

Abstract. In 2001, Spam e-mails accounted for 8% of all e-mails sent over the internet. By 2002, that number had risen to 36%. Despite these staggering numbers, there are only about 150 people responsible for the bulk of spam e-mail. Because of this, there are many words and phrases common to most spam e-mails that do not occur in desirable e-mail messages. A naïve Bayesian classifier can be employed to detect these spam messages, but the computing costs associated with classifying a large number of messages can be very high. To resolve this problem, two modifications are made to the classifier. The computation is simplified using unique classifiers for each message. In addition, the vocabulary used by the classifiers is reduced using an entropy-based technique. These modifications reduce the cost of the classification and reduce the error rate of the classifier.

1 Introduction

In 2001, it was estimated by the anti-spam firm, Brightmail[1], that "Spam" e-mails accounted for 8% of all e-mails sent over the internet. By 2002, that number had risen to 36% and is projected to rise even further in the future. Spam e-mail typically promotes fraudulent businesses or sells pharmaceuticals and pornography. Spamhaus[2], an anti-spam organization, maintains the "Register of Known Spam Operations," a set of dossiers on over 100 individuals and organisations responsible for the majority of all spam e-mails.

Because of the limited number of authors of spam e-mails, statistical natural language processing techniques can be applied to detect spam e-mails. Humans typically use language based on the principle of "least resistance." Each author will have a distinctive style that they use to write their messages and will repeatedly use specific sequences of words.

There is much interest in technical means to reducing the volume of spam e-mail. The Mail Abuse Prevention System LLC[3] publishes the Real-time Black List, a listing of internet service providers and individual e-mail servers that are

[1] Brightmail Inc.: http://www.brightmail.com/
[2] The Spamhaus Project: http://www.spamhaus.org/
[3] Mail Abuse Prevention System LLC: http://www.mail-abuse.org/

currently serving spam e-mails. This list is used by internet service providers to block incoming e-mails and web traffic to offending organizations and forces companies to abide by e-mail usage guidelines.

Another technical solution to the spam problem is Vipul's Razor[4], a collaborative spam-tracking database. Vipul's Razor exploits the fact that most spam mailings send the exact same message to every address. When a user or e-mail server classifies a spam e-mail, their "Reporting Agent" submits a digital signature for that message to a catalogue server where it is shared with other users who will block messages with the same signature.

Both the MAPS RBL and Vipul's Razor are "after-the-fact" solutions to the problem. The spam messages need to be received and classified before they can be submitted to either collaborative system.

Instead of relying on human decisions, much research has been done towards the automated classification of spam e-mail. SpamAssassin[5] is a rule-based classifier that tags e-mails based on a set of human-defined characteristics of spam e-mails. A great amount of effort is required to maintain the rule set and scoring system for this classifier.

Manually creating rules is both time consuming and, due to scope issues, can lead to false positives. Crawford [4] describes a technique for automatic rule induction for spam detection.

There has been extensive research on advanced machine learning techniques for spam detection. Carreras and Márquez have applied boosted decision trees [3] to this problem with good results. Androutsopoulos, Sahami et. al. have investigated the use of naïve Bayesian networks [1, 6] for spam detection. Androutsopoulos et. al. have also done a comparison of naïve Bayesian and memory-based learning [2]. Manco et. al. have applied clustering techniques [5] to better adapt to trends in e-mail messages over time.

The approach presented in this paper attempts to optimize the performance of Naïve Bayesian classifiers for spam detection. By intelligently reducing the feature set of dynamically-created classifiers, it may be possible minimize the computational cost of a classification and reduce over-fitting of the training data.

2 Methodology

The goal of this experiment is to minimize the computational cost and maximize the accuracy of the classifier. With the large volume of e-mail traffic on the internet, a high rate of classification and low storage requirements for probability tables is important for reducing the cost of widespread adoption of spam filtering. The Naïve Bayesian network has been optimized for speed in two ways: The computation performed by the classifier has been simplified and the feature set used to generate the classifiers has been reduced.

[4] Vipul's Razor: `http://razor.sourceforge.net/`
[5] SpamAssassin: `http://www.spamassassin.org/`

Two sets of experiments are performed. The first set uses probabilities of spam e-mail given individual terms and the second uses probabilities of spam e-mail given pairs of adjacent words (bi-grams).

Each set of experiments consisted of 20 experiments run using a ten-fold cross-validation with a corpus of 22000 e-mail messages. The probability threshold is varied for each experiment and the mean error rate of the classifier is observed.

