Representation of Electronic Mail Filtering Profiles: A User Study

Michael J. Pazzani
Department of Information and Computer Science
University of California, Irvine
Irvine, CA 92697
+1 949 824 5888
pazzani@ics.uci.edu

ABSTRACT

Electronic mail offers the promise of rapid communication of essential information. However, electronic mail is also used to send unwanted messages. A variety of approaches can learn a profile of a user's interests for filtering mail. Here, we report on a usability study that investigates what types of profiles people would be willing to use to filter mail.

Keywords

Mail Filtering; User Studies

1. INTRODUCTION

While electronic mail offers the promise of rapid communication of essential information, it also facilitates transmission of unwanted messages such as advertisements, solicitations, light bulb jokes, chain letters, urban legends, etc. Software that automatically sorts mail into categories (e.g., junk, talk announcements, homework questions) would help automate the process of sorting through mail to prioritize messages or suggest actions (such as deleting junk mail or forwarding urgent messages to a handheld device). Such software maintains a profile of the user's interests. Here, we investigate the representation of user profiles from a usability point of view. The goal of this paper is investigate alternative representations of user profiles and show how these alternatives affect the willingness of users to accept an automated system for filtering mail. In particular, alternative representations of a profile may be equally accurate, yet people may have more confidence in a profile presented in one representation over another.

Many commercially available mail-filtering programs that allow a user to inspect the representation are based on rules that look for patterns in the text. Rule learning programs such as Ripper could easily learn rules for such a representation [Cohen, 1996]. Other approaches to classifying text such as electronic mail include using linear models [e.g., Lewis, Schapire, Callan and Papka, 1996; Dumais, Platt, Heckerman and Sahami, 1998] learned by a perception or support vector machine. Naïve Bayesian classifiers [Duda & Hart, 1973] have also proved effective in some applications [Pazzani & Billsus, 1997; Sahami, Dumais, Heckerman, and Horvitz, 1998].

Some of the earliest text classification methods (e.g., Rocchio, 1971) were based upon finding the centroid of examples of each class. Such methods are often competitive with more recent approaches to text classification. Here, we introduce a new *prototype* representation for profiles and show that it is as accurate as alternative approaches and that users place more confidence in the profiles in this representation than rule-based representation or linear models.

To illustrate the three alternative representations, we collected a sample of 193 mail messages sent to a faculty member, of which 100 were unwanted and 93 were important. This task is more difficult than junk mail filtering because the unwanted mail also included unwanted items such as talk announcements, grant opportunities, and calls for papers that the faculty member was not interested in that were similar in style to important messages.

2. BACKGROUND: RULES, LINEAR MODELS, AND PROTOTYPES FOR MAIL FILTERING

Rules are the most commonly used representation for mail filtering profiles that are hand-coded. Cohen [1996] argues for learning this type of representation: "the greater comprehensibility of the rules may be advantageous in a system that allows users to extend or otherwise modify a learned classier." Figure 1 presents the set of rules learned with Ripper on all 193 examples of mail messages.

Discard if
The BODY contains "our" & "internet"
The BODY contains "free" & "call"
The BODY contains "http" & "com"
The BODY contains "UCI" & "available"
The BODY contains "all" & "our" & "not"
The BODY contains "business" & "you"
The BODY contains "by" & "Humanities"
The BODY contains "over" & "you" & "can"
Otherwise Forward

Figure 1. Rules learned by Ripper for filtering e-mail.

IUI 2000 New Orleans LA USA

Linear models have been shown to form accurate profiles of user interests [e.g., Lewis, Schapire, Callan and Papka, 1996]. Figure 2 shows a linear model learned by a perceptron from the mail examples. The 32 most informative terms were used as binary variables. The linear model fewer than 32 variables because some variables had coefficients equal to 0.

```
IF (  11"remove" + 10"internet" + 8"http" + 7"call" + 7"business"
      +5"center" +3"please" + 3"marketing" + 2"money" + 1"us" +
      1"reply" + 1"my" + 1"free"
      -14"ICS" - 10"me" - 8"science" - 6"thanks" - 6"meeting" -
      5"problem"-5"begins" - 5"I" - 3"mail" - 3"com" - 2"www" -
      2"talk" - 2"homework"-1"our" - 1"it" - 1"email" - 1"all" - 1) >0
THEN Discard
ELSE Forward
```

Figure 2. A linear model for mail filtering learned by a perceptron.

