Phrases and Feature Selection in E-Mail Classification

Elisabeth Crawford

Computer Science Department
Carnegie Mellon University
PA 15213 USA

ehc@cs.cmu.edu

Irena Koprinska and Jon Patrick

School of Information Technologies
University of Sydney
NSW 2006 Australia

{*irena,jonpat*}*@it.usyd.edu.au*

Abstract *In this paper we study the effectiveness of using a phrase-based representation in e-mail classification, and the affect this approach has on a number of machine learning algorithms. We also evaluate various feature selection methods and reduction levels for the bag-of-words representation on several learning algorithms and corpora. The results show that the phrase-based representation and feature selection methods can be used to increase the performance of e-mail classifiers.*

Keywords E-Mail Classification, Text Categorization, Feature Selection

1 Introduction

E-mail management is a significant and growing problem for individuals and organisations. E-mail users commonly try to manage the large amount of e-mail they receive by sorting it into folders. Most e-mail managers allow the user to hand construct rules to automatically assign e-mails to folders. However, this feature is rarely used [4]. A system that can automatically *learn* how to classify e-mail messages is desirable.

Several systems for automatic e-mail classification have been developed. Cohen [1] used RIPPER to induce keyword-spotting rules. Bayesian approaches have been used e.g. [11] as well as Nearest-neighbour techniques [13]. Previous studies have been limited because they have failed to explore the use of feature selection and phrase representations. In this paper we show how these techniques can be used to increase classification accuracy.

2 Phrase Representation

Bag of Words (BOW) is the approach most commonly used to represent documents in Text Categorization (TC). In BOW, each document is represented by a vector that contains an importance weighting for every word in the corpus. Phrase-based approaches use whole groups of words as features in the bag and as such preserve more of the document's semantics.

Proceedings of the 9th Australasian Document Computing Symposium, Melbourne, Australia, December 13, 2004.

In order to use a phrase-based document representation we need a method for choosing the phrases. Syntactic and semantically derived phrases have been used without any significant improvement over the BOW approach e.g. [8, 5, 12]. Statistically selected phrases have proved more successful. Mladenic and Grobelnik [9] found that for WWW documents, n-grams (of length 3-4) selected using odds ratio improved the performance of Naive Bayes. Similar results were reported by Furnkranz [6] on Reuters data for the RIPPER classifier using frequency based selection of 2 and 3-grams.

We construct the phrase-based representation used in this paper by first stemming and removing stopwords. We then generate all 1 and 2-grams and place them in a bag of features weighted by their normalized tf-idf. Phrases are then selected statistically using one of the methods described in the next section.

3 Feature Selection

Feature selection is often an essential step in TC as text collections can have more than 100,000 unique terms (words or phrases). Removing less informative and noisy terms reduces the computational cost and often improves classifier generalization. Feature selection works by ranking all the terms and then selecting some percentage. A variety of ranking criteria have been used in TC with varying degrees of success. Some of the more successful approaches are variants of χ^2 [17], information gain [17] and odds ratio [10].

We have chosen to experiment with χ^2, Information Gain (IG) and Document Frequency (DF). We included DF becuase it is computationally efficient and Yang and Pedersen [17] showed that except for aggressive levels of feature selection (bigger than 90 percent), it performed similarly to the first two.

4 Experimental Setup

We use a corpus consisting of e-mail messages for 4 different users (first 4 users in [2]). Each of the users has a different set of criteria for classifiying their e-mails. For instance, users 1 and 3 categorize e-mail mostly on the basis of topic and sender, while user 2 categorizes their mail based on when it needs to be acted upon. User 4 is different again, classifying according to the actions performed (e.g Delete, ReplyAndKeep) as well as topic and sender. The corpora contain between

430 and 972 email messages classified into between 7 and 39 folders. The e-mails in each corpus were ordered by date with the first two thirds becoming the training set and the final third the test set. Note that the strongly temporal nature of e-mail makes cross validation an unsuitable option. For each corpus 5, 10 and 30 percent of the features in the BOW representation were calculated. These numbers were then used to define the number of features selected both for the BOW and phrase representation, i.e. the *same* number of features was used for both.

We have taken the approach of building for each category a binary classifier. This allows e-mails to be placed in more than one category as necessary. We look at six learning algorithms: Support Vector Machines (SVMs), K-Nearest Neighbour (KNN), Decision Trees (DTs), Perceptron, Widrow-Hoff (WH), and Naive Bayes (NB). The following options were used: KNN with cosine similarity as distance metric and distance weighted voting, for k equal to 1, 5 and 30; SMO algorithm for SVMs with both linear and quadratic kernels, C4.5 for DTs and the standard algorithms for WH, NB and Perceptron. We implemented WH and used the WEKA implementations [14] of the other algorithms. Micro-averaged F1 is used to measure the average perfromance of the ML algorithms over multiple categories.