2.1 Training Data

To satisfy the requirement of minimal false positives and cover the wide spectrum of e-mails that one may receive, a large training set of e-mail data was gathered.

In September and October 2002, several Dalhousie students and staff members saved the spam e-mails that they received in their various e-mail accounts. The majority of these e-mail messages were gathered from Dalhousie University and Microsoft "Hotmail" e-mail servers. Approximately 10000 spam e-mail messages were collected in this period. Due to the sheer volume of messages and multiple recipients, duplicate messages may exist in the corpus.

For the non-spam corpus, 15000 e-mail messages were gathered from the e-mail archives of the contributors. These messages consisted of personal and business correspondance and archives of informal and technical mailing lists. Due to the personal interests of the contributors, the subject matter of the messages was typically related to music, open source software, academic research or internet protocol standards.

Since the non-spam corpus was almost entirely English and entirely in the Latin character set, 3000 spam e-mail messages in foreign languages and international character encodings were removed from the spam corpus. These messages would have biased the classifier to classify all foreign language e-mails as spam and artificially inflate the observed precision and recall.

To pre-process the two data sets, the following steps were taken:

1. For all messages with a MIME type of `text/html`, the HTML tags were stripped from the message. The HTML tags were left in messages of MIME type `text/plain` since HTML being embedded within plain text messages was observed to be a common trait of spam e-mails.
2. Being only interested in the content contained within the body of the messages, the e-mail headers were discarded.
3. The text was normalized to be lower case.
4. Individual words were extracted from the e-mail messages, discarding all formatting information.

After pre-processing the messages, each word (or pair of words) is enumerated. Figure 1 shows the normalized distribution of the term frequencies. To normalize the plot, the X axis is the ratio of the rank to the total number of distinct terms and the Y axis is the ratio of the term frequency to the total number of term occurrences. The plot shows that the non-spam messages contain a wider variety of terms than the spam messages.

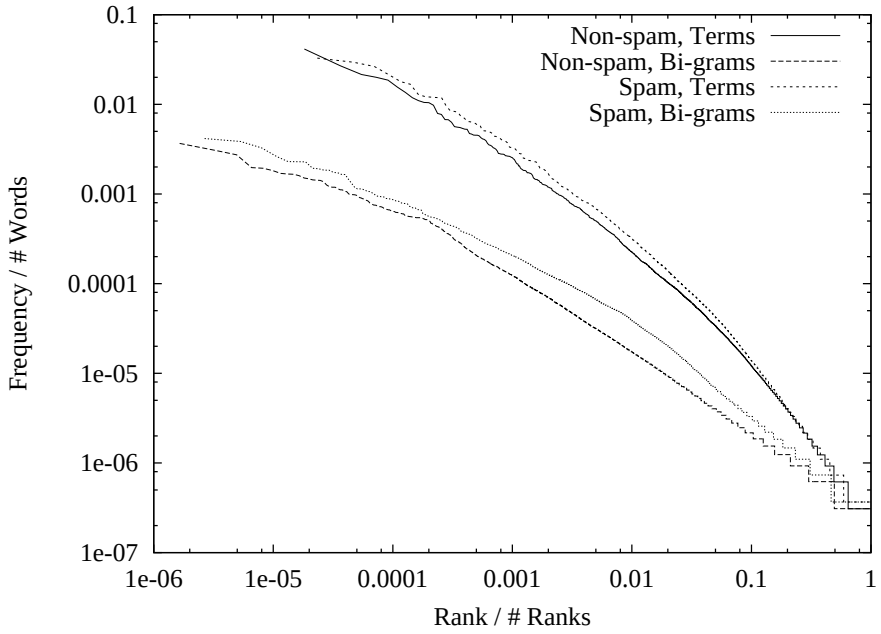

Fig. 1. The normalized rank-frequency plots of the data sets.

Table 1 confirms this observation. The non-spam corpus contains more unique words and almost twice as many unique bi-grams. This means that the spam messages contain many common sub-sequences in them.

The term frequencies from the corresponding spam and non-spam corpora are combined to generate the probability tables used by the classifier.

The frequency table, a set of tuples $f(t) \in \mathbb{N}^2$, is created where $f(t)_S$ is the number of occurrences of term t in the spam corpus and $f(t)_{\overline{S}}$ is the number of occurrences of term t in the non-spam corpus. The frequency tables are smoothed to account for words that occur in one corpus and not the other. If $f(t)_x = 0$ for any term t, then $f(t)_x \leftarrow 1$.