The linear model can be viewed as summing evidence for and against discarding a mail message. Some of the signs of the coefficients in the equations in Figure 2 may be counterintuitive. For example, "com" has a negative coefficient indicating the presence of this term is evidence for forwarding a message, but this term occurs much more frequently in messages that should be deleted. Pazzani & Bay [1999] report that people prefer linear models where the sign of each coefficient in the equation indicates the direction of the correlation between the explanatory variable and the dependant variable.

The third representation we investigate is a "prototype" representation, which can be viewed as summing evidence for or against certain decisions like the perceptron. However, rather than having weights and thresholds, the categorizations are made by a similarity comparison between the example and a prototype. Figure 3 shows the prototype representation learned from the training examples. Later in the paper, we describe the prototype learning algorithm in more detail. The prototype classification process we consider simply categorizes an example to the class whose prototype has the most terms in common with the example.

```
IF the message contains more of
    "papers" "particular" "business" "internet" "http" "money" us"
THAN
    "I "me" "Re:" "science" "problem" "talk" "ICS" "begins"
THEN Discard
ELSE Forward
```

Figure 3. A prototype model for mail filtering

In this paper, we also address whether profiles should use individual words as terms or should also consider pairs of terms. The general finding in a number of papers has been that using pairs of words as terms has a slight positive or no effect on accuracy [e.g., Cohen, 1995]. However, we speculate that if the profile is to be displayed to a user, then the user would prefer profiles with word pairs. In previous research on profiles learned for restaurant recommendations [Pazzani, in press], word pairs made more sense to us because they included terms such as "goat cheese" and "prime rib" rather than just "goat" and "prime". Here we investigate this hypothesis empirically on a group of subjects. Figure 4 shows a prototype that includes word pairs as terms learned from the same e-mail messages as Figure 3. Space limitations prohibit us from showing rule or linear models with word pairs.

```
IF the message contains more of
    "service" "us" "marketing" "financial" "the UCI"
    "http:// www" "you can" "removed from" "com"
THAN
    "I" "learning" "me" "Subject: Re:" "function"
    "ICS" "talk begins" "computer science" "the end"
THEN Discard
ELSE Forward
```

Figure 4. A prototype model with word pairs as terms.

The final issue we investigate empirically is whether a person can detect whether the profile in a particular representation is accurate. We intentionally created inaccurate profiles by introducing noise into the classification of the training examples. We achieved this by inverting the categorization of 20% of the examples chosen at random. These profiles make mistakes on 20% of the original e-mail.

The goal of this paper is to explore alternative representations of user profiles. We speculate that various representations and representational changes affect the willingness of people to use the results of text mining algorithms to create understandable profiles of a user's interests. In the next section, we treat these intuitions as hypotheses and conduct a study that evaluates the following hypotheses:

1. Prototype representations are more acceptable to users than rule representations.
2. Prototype representations are more acceptable to users than linear model representations
3. Using word pairs as terms increases the acceptance of profiles (for linear models, prototypes, and rules).
4. Inaccurate profiles (learned from noisy training data) are less acceptable to users than accurate ones (for linear models, prototypes, and rules).

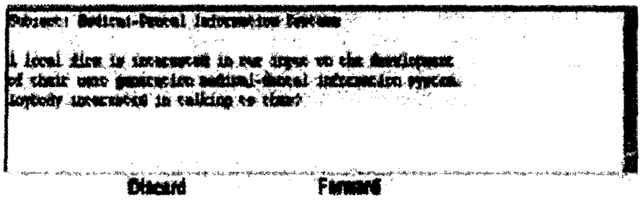
3. MAIL FILTERING PROFILES: AN EXPERIMENT

In this section, we report on a study in which subjects were first asked to filter mail manually, and then were asked to judge the various profiles learned from the data that were shown in the figures in the previous section. We decided to focus on a two-choice task (i.e., either forward or discard mail) rather than filing to many different folders to establish results on people performing the simplest case before addressing the more complex tasks.

Subjects. The subjects were 133 male and female students majoring in Information and Computer Science at the University of California, Irvine who participated in this experiment to receive extra credit.

Stimuli. The stimuli consisted of 193 mail messages and 9 mail filtering profiles. The profiles were generated by Ripper, a perceptron, and a prototype learner. For each learner, they learned once with a representation that included only individual words, once with a representation that included word pairs as terms, and once with word pairs learned from data with 20% noise added.

Procedures. Subjects were asked to imagine that they were the assistant of a faculty member and their job was to decide whether to forward or discard mail messages sent to the faculty member. An example of the stimuli is shown in Figure 5.



Subject: Medical-Dental Information Systems

A local firm is interested in new input to the development of their own generic medical-dental information system. Anybody interested in talking to them?