5 Results and Discussion

5.1 Effect of ML aglorithm

Figures 1 to 4 show, for each user, the performance of both the BOW and 2-gram phrase approach for the different learners. For each learner we chose the best result achieved on any of the feature selection settings.

The results demonstrate a strong difference in the difficulty of automatically categorizing different users' e-mails. Consistent with [2] our results show that the coarser grained e-mail sets of users 1 and 3, where many e-mails were classified according to topic, were easier to classify than the finer grained user 2 and 4 corpora. These corpora were harder to classify due to there being less training data per category and the action based classifaction policies of the users.

For the phrase-based representation, SVMs produced the highest classification performance for users 1, 2 and 3, and DTs for user 4; WH and KNN also performed very well. For the BOW representation, KNN produced the highest results for users 2 and 4, WH for user 1 and 3. Overall, SVMs performed the best and NB the worst, which is consistent with Yang and Liu's [16] comparison on Reuters data using BOW. The NB classifier had very high recall, but low preicision. This problem could perhaps be lessened by careful thresholding. WH also performed very well (phrases: 2nd for user 3, 3rd for user 1; BOW: 1st for users 1 and 3). We note that while KNN was the top learner on Reuters and Ohsumed [15], it was less successful on e-mail data. Compared to these corpora, e-mail contains a great deal of noise because of different writing styles and most probably inconsistencies in classifications.

5.2 BOW vs Phrases

Figure 1 shows that on the user 1 corpus, the phrase based representation led to performance increases for all learners except WH and kNN5 with highest improvement for SVM1 and SVM2. On the user 2 data (Figure 2), the phrases worked better on five learners, almost equal on two and worse for NB and KNN1. The biggest improvement was achieved for SVM2 and DTs. Similar results were obtained for user 4 (Figure 4). The classification performance of BOW and phrases is closest on the user 3 corpora: phrases outperformed BOW on four learners, achieved similar performance on another four and were only slightly worse on NB.

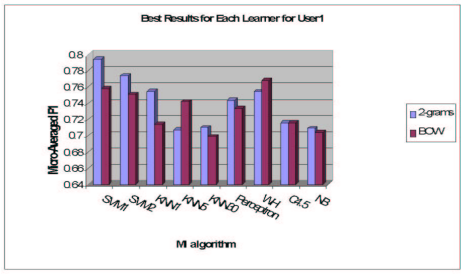

Figure 1: Best results for user 1

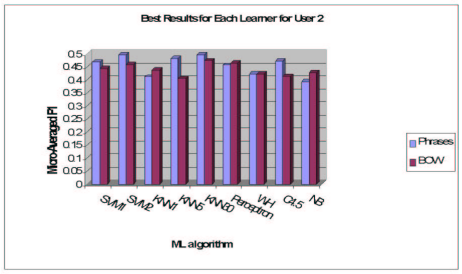

Figure 2: Best results for user 2

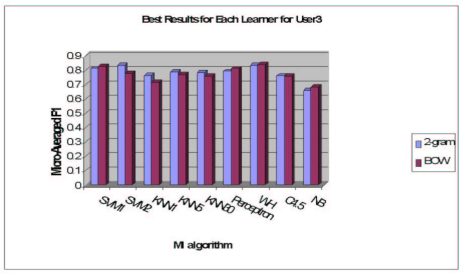

Figure 3: Best results for user 3

The results clearly show that the proposed phrase based representation is useful for e-mail classification — the representation has improved the performance of a variety of ML algorithms over varied corpora. In the next two sections we examine how feature selection can be used to improve e-mail classification using a BOW representation.

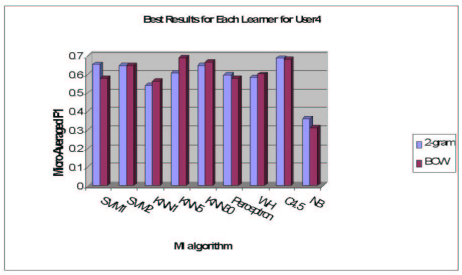

Figure 4: Best results for user 4

5.3 Feature Selection and ML algorithms

The effect of the feature selection algorithms and reduction level has been investigated for BOW on the standard TC corpora. For example, Joachims [7] found feature selection improved the perfromance of KNN and C4.5, but not linear and quadratic SVMs or NB on Reuters and Ohsumed. Yang and Pedersen [17] found that DF, IG and χ^2 have similar characteristics (prefer common terms) and similar performance on Reuters data using KNN and LLSF classifiers. DF was found to perform comparably with IG and χ^2 with up to 90% term removal. To see how these results translate to e-mail data, we first look at the effect of the feature selection algorithm and then the effect of feature selection level.