Corpus	Messages	Total Words	Unique Terms	Common Terms
Non-Spam (Tokens)	15363	3250387	54891	
Spam (Tokens)	7145	2730757	42837	
Combined	22508	5981144	79191	18537
Non-Spam (Bi-grams)	15363	3219680	607570	
Spam (Bi-grams)	7145	2716470	375819	
Combined	22508	5936150	891636	91753

Table 1. Statistics for the e-mail corpora.

The probability table, $p(S|t)$ is estimated for all terms in the vocabulary and is the ratio of the term frequencies in the spam corpus to the ratio of the term frequencies in both corpora. The probability table is defined in equations 1. $p(\overline{S})|t)$ is the complement of $p(S|t)$, defined in equation 2.

$$p(S|t) = \frac{f(t)_S}{f(t)_S + f(t)_{\overline{S}}} \tag{1}$$

$$p(\overline{S}|t) = 1 - P(S|t) \tag{2}$$

Using a kernel density estimation with a Gaussian kernel and bandwidth $h = 0.1$, figure 2 displays the estimated distribution of the values in the probability table. The density estimation function at x given data points $x_1, \ldots, x_n$ is defined as:

$$\widehat{f}_h(x) = \frac{1}{nh} \sum_{i=1}^{n} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2} \left(\frac{x_i - x}{h} \right)^2 \right) \tag{3}$$

The bump in the middle of the graph is composed of commonly used words such as "the" and "www," as well as words that occur only once or twice in the whole corpus. The frequencies were smoothed so the infrequently occurring terms will occur in the middle of the graph instead of on the edges.

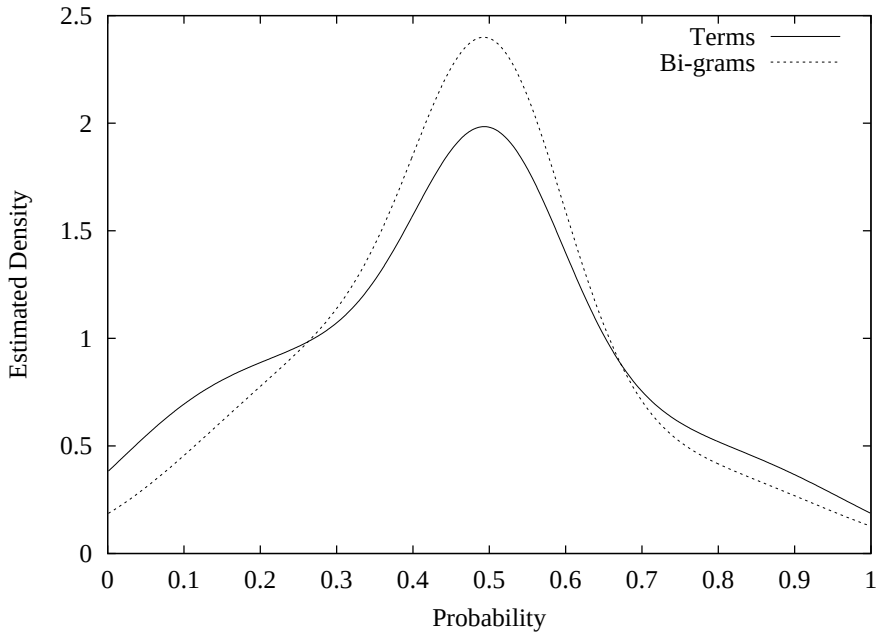

Fig. 2. The estimated density function for the probabilities.

2.2 Naïve Bayesian Classifier

The naïve Bayesian classifier performs the following computation to determine which category best suits a given input:

$$s^*(t_1, \ldots, t_n) = \underset{s \in \{S, \overline{S}\}}{\arg\max}\ p(s) \prod_{i=1}^{n} p(t_i|s) \tag{4}$$

Floating point addition is faster than floating point multiplication, therefore equation 4 was changed to:

$$\begin{aligned} s^*(t_1, \ldots, t_n) &= \underset{s \in \{S, \overline{S}\}}{\arg\max}\ \ln \left[p(s) \prod_{i=1}^{n} p(s|t_i) \right] \\ &= \underset{s \in \{S, \overline{S}\}}{\arg\max}\ \ln p(s) + \sum_{i=1}^{n} \ln p(t_i|s) \end{aligned} \tag{5}$$

Bayes' theorem states that $p(t_i|s) = p(s|t_i)\frac{p(t_i)}{p(s)}$. Because the equation is being taken as an arg max over s, $p(t_i)$ can be discarded. Equation 6 can now be changed to:

$$s^*(t_1, \ldots, t_n) = \underset{s \in \{S, \overline{S}\}}{\arg\max}\ (1-n) \ln p(s) + \sum_{i=1}^{n} \ln p(s|t_i) \tag{6}$$

Every classification will require $\mathrm{O}(n)$ additions, where n is the size of the feature set. The very large feature sets cause this to be computationally expensive.

