The Enron Corpus: A New Dataset for Email Classification Research

Bryan Klimt and Yiming Yang

Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15213-8213, USA
{bklimt,yiming}@cs.cmu.edu

Abstract. Automated classification of email messages into user-specific folders and information extraction from chronologically ordered email streams have become interesting areas in text learning research. However, the lack of large benchmark collections has been an obstacle for studying the problems and evaluating the solutions. In this paper, we introduce the Enron corpus as a new test bed. We analyze its suitability with respect to email folder prediction, and provide the baseline results of a state-of-the-art classifier (Support Vector Machines) under various conditions, including the cases of using individual sections (From, To, Subject and body) alone as the input to the classifier, and using all the sections in combination with regression weights.

1 Previous Work on E-mail Classification

Email classification can be applied to several different applications, including filtering messages based on priority, assigning messages to user-created folders, or identifying SPAM. We will focus on the problem of assigning messages to a user's folders based on that user's foldering strategy. One major consideration in the classification is that of how to represent the messages. Specifically, one must decide which features to use, and how to apply those features to the classification. Manco, et al. [8] defined three types of features to consider in email: unstructured text, categorical text, and numeric data. Relationship data is another type of information that could be useful for classification.

Unstructured text in email consists of fields like the subject and body, which allow for natural language text of any kind. Generally, these fields have been used in classification using a bag-of-words approach, the same as with other kinds of text classification [2,8,10]. Stemming and stop word removal are often used, as they are useful in general text classification, although their usefulness in email in particular has not yet been studied thoroughly. It has been found that some of these fields are more important that others in classifying email [4,8].

Categorical text includes fields such as "to" and "from" [8]. These differ from unstructured text fields in that the type of data which can be used in them is

very well defined. However, these fields have typically been treated the same as the unstructured text fields, with the components added to the bag of words [2,8]. These fields have been found to be very useful in automatic email classification, although not as useful as the unstructured data [4,8].

Numeric Data in email includes such features as the message size, number of recipients [8], and counts of particular characters [4]. So far, every test has found that these features can contribute little towards email classification.

Studies on the use of relationship data in email foldering have not yet been published to our knowledge. Relationship data consists of the connections between an email message and other types of objects, such as users, folders, or other emails. One such relationship between emails is that of *thread membership* [7,9]. A thread is a set of email messages sent among a set of users discussing a particular topic. We believe that use of thread information could improve the results of email classification. The reasoning behind this is that often, in a message that is part of a long discussion, not everything from the earlier parts of the discussion is repeated. This missing information could provide important clues about how to classify a message.

The difference between thread information and the other kinds of email data is that thread data is not provided explicitly. It must be deduced from other data fields. There has not been much work in how to detect thread information automatically. One study by Murakoshi, et al. looked at finding thread structure using linguistic analysis, which is a difficult natural language problem, and is difficult to evaluate [9]. Another approach, introduced by Lewis and Knowles, was to use the hierarchy of message replies as an approximation of the thread structure, although it is obvious that it will not be perfect [7]. They examined how to find this structure automatically. Their best results were from using quoted text in a message as a query and ranking the other emails in the corpus by their cosine similarity to the query. They reported about .71 accuracy in determining the parent message of messages known to be in threads. The retrieval algorithm in their study did not take advantage of word order. It also did not address how to determine whether or not a particular email was a member of a thread to begin with.

It is not yet known which classification algorithms will work best for automatic folder classification. In fact, the literature suggests that the variation in performance between different users varies much more than the variation between different classification algorithms [1]. Kiritchenko and Matwin found that SVM worked better than Naïve Bayes, with 75-87% accuracy, depending on user [6]. Brutlag and Meek also found SVM to perform best, but only for dense folders, i.e. folders with at least twenty messages [1]. They found that for sparse folders, on the other hand, TF-IDF similarity worked best for most users, with 67-95% accuracy. TF-IDF has been tested by others [3,11], with accuracies in a

similar range, as has Naïve Bayes [4,10]. Several studies have tried approaches based on automatic learning of rules for classification, with accuracies similar to those of the other methods [2,3,5]. None of these studies, however, has shown that any particular method outperforms the others for a large variety of data sets.