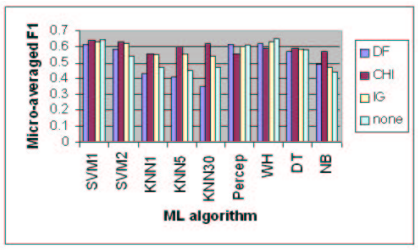

Figure 5: Comparison of feature selection methods across learners for BOW

To investigate the effect of the feature selection methods on each learner, we averaged the performance results across the four corpora and the feature selection levels. Figure 5 shows feature selection was useful for all learners except SVM1 and WH. These results support the theoretical and empirical evidence that SVMs learn independently of the dimensionality of the feature space [7]. Our results show that SVM1 is a very stable classifier regardless of the different feature selection mechanisms and levels. However, for user 4, χ^2 level 5 significantly increases the performance of SVM1 over the full data set. Thus, the corpora has an effect on the usefulness of the feature selection even for stable classifiers.

The results for the quadratic kernel SVM (SVM2), however, are not consistent with [7]. As can be seen, SVM2 benefits from feature selection and χ^2 was found to be the best selector. χ^2 was also found to be the best feature selection method for KNN (k=1, 5 and 30), DT (C4.5) and NB while DF was the best for Perceptron. Unlike Yang and Pedersen, we see that for KNN DF is not comparable to IG and χ^2 even for relatively low feature reduction; in fact it was even worse than using the full feature set. The biggest improvement due to the use of feature selector is for KNN30 and NB, where χ^2 improves the performance by more than 11%. The combination of ML algorithm and feature selection mechanism is clearly important. Different feature selection mechanisms produce different changes in performance across a variety of ML algorithms.

Figure 6 shows the effect of the feature selection level on each learner. We have averaged the effects of the different feature selection algorithms and corpora over the three levels. Overall, level 5 (i.e. 95% term reduction) was the best for KNN, DT and NB. The quadratic kernel SVM (SVM2) performed best with level 30 feature selection while Perceptron, WH and SVM1 performed best without feature selection. NB was the most stable in terms of feature selector and level it always prefers DF and high reduction (level 5, 10).

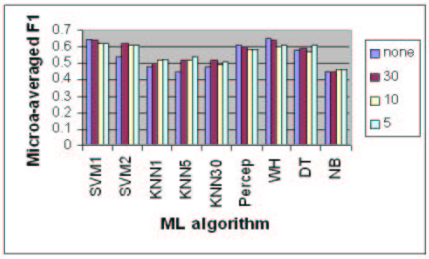

Figure 6: Comparison of feature selection levels across learners for BOW

For each corpora we averaged, the performance of DF, χ^2 and IG averaged over all ML algorithms (figures not shown). We found DF performs slightly worse than IG and χ^2 and there is no evidence that its performance drops at 90% term reduction as Yang and Pedersen found on Reuters. The KNN results (i.e. without the averaging across ML algorithms) also do not show this. On the other hand, like Yang and Pedersen we did find that IG and χ^2 are highly correlated suggesting that this patter might be general as opposed to corpus dependent.

5.4 Corpora and Feature Selection

To study the effect of the *feature selection methods* on each e-mail corpora, we have averaged performance results for all learners and feature selection levels (Figure 7). Users 1 and 3 (who classify based on topic and subject) benefit most from feature selection. Overall, IG is the best feature selector for all corpora and χ^2 obtained almost the same performance as IG on two corpora. For one of the difficult corpora (user 2), the performance of all feature selectors is comparable and not significantly better than not using feature selection. The reason for this is that user 2 classifies e-mails mainly based on the action performed on them, thus, since the correct class cannot always be predicted based on the text, feature selection less useful. Feature selection was

beneficial for the other difficult corpora - user 4. This is because user 4 classifies e-mails according to actions to a lesser extend than user 2 and the vocabulary for this corpus is much larger than any of the others. A large vocabulary implies there may be more noise in the training data makeing feature selection more useful.