Most e-mail messages use no more than a few hundred distinct words. If the vocabulary of the message is used to create the feature set of the naïve Bayesian network rather than the vocabulary of the corpus, the cost of classification will be very small. This is a trade-off between accuracy and computational efficiency and ignores the fact that a certain word not being present can be more of an indicator of spam than a specific word being present.

2.3 Feature Set Reduction

Figure 2 shows that the occurrence of most of the terms in the two vocabularies is not very indicative of whether a given message is spam. The two-class entropy function, shown in figure 3, describes the "randomness" of an entry in the probability table, $p(t)$.

The entropy for the entries in the probability table where $p(t)$ is close to 0.5 is very high. This means that the corresponding terms do not provide a very good clue as to whether a message is spam. If a probability threshold, $\pi \in \mathbb{R}[0.5 \ldots 1]$, is chosen and all terms where $1 - \pi < p(t) < \pi$ are pruned from the vocabulary, the classifier should produce similar results to a classifier using the full vocabulary while requiring fewer computations.

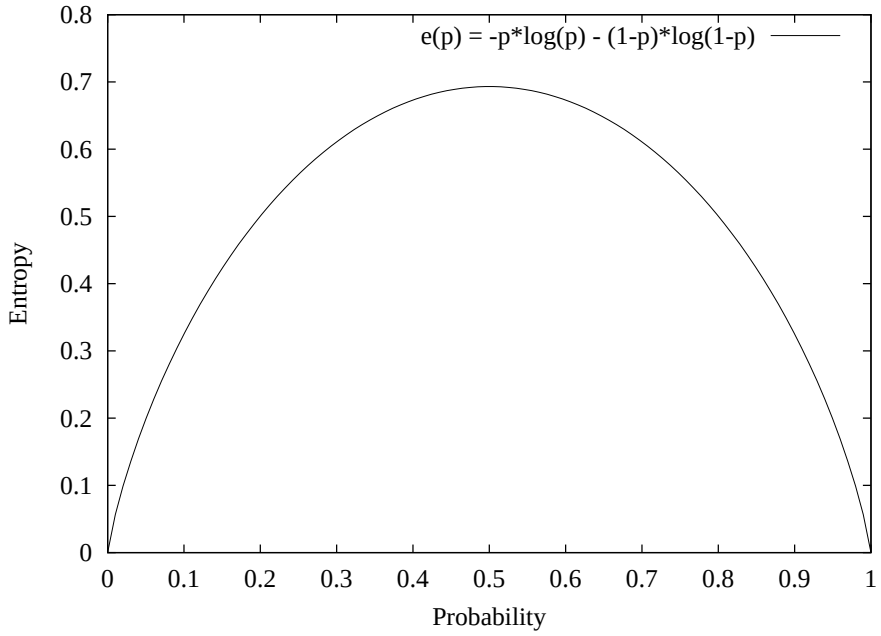

Fig. 3. The entropy function for a two-class system.

Figure 4 shows the mean proportional vocabulary size obtained using the data sets from a 10-fold cross validation. The mean vocabulary size given a specific threshold was normalized by dividing by the mean vocabulary size for $\pi = 0.5$.

The plot shows that a very large number of bi-grams occur in an approximately equal proportion in both sets. This is explained by Zipf's law: A very large number of terms occur a very few number of times.

The steps on the plot are due to the same phenomenon. The probabilities are ratios of integers and most terms occur very infrequently and in only one corpus. The steps occur when π is near $\frac{1}{2}$, $\frac{2}{3}$, $\frac{3}{4}$ and $\frac{4}{5}$.

3 Experimental Results

Figure 5 shows the observed error rate of the classifiers given the probability threshold, π, that was manipulated for each experiment. When π is near 0.5, the error rate tends to be large due to over-fitting of the training data. As π approaches 1.0, the higher error rate is due to over-pruning. Most of the terms have been eliminated and some messages will not have any words in the vocabulary at all.