There has not yet emerged a common data set for use in evaluating email folder classification, so it is hard to compare the results of different researchers. Most studies so far have used personal collections of the people working on the experiments [1,2,3,4,6]. These sets have been incredibly small, on the order of one to five users. Since email organization strategies vary from user to user, it will be necessary to perform studies with larger data sets before conclusions can be made about which algorithms work best for email classification.

2 Enron Dataset

A large set of email messages, the Enron corpus, was made public during the legal investigation concerning the Enron corporation. The raw corpus is currently available on the web at `http://www-2.cs.cmu.edu/~enron/`. The current version contains 619,446 messages belonging to 158 users. We cleaned the corpus for use in these experiments by removing certain folders from each user, such as "discussion_threads" and "notes_inbox". These folders were present for most users, and did not appear to be used directly by the users, but rather were computer generated. Many, such as "all_documents", also contained large numbers of duplicate email messages, which were already present in the users' other folders. Since our goal in this paper is to explore how to classify messages as organized by a human, these folders would have likely been misleading.

In our cleaned Enron corpus, there are a total of 200,399 messages belonging to 158 users with an average of 757 messages per user. This is approximately one third the size of the original corpus. Figure 1 shows the distribution of emails per user. The users in the corpus are sorted by ascending number of messages along the x-axis. The number of messages is represented in log scale on the y-axis. The horizontal line represents the average number of messages per user (757).

As can be seen from the graph, the messages are distributed basically exponentially, with a small number of users having a large number of messages. However, there are users distributed along the entire graph from one message to 100,000 messages, which shows that the Enron dataset provides data for users with all amounts of email. More important in folder classification, though, is the number of folders each user has. The distribution of folders for each user is shown in figure 2. Each point is a user, and shows the number of folders and messages the user has.

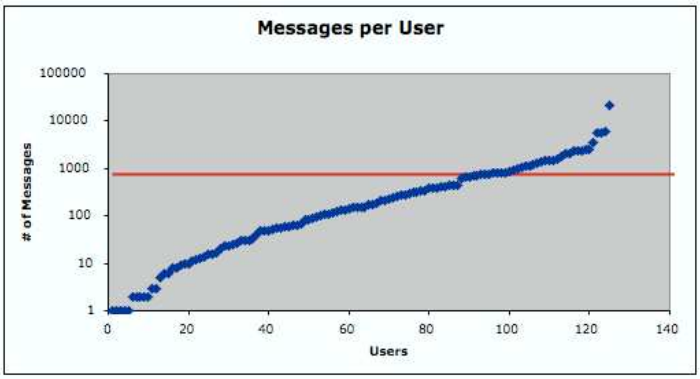

Fig. 1.

Figure 2 illustrates that the Enron dataset is consistent with many of the assumptions made about email folder classification. Most importantly, it shows that most users do use folders to organize their email. If users did not categorize their email into folders, then automatic classification would not be useful for them. Secondly, it shows that the number of messages a user has does not necessarily provide a lower bound for the number of folders that person uses. Some users with many messages have a relatively small number of folders. The number of messages does, however, obviously provide an upper bound for the number of folders the user has. Unsurprisingly, no user has more folders than messages. More interesting is the fact that the upper bound for the number of folders of a user appears to be a log of the number of messages of that user. In other words, users with more total messages tend to have more messages in each individual folder.

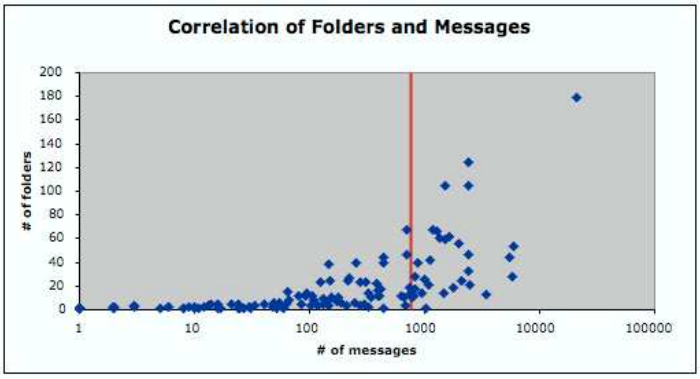

Fig. 2.