Figure 5. An example of unwanted electronic mail.

In the first phase of the experiment, each subject was shown 40 mail messages one at a time. For each message, the subject indicated whether the message should be forwarded or discarded and received feedback on whether the decision was correct. The mail messages were selected randomly without replacement. We recorded the number of errors made by subjects on each block of ten messages.

In the final phase of the experiment, each subject was shown the nine mail filtering profiles one at a time in random order and asked to indicate on a scale from -3 to +3 how willing they would be to use that profile to perform the same decisions that they had just made for forwarding or discarding mail. They indicated a rating by selecting a radio button. Next they clicked on "Record Rating" and were shown another profile. The radio button was reset to 0 before displaying the next profile. This continued until the subject rated all 9 profiles. Figure 6 shows an example display. We recorded the rating of the subject on each profile to allow us to determine what types of profiles subjects would be most willing to use.

Results. An analysis of subject errors showed that subjects improved by getting feedback. In the first block of ten messages, 15 of the 133 subjects made 0 or 1 errors. In the next block, 66 subjects improved to this level, followed by 83 on the third block of ten and 81 on the fourth block.

The average rating of subjects for each type of mail profile is shown in Table 1. We performed a paired one-tailed t-test to determine which of the eight hypotheses discussed in Section 2 were supported by the experimental findings. We used a Bonferroni adjustment to account for the fact that we are making 8 comparisons.

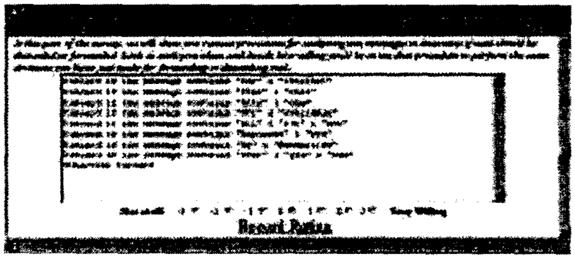


Figure 6. Subjects provide feedback on mail profiles.

The following differences were highly significant (at least at the .005 level).

- Prototype representations with word pairs received higher ratings than rule representations with word pairs t(132) = 5.64.
- Inaccurate prototype models (learned from noisy training data) are less acceptable to users than accurate ones t(132)= 4.88.

Algorithm	Mean Rating
Rules	0.015
Rules (Pairs)	-0.135
Rules (Noise)	-0.105
Linear Model	0.421
Linear Model (Pairs)	0.518
Linear Model (Noise)	-0.120
Prototype	0.677
Prototype (Pairs)	1.06
Prototype (Noise)	0.195

Table 1. Mean rating of subjects on each type of mail profile.

The following differences were significant (at least at the .05 level).

- Prototype representations with word pairs received higher ratings than linear model representations with word pairs t(132) = 2.84.
- Inaccurate linear models are less acceptable to users than accurate ones. t(132)=2.99.

The following difference was marginally significant (between the 0.1 and .05 level).

- For prototype representations, using word pairs as terms increases user ratings: t(132) = 2.37.

4. DISCUSSION

The results of the study confirm many of the intuitions. We had initially planned to focus only on rule representations in this study, exploring the differences between individual words and words pairs. However, after seeing the results of Ripper (and other rule learners) under a wide variety of parameter settings, it became apparent that although it is easy to understand how the learned rules classify text, the learned rules do not seem credible. We believe this is because subjects can easily imagine counterexamples to any rule. For example, rules that filter out mail that contains the term "XXX" assuming it is pornographic, fail on mail about "Superbowl XXX." Subjects greatly preferred prototypes to rules, and subjects did not prefer accurate rules to less accurate rules learned on noisy data. Increasing the representational power of the learned rules to include word pairs did not help increase user confidence in the rules.

The prototype representation with word pairs received higher ratings than all other profiles. The differences between this profile and linear models with word pairs, rule models with word pairs, and prototype models learned from noisy data were all significant. The difference between prototype models with only individual words as terms (0.677) and prototypes with word pairs (1.06) is only marginally significant in the context of the 8 comparisons being done in this study. The linear model is not as acceptable to users as the prototype model[1]. However, users were able to distinguish a linear model that was accurate on the training data to one that was learned from noisy data. This may be the result of noise influencing the terms. We don't believe users can distinguish small changes in coefficients that could affect the accuracy.