Figure 6 shows another view of the observed error rate of the classifiers. The "bi-grams" vocabulary is significantly larger than the "terms" vocabulary and

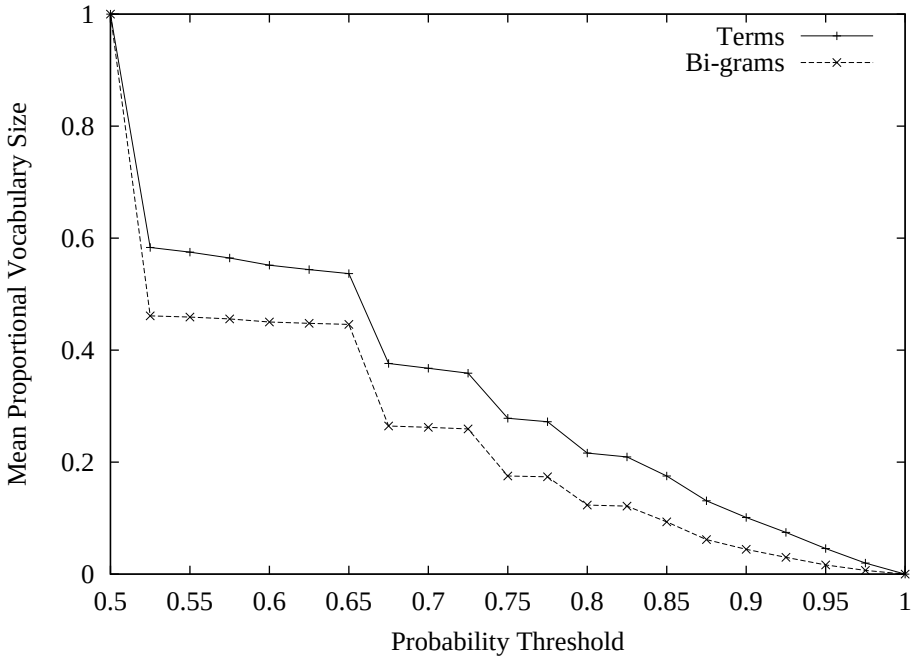

Fig. 4. The mean proportional vocabulary given a probability threshold π.

the number of computations required to classify an e-mail is much larger given a certain probability threshold. While the "bi-grams" vocabulary produces lower error rates given a certain probability threshold, this second plot reveals that as the vocabulary gets smaller the "terms" vocabulary produces better error rates.

The classifiers tended to make similar types of mistakes. The most frequently misclassified messages were invoices for online purchases and automated business communications. Very short messages with ten or fewer distinct terms were often misclassified. With higher probability thresholds, all of the words in the short messages were pruned from the vocabulary.

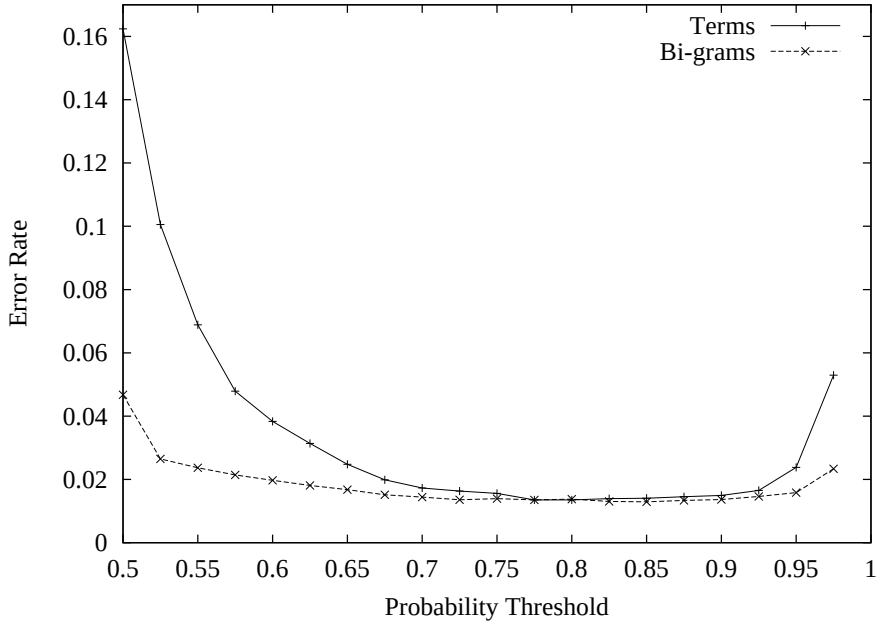

Fig. 5. The error rate of the classifiers given a probability threshold π.