3 SVM Classification

The dataset was evaluated using a state-of-the-art text classifier on various representations of each email. SVM was used first to classify the folder of each email based solely on a particular field of data from the email. The fields used were "From", "Subject", "Body", and "To, CC". The date field was not used, as it is not text information, and the problem of how to apply date information to email classification has not been fully explored. Next, SVM was used on each email treated as a single bag-of-words. This approach is labeled "All" in the analysis below. For this representation, the fields used in the previous experiments were concatenated and used in the classification. Thus, if the same term appears in both the subject and body of a message, it is considered to be multiple occurrences of the same feature. For the final approach, labeled "linear combination" below, the SVM scores from the "From", "Subject", "Body", and "To, CC" classifiers were combined linearly. The weights for each section were learned for each folder of a particular user, using ridge regression on the training data.

To create training and testing sets, the data for each user was sorted chronologically, and then split in half. The earlier half of the messages was used for training, while the later half was used for testing. Standard text parsing routines were applied to each of the fields in the email to produce the list of terms. Stemming was also performed on the body of the message. The terms were then given weights using the standard "ltc" formula, and given to SVM, using the one-vs-rest method for multi-class classification.

For binary decisions, optimal thresholds were found for each folder (category). In many previous experiments, binary decisions were based on choosing only the highest ranked folder for each message, making precision the only relevant evaluation metric. However, it has been suggested [11] that it can be beneficial to the user to present multiple possible assignments for each email. Therefore, we obtained thresholds using score-based local optimization, known as SCut [12], and evaluated using F1 scores, which measure both precision and recall. The folder hierarchy was flattened for these experiments. In other words, the "correct" folder for a given message was considered to be the lowest level folder containing the email. The reason for this is so that a correct classification is only given credit once. Otherwise, scores would be inflated significantly by large root folders, such as "Inbox", which contain many other folders. The results for the evaluation are given in Figure 3.

Each bar in the figure represents the average score of all of the users for a particular test. The most useful feature on average is the body of the email, although it was not significantly better than the From field. The least useful feature is clearly the To and CC fields of the email. This is understandable, as most of the messages going to a user have the same address in the To field, so that address is not a very discriminative feature. As for the methods that used multiple fields of the emails' data, using ridge regression to combine the indi-

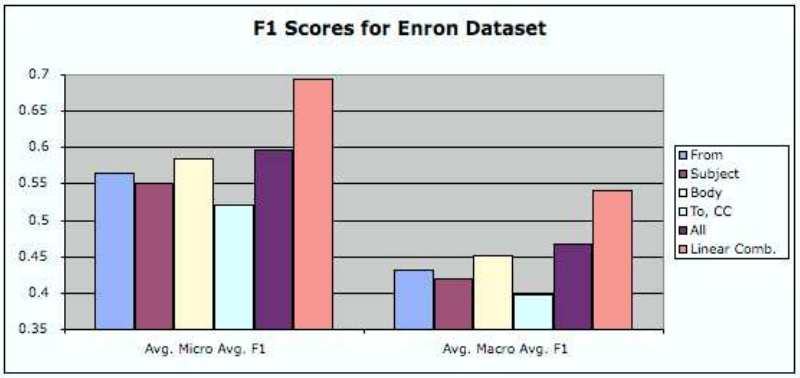

Fig. 3.

vidual scores linearly proved significantly more effective than treating the fields as a single bag of words. The fact that one feature does not dominate the other features here shows that the users in the Enron corpus do not use a single field in determining email organization, but rather use of combination of the data in their organizational scheme.