5. LEARNING TEXT CLASSIFICATION PROTOTYPES

Subjects in our study had a preference for the prototype representation for text classifiers. We feel this is a useful representation because it has a relatively small number of meaningful words for each class. Here, we describe how the prototype representation is constructed. Initial attempts to create prototypes by simply selecting the *k* words with the highest value according to some criteria (e.g., information gain) did not result in profiles that were accurate classifiers. We adopted a genetic algorithm approach to create a prototype for each class.

To learn a pair of prototypes for text classification, first each training example is converted to a bit vector of length 128 where each bit represents the presence or absence of an individual word. The 128 most informative terms (i.e., those that best distinguish positive examples of discard from negative examples) are selected binary features. An individual in the population is represented as a bit vector of length 256, the first 128 bits representing the prototype for discard and the remaining 128 representing a prototype for discard. In the individual, a 1 indicates that the term is present in the prototype and a 0 represents that it is not present in the prototype. The classification procedure for prototypes simply counts the number of times a 1 appears in the example (indicating that the term is present in the example) and a 1 occurs in the corresponding location of the discard and forward prototypes. If there are more terms in common with the discard prototype than the forward prototype, the message is classified as discard. Otherwise, it is classified as forward.

The genetic algorithm operates by first initializing the population to 100 individuals. An individual of the initial population is formed by concatenating a randomly selected positive example with a randomly selected negative example. Each individual is then scored with a fitness function that simply checks for accuracy on the training example. To produce the next generation, the following procedure is used. Two individuals are selected from the population with a probability proportional to the individual fitness. Next a new individual is created through application of the crossover operator and a mutation operator (with a bit replaced with a random bit with a probability of 0.005). Once 100 new individuals are created, the 100 most fit of the new and previous generation are retained. The genetic algorithm is allowed to run until 10 generations produce no improvement in the fitness function or for 100 generations. The fittest individual is returned. Experiments with this algorithm showed that it is comparable in accuracy to other models, such as Rocchio's method and a naïve Bayesian classifier.

One drawback of the prototype representation involves the ease with which a user may edit the prototype representation. In particular, it would be hard for a user to anticipate the effects of adding or deleting a term from a prototype. An editing

environment in which these effects are easily visualized on a training set is planned.

6. ACKNOWLEDGMENTS.

This research was funded in part by the National Science Foundation grant IRI-9713990. Comments by Dorrit Billman, Robin Burke, Daniel Billsus and Cathy Kick on an earlier draft of this paper help to clarify some issues and their presentation.

7. CONCLUSION

We have explored factors that affect the user preferences of automatically learned mail filtering profiles. By asking subjects to rate learned profiles that automate a task that the subjects had learned to perform, we found that subjects have little confidence in learned rules for text classification. A prototype representation was developed, and experiments showed that it is competitive in accuracy with other text classifiers but more readily accepted by users. In addition, the findings suggest that using word pairs as terms may improve the user acceptance of learned prototype profiles.

8. REFERENCES

[1] Cohen, W. (1996). *Learning Rules that Classify E-Mail* In the 1996 AAAI Spring Symposium on Machine Learning in Information Access.

[2] Duda, R. & Hart, P. (1973). *Pattern classification and scene analysis*. New York: John Wiley & Sons.

[3] Dumais, S., Platt, J., Heckerman, D., and Sahami, M. (1998). Inductive learning algorithms and representations for text categorization. Proceedings of ACM-CIKM98.

[4] Lewis, D., Schapire, R., Callan, J., & Papka, R. (1996). *Training algorithms for linear text classifiers*. SIGIR, 298-306.

[5] Pazzani, M. & Billsus, D. (1997). *Learning and Revising User Profiles: The identification of interesting web sites*. Machine Learning, 27, 313-331.

[6] Pazzani, M. (in press). *A Framework for Collaborative, Content-Based and Demographic Filtering*. Artificial Intelligence Review.

[7] Quinlan, J. (1993). C4.5: *Programs for Machine Learning*. Morgan Kaufmann, Los Altos, California.

[1] A caution on comparisons between representations is in order. In particular, there may be better ways to visualize a linear model for the user. For example, we could display words with positive coefficients in green and negative coefficients in red, and represent the magnitude of the coefficient by the brightness of the color.

[8] Rocchio, J. (1971). *Relevance feedback information retrieval*. In Gerald Salton (editor), The SMART retrieval system- experiments in automated document processing (pp. 313-323). Prentice-Hall, Englewood Cliffs, NJ.

[9] Sahami, M., Dumais, S., Heckerman, D. and E. Horvitz (1998). *A Bayesian approach to filtering junk e-mail.* AAAI'98 Workshop on Learning for Text Categorization, Madison, Wisconsin.