Figure 7 is a scatter plot of the precision and recall of the classifiers on both corpora. The points towards the bottom right of the plot are obtained from the classifiers using the lower probability thresholds and the points towards the top left are from the higher probability thresholds. The large clusters of points correspond to the plateaus on the plot in figure 5 and show the range of probability thresholds that produce the best results.

The F-measure [7] is the harmonic mean of the precision and recall, combining the two into a single efficiency measure, $F(\text{precision}, \text{recall}) = 2\frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$. Figures 8 and 9 show the F-measure given the probability threshold or vocabulary size respectively for each corpus. As with the error rate, the classifiers using the "bi-grams" vocabulary out-performed the classifiers using the "terms"

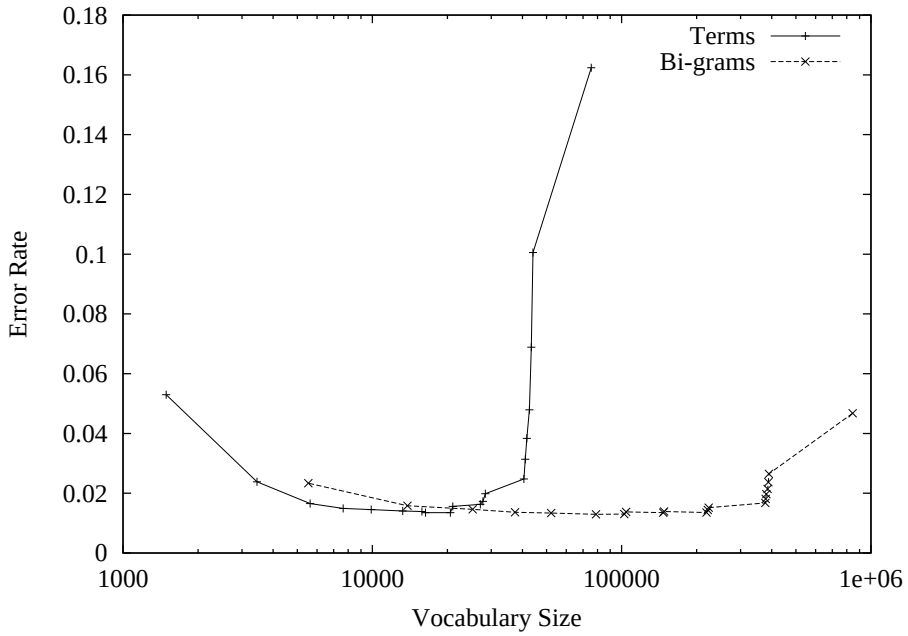

Fig. 6. The error rate of the classifiers given the vocabulary size.

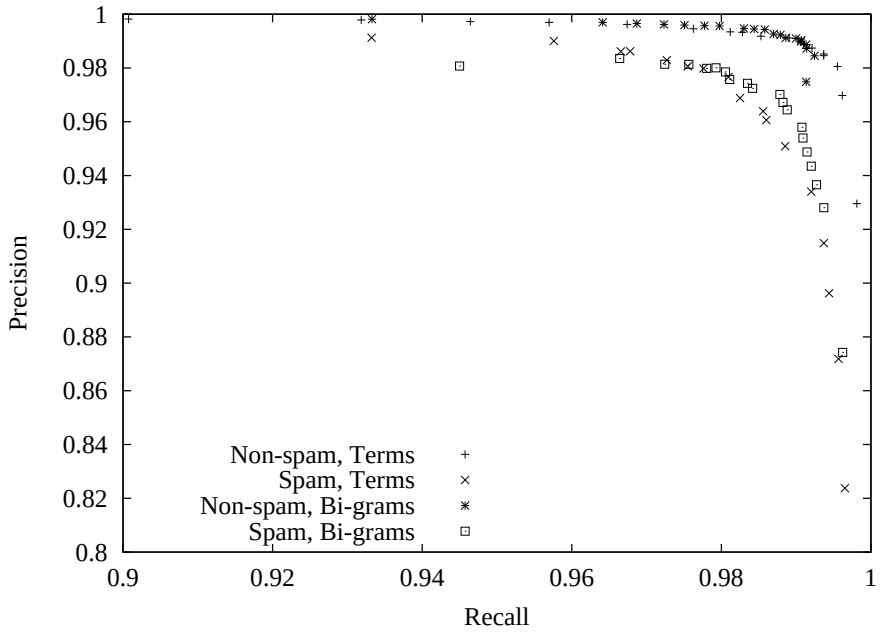

Fig. 7. Scatter plot of the precision and recall of the classifiers.

vocabulary when the probability threshold is taken into account. For pure computational efficiency, the classifiers using the "terms" vocabulary produce better error rates for the amount of computational effort.