Figure 4 shows the correlation between the number of messages a user has and the linearly combined F1 score for that user. The number of messages a user has is clearly not strongly correlated with the performance of the text classifier on his or her email. This result is reasonable, though. If a user has many messages, but they are all in the same folder, classification is trivial. If however, they are spread out, the performance of the classifier depends on the foldering strategy of the user. In other words, the number of folders a user has should be a much bigger predictor of the ease of automatic classification for a user.

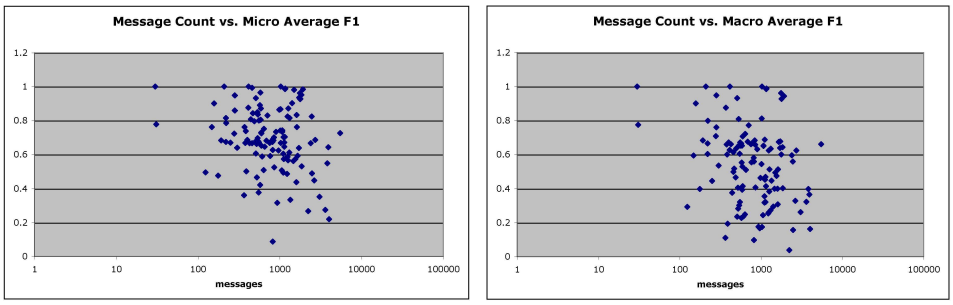

Fig. 4.

Figure 5 shows that this correlation does exist. Obviously, the three users with only one folder each had a perfect score. Users with more folders tended to have lower scores, which can be seen as evidence of their more complex foldering strategies. However, as we saw previously, the users with more folders tended to have more items in each folder. Since SVM generally performs better on classes with more training examples [1], there must be important features of the email which are not being modeled in these experiments.

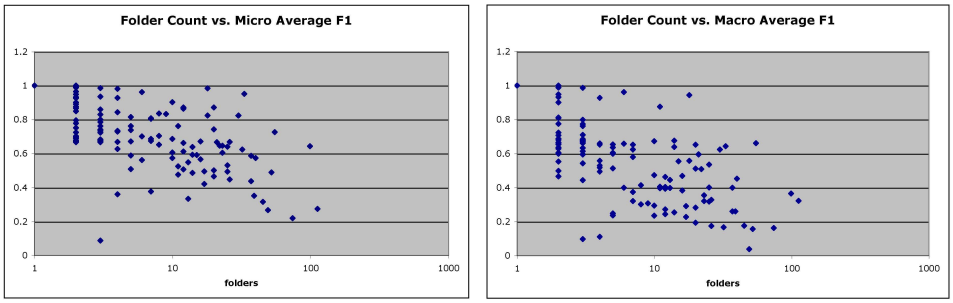

Fig. 5.

4 CMU Dataset

To determine if the classification results for the Enron dataset were reasonable, a second data set was used for the experiments. The CMU dataset was collected over several months from several students and a faculty member at the Language Technology Institute of CMU.

	number of messages
user 1	1338
user 2	1438
user 3	403
user 4	703
user 5	4381

The results with this test set are similar to the results with the Enron dataset, with a micro average F1 score near .7 and a macro average score near .55. Micro averages are generally higher than their corresponding macro averages, which may reflect the previous observation that SVM tends to work much better on folders with more messages [1], as macro-averages are dominated by small categories. We see that From and Body are the best performing representations for

email classification, while To and CC are less useful. The linear combination of scores again outperformed the bag of words representation on average, although not by as large a margin as with the Enron dataset. This shows that, in total, the users in the Enron dataset most likely use more diverse foldering strategies than the users in the CMU dataset, as combining evidence from different sections better improves results relative to the individual sections used seperately. The overall similarity of these results, however, reinforces the idea that the Enron dataset is a useful dataset representative of common users.