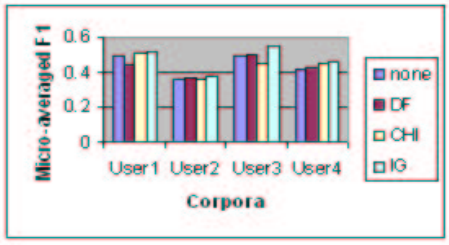

Figure 7: Comparison of feature selection methods across corpora for BOW

Figure 8 shows the effect of the *feature selection levels* on each corpus. We have averaged performance results for all learners and feature selection methods. Aggressive feature selection was found to work best for user 1, 3 and 4. For the more difficult corpora users 2 and 4, the performance of all reduction levels is similar due to the reasons discussed above, overall the moderate level 30 works best.

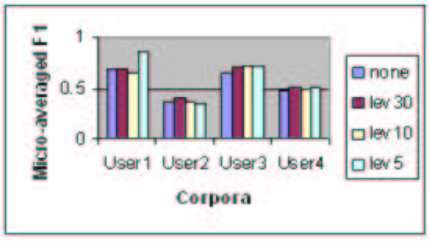

Figure 8: Comparison of feature selection levels across corpora for BOW

6 Conclusions and Future Directions

In summary:

- There can be a strong difference in the difficulty of automatically categorizing different users e-mail;
- The *combination* of ML algorithm and feature selection mechanism effects performance. Thus, it is important for feature selection mechanism in TC to be evaluated across a variety of ML algorithms, which often has not been the case in previous work;
- The phrase-based representation improves performance;
- Overall, SVMs (both linear and quadratic kernel), WH and KNN were found to be the best classifiers for both phrases and BOW;
- Feature selection is useful and overall improved the performance of all ML algorithms except for linear kernel SVM and WH;
- χ^2 was found to be the best feature selector for four out of nine learners;
- Aggressive feature selection (90-95% reduction) worked quite well on all corpora.

When analysing our experiments we noted how performance differed according to the *combination* of ML algorithm, feature selection mechanism and level, text representation and corpora. In the future we would like to explore an approach similar to that described in [3] to choosing a combination of feature selection mechanism and level, text representation and ML algorithm that is best suited to the particular user's e-mail.

References

[1] W. Cohen. Learning rules that classify e-mail. In *AAAI Spring Symposium on Machine Learning in Information Access, pp. 18-25*, 1996.

[2] E. Crawford, J. Kay and E. McCreath. Iems - the intelligent email sorter. In *19th Int. Conf. on Machine Learning*, 2002.

[3] E. Crawford, I. Koprinska and J. Patrick. A multi-learner approach to e-mail classification. In *ADCS*, 2002.

[4] N. Ducheneaut and V. Bellotti. E-mail as habitat: an exploration of embedded personal information management. *Interactions v.8, n.5, pp.30-38*, 2001.

[5] S. Dumais, J. Platt, D. Heckerman and M. Sahami. Inductive learning algorithms and representations for text categorization. In *Proc. of CIKM-98*, 1998.

[6] J. Furnkranz. A study using n-gram features for text categorization, 1998.

[7] T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In *10th ECML*, 1998.

[8] D. Lewis. An evaluation of phrasal and clustered representations on a text categorization task. In *Proc. of SIGIR-92*, pages 37–50, 1992.

[9] D. Mladenic and M. Grobelnik. Word sequences as features in text learning. In *Proc. of the 17th Electrotechnical and Computer Science Conference*, 1998.

[10] Dunja Mladenic. Feature subset selection in text-learning. In *ECML*, pages 95–100, 1998.

[11] P. Pantel and D. Lin. Spamcop: A spam classification & organization program. In *AAAI-98 Workshop on Learning for Text Categorization*, 1998.

[12] S. Scott and S. Matwin. Feature engineering for text classification. In *Proc. 16th ICML*, 1999.

[13] R. Segal and M. Kephart. Mailcat: An intelligent assistant for organizing e-mail. In *3d Int. Conf. on Autonomous Agents, pp.276-282*, 1999.

[14] I. Witten and E. Frank. *Data Mining - Practical Machine Learning Tools and Techniques with Java Implm.* Morgan Kaufmann, 2000.

[15] Y. Yang. An evaluation of statistical approaches to text categorization. *Information Retrieval v.1, n.1/2, pp.69-90*, 1999.

[16] Y. Yang and X. Liu. A re-examination of text categorization methods. In *22nd ACM International Conference on Research and Development in Information Retrieval*, pages 42–49, 1999.

[17] Y. Yang and Jan O. Pedersen. A comparative study on feature selection in text categorization. In *ICML*, pages 412–420, 1997.