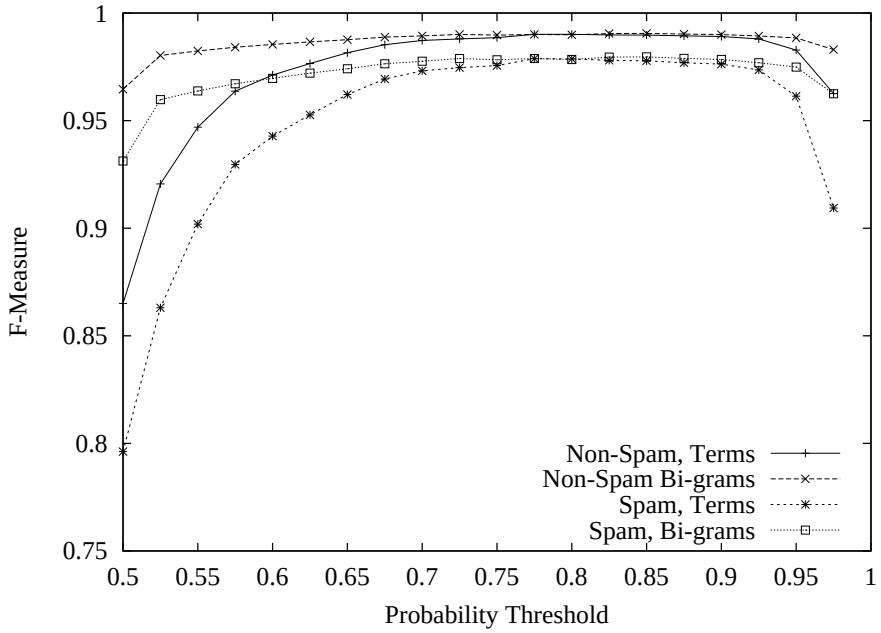

Fig. 8. The F-measure for the classifiers given a probability threshold π.

Figure 10 represents the data from figure 2 "folded" over the line $\pi = 0.5$ and divided by $entropy(\pi)$. When compared with the results shown in figures 5 and 8, spikes are observed over the intervals $\pi \in \mathbb{R}[0.5 \ldots 0.7]$ and $\pi \in \mathbb{R}[0.9 \ldots 1.0]$ and a plateau is observed in the interval $\pi \in \mathbb{R}[0.7 \ldots 0.9]$.

4 Conclusions

Empirical evidence suggests that the hypotheses presented in sections 2.2 and 2.3 are true. Constructing a classifier for each example using only the terms that occur in that example does not seem to adversely affect the error rate of the classification. Eliminating terms with high entropy both reduces overfitting of the training data and significantly reduces the computational cost of a classification.

If a high F-measure is important at the cost of computational efficiency, using bi-grams instead of individual terms produces much better results. It is possible that using longer n-grams or finding common subsequences of words, better results may be obtained provided that the training data is not over-fit.

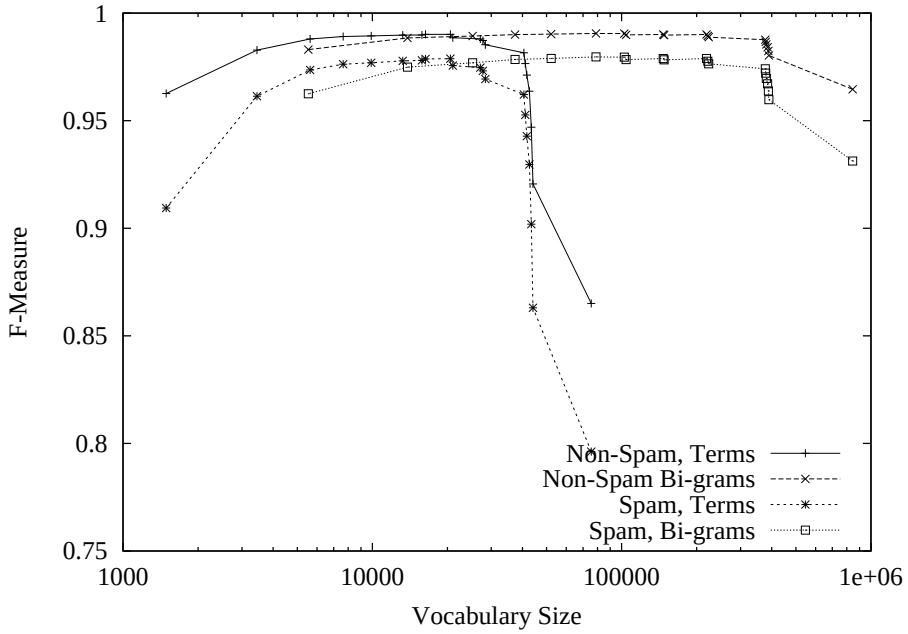

Fig. 9. The F-measure for the classifiers given the vocabulary size.