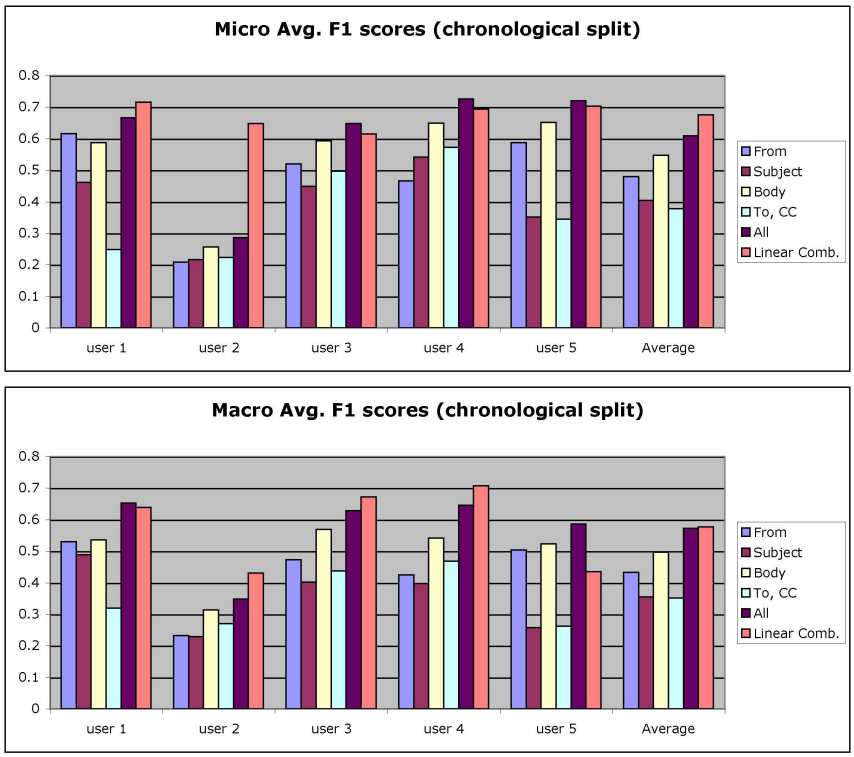

Fig. 6.

5 Threads

We have also briefly analyzed the nature of email "threads" in this corpus. For this analysis, membership in a thread was determined by two factors. Emails were considered to be in the same thread if they contained the same words in their subjects and they were among the same users (addresses). Messages with

empty subjects were not considered to be a thread. No evaluations were conducted to test the quality of the thread detection algorithm. The main reason for this is that threads are rather subjective, and user judgments are difficult to obtain. Lewis [7] had used the "in-reply-to" headers in email messages as the truth about thread membership to test his thread detection algorithm. Unfortunately, it appears that some current email clients do not use this header, as the Enron corpus has very few messages with it. Also, the algorithm could not be evaluated against the results from the previous paper, as the corpus used in that paper is no longer available. The algorithm used in that paper also does not provide a mechanism for determining whether a message is in a thread at all, as only messages known to be in threads were used in the experiments.

Investigation of the use of threads to improve email classification would provide an indirect evaluation of our detection algorithm, and would be an interesting topic for future research. We will now, however, only provide some statistics of the threads present in the Enron dataset as determined by our algorithm. Hopefully, our analysis will be useful for future research on thread detection and use.

Out of the total 200,399 messages in the Enron corpus, we detected 101,786 threads. 71,695 of these threads were *trivial* threads, consisting of only one message. These threads would clearly not be useful for classification, so there are 30,091 remaining useful threads in the corpus, consisting of 123,501 of the 200,399 total messages. In other words, a full 61.63% of messages in the corpus is in a thread. This makes the average thread size 4.10 messages. The median useful thread size, however, is only 2.00. So, there are a few large threads in the corpus, and many small threads. In fact, the distribution of thread sizes is as follows:

thread size:	2	3	4	5	6	7	8	9	10	(10-20]	(20-30]	(30-40]	(40-50]	51+
# of threads	16736	4782	3049	1282	879	903	378	214	178	1260	209	79	54	88

The larger threads should be more useful, since they provide more information about the relationships of a message, but they are less common. However, more important than the size of the thread is the information the thread can provide. The average number of folders containing the messages of a thread is 1.37. This means, given an average thread, the messages in that thread are distributed among only 1.37 folders. This information could be very useful in email classification.