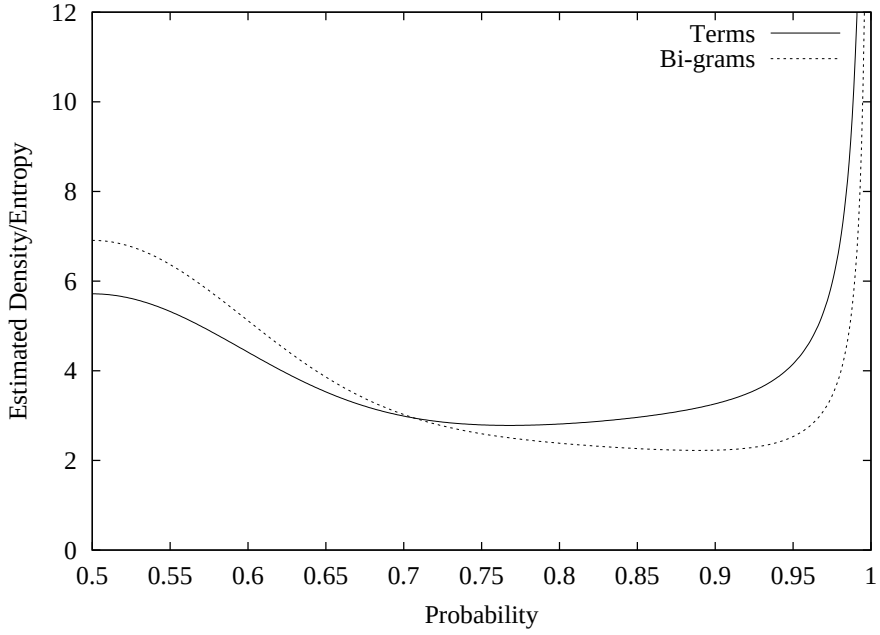

Fig. 10. The ratio of estimated density to entropy plotted against probability.

If computational and storage efficiency is important without a significant reduction of error rate, the small vocabularies obtained by eliminating single terms provides satisfactory results.

From the results in this experiment, it can be suggested that the optimal probability threshold for vocabulary reduction over other data sets can be obtained by examining the ratio of estimated density to entropy.

5 Acknowledgements

I would like to thank Dr. Vlado Keselj, Dr. Mike Shepherd and Bin Tang for their support and collaboration during this project. I would also like to thank everyone at Dalhousie University who provided me with e-mails to build my corpora. In order of volume of contribution, they are: Billy Biggs, Mike Stern, Dr. Jamie Blustein, Carrie Gates, Peter Cordes, Andy VanSlyke and Melanie Schwarz.

References

1. I. Androutsopoulos, J. Koutsias, K. Chandrinos, G. Paliouras, and C. Spyropoulos. An evaluation of naive bayesian anti-spam filtering, 2000.
2. I. Androutsopoulos, G. Paliouras, V. Karkaletsis, G. Sakkis, C. Spyropoulos, and P. Stamatopoulos. Learning to filter spam e-mail: A comparison of a naive bayesian and a memory-based approach, 2000.
3. X. Carreras and L. Márquez. Boosting trees for anti-spam email filtering. In *Proceedings of RANLP-01, 4th International Conference on Recent Advances in Natural Language Processing*, Tzigov Chark, BG, 2001.
4. E. Crawford, J. Kay, and E. McCreath. Automatic induction of rules for e-mail classification, 2001.
5. G. Manco, E. Masciari, M. Ruffolo, and A. Tagarelli. Towards an adaptive mail classifier, 2002.
6. M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz. A bayesian approach to filtering junk E-mail. In *Learning for Text Categorization: Papers from the 1998 Workshop*, Madison, Wisconsin, 1998. AAAI Technical Report WS-98-05.
7. C. J. Van Rijsbergen. *Information Retrieval*. Butterworths, London, 2nd edition, 1979.