The major drawback of this thread information is that it may be redundant when used with the other kinds of evidence discussed in this paper. Since subject words are used to detect threads, a thread based classifier may not provide any information not already available, if it is used as just another feature. One example of a redundancy problem is with the largest thread in the Enron corpus, messages with the subject "Demand Ken Lay Donate Proceeds from Enron Stock Sales" belonging to user "lay-k". There are 1124 messages in this thread

and they are all in the same folder (Deleted Items)! This would be incredible evidence for classification by itself. However, all of the messages in the thread are virtually identical; they appear to be SPAM. Since the messages are identical, there is already incredibly strong evidence from other features, without even detecting the thread.

Thread information may be useful in conjunction with other fields, but one must use a method that can infer more from thread membership than the redundant information provided by other fields. The Enron corpus has a large number of threads and would be useful as a test set for these methods.

6 Acknowledgements

The authors would like to thank Fan Li for providing original experiment designs and baseline results for the CMU dataset using SVM with ridge regression. We would also like to thank William Cohen for promoting the Enron dataset, by providing it to us, and by giving it a home on the web.

7 Conclusions and Future Work

There are many more ways to model email than the methods attempted in this paper. More research needs to be done into using the relationships between emails to reinforce knowledge about a particular message. One of these relationships is thread membership. While a small amount of research has been done into how to detect threads, no one has studied how to use the threads for the task of email classification. Time information was also left out of these experiments, while it seems clear that it could be useful. Time cannot be used in the same way as other fields though, obviously, so work must be done to determine how time affects the foldering strategies of a user.

In order to compare new techniques for email classification, a large standard test dataset, such as the newly available Enron corpus, could be very valuable. It can be used both as a large sample of real life email users and as a standard corpus for comparison of results using different methods. We have provided evaluations of baseline experiments on the dataset. We have also provided a brief analysis of the dataset itself.

References

1. J. D. Brutlag, C. Meek: Challenges of the Email Domain for Text Classification. ICML 2000: 103-110

2. W. W. Cohen: Learning Rules that classify E-mail. In Proc. of the 1996 AAAI Spring Symposium in Information Access, 1996.
3. E. Crawford, J. Kay, and E. McCreath: Automatic Induction of Rules for e-mail Classification. In ADCS2001 Proceedings of the Sixth Australasian Document Computing Symposium, pages 13-20, Coffs Harbour, NSW Australia, 2001.
4. Y. Diao, H. Lu, and D. Wu: A comparative study of classification-based personal e-mail filtering. In Proc. 4th Pacific-Asia Conf. Knowledge Discovery and Data Mining (PAKDD'00), pages 408-419, Kyoto, JP, 2000.
5. E. Hung: Deduction of Procmail Recipes from Classified Emails. CMSC724 Database Management Systems, individual research project report. May, 2001
6. S. Kiritchenko, S. Matwin: Email classification with co-training. In Proc. of the 2001 Conference of the Centre for Advanced Studies on Collaborative Research, page 8, Toronto, Ontario, Canada, 2001
7. D. D. Lewis, K. A. Knowles: Threading Electronic Mail: A Preliminary Study. In Information Processing and Management, 33(2): 209-217, 1997
8. G. Manco, E. Masciari, M. Ruffolo, and A. Tagarelli: Towards an Adaptive Mail Classifier. AIIA 2002, Sep. 2002.
9. H. Murakoshi, A. Shimazu, and K. Ochimizu: Construction of Deliberation Structure in Email Communication In Pacific Association for Computational Linguistics (PACLING'99), pages 16-28, Aug. 1999.
10. J. Rennie: ifile: An Application of Machine Learning to E-Mail Filtering. In Proc. KDD00 Workshop on Text Mining, Boston, 2000.
11. R. B. Segal and J. O. Kephart. MailCat: An Intelligent Assistant for Organizing E-Mail. In Proc. of the 3rd International Conference on Autonomous Agents, 1999.
12. Y. Yang: A Study of Thresholding Strategies for Text Categorization. In Proc. of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 137-145, New Orleans, LA, 2001