Toward Harnessing User Feedback For Machine Learning

Simone Stumpf, Vidya Rajaram, Lida Li, Margaret Burnett,
Thomas Dietterich, Erin Sullivan, Russell Drummond, Jonathan Herlocker
Oregon State University
School of Electrical Engineering and Computer Science
Corvallis, OR 97331 USA
1-541-737-3617
{stumpf,rajaramv,lili,burnett,tgd,sullivae,drummond,herlock}@eecs.oregonstate.edu

ABSTRACT

There has been little research into how end users might be able to communicate advice to machine learning systems. If this resource—the users themselves—could somehow work hand-in-hand with machine learning systems, the accuracy of learning systems could be improved and the users' understanding and trust of the system could improve as well. We conducted a think-aloud study to see how willing users were to provide feedback and to understand what kinds of feedback users could give. Users were shown explanations of machine learning predictions and asked to provide feedback to improve the predictions. We found that users had no difficulty providing generous amounts of feedback. The kinds of feedback ranged from suggestions for reweighting of features to proposals for new features, feature combinations, relational features, and wholesale changes to the learning algorithm. The results show that user feedback has the potential to significantly improve machine learning systems, but that learning algorithms need to be extended in several ways to be able to assimilate this feedback.

ACM Classification: H.5.2 [Information interfaces and presentation (e.g., HCI)] User Interfaces: Theory and methods, Evaluation/methodology. H.1.2 [Models and Principles]: User/Machine Systems: Human information processing, Human factors.

General terms: Algorithms, Human Factors, Design.

Keywords: Machine learning, explanations, user feedback for learning.

INTRODUCTION

Statistical Machine Learning methods have the potential to improve user interfaces by learning models of user behavior and applying these models to optimize and customize for each user. However, statistical methods rarely achieve 100% accuracy. Therefore, approaches have begun to emerge in which the user and machine learning component *communicate with each other* to improve the machine's accuracy or otherwise supplement the machine's inferences by incorporating user feedback.

The norm for learning systems that communicate with users is to allow the user to indicate only that a prediction was wrong or to specify what the correct prediction should have been. This is just a glimpse of the rich knowledge users have about the correct prediction, and we would like to better harness the user's knowledge to improve learning.

In this paper, we explore the possibility of closer and richer collaboration between machine learning systems and the user. If the machine learning system could explain its reasoning more fully to the user, perhaps the user would, in return, specify *why* the prediction was wrong and provide other, rich forms of feedback that could improve the accuracy of learning systems.

Such a step forward requires two directions of communication. First, the system's explanations of why it has made a prediction must be usable and useful to the user (c.f. [12]). Second, the user's explanations of what was wrong (or right) about the system's reasoning must be usable and useful to the system.

To investigate possibilities for both directions of communication, we conducted a formative think-aloud study with email users. In the study, machine learning algorithms sorted email messages into folders and explained their reasoning using three different explanation paradigms: Rule-based, Keyword-based, and Similarity-based. The participants were asked to provide feedback to improve the predictions. No restrictions were placed upon the form or content of participants' feedback.

From a user perspective, we assessed the participants' willingness to provide feedback, accuracy in doing so, and ability to understand the different explanations. From an algorithm perspective, we analyzed the participants' feedback to determine how easily its types of advice could be understood and assimilated by machine learning algorithms. Our research questions were the following:

IUI'07, January 28–31, 2007, Honolulu, Hawaii, USA.
Copyright 2007 ACM 1-59593-481-2/07/0001...$5.00.

1. Is it possible for machine learning systems to explain themselves such that users (a) can understand the system's reasoning, and (b) can provide the system rich, informative feedback that could improve the system's accuracy?

2. What types of user feedback could be assimilated by existing learning algorithms, what sources of background knowledge underlie the users' feedback, and how much of this knowledge could be incorporated easily?

RELATED WORK

Although various supervised learning algorithms have been developed to automatically classify email messages into a set of categories or folders defined by users (e.g., [5, 7, 25]), the reported accuracy of these algorithms indicates that email classification is a very challenging problem. The challenges stem from numerous factors, such as imbalanced categories, incomplete information in the email messages, and the fact that the categories (folders) set up by the users are often idiosyncratic and non-orthogonal. These challenges are what motivate our interest in using rich user feedback to improve email predictions.

A first step in any effort to obtain user feedback about a learning system is to ensure that explanations by the learning system are understandable and useful to users. It has been shown that explanations that answer why certain outcomes happened, based on user actions, can contribute positively to system use [11, 18]. Similarly, it has been shown that highlighting the relationship between user actions and ensuing predictions can influence user preference [2]. There is also previous research on the characteristics of explanations that help users choose between predictions. For example, showing contrasting features in recommendations can play a role in user trust [11, 23]. One way that this relationship can be expressed is by making use of various ways of reasoning, such as analogical reasoning [14]. Of particular relevance to our work is a complementary study conducted by Pazzani [20]. In this experiment, users were asked which email learning system they trusted more to classify the email correctly as junk or not junk, given the choice between rules, signed-weighted keywords, and a new approach that used general descriptions employing keywords.

Different methods for gathering user feedback have also been investigated, along a spectrum of formality and richness. An obvious way to gather user feedback is to allow interactions in natural language [4]. Semi-formal types of feedback that have been shown to be preferred by users make use of editing feature-value pairs [16, 6]. Other approaches allow the user to edit models produced by a learning algorithm using a formal description language [19]. However, so far there has been a lack of research that integrates an investigation into the understanding of machine learning systems' explanations with an analysis of the *content* of the rich feedback users give when they have an unconstrained opportunity to do so.

EXPERIMENT SET-UP

To maximize external validity, it was important to base our experiment on real-world data. To allow as thorough investigation of users' potential as possible, it was also important to allow participants to express feedback freely. Thus, our first two design principles were:

(P1) Real-world email data: 122 messages from a user's email (farmer-d), which had sufficient content for human and machine classification, were drawn from the publicly available Enron dataset [13]. (Our data will be provided upon request.) The emails had been categorized by the user into Personal, Resume, Bankrupt, and Enron News folders.

(P2) Rich collecting of result data: We employed a qualitative "think-aloud" design in order to extract the richest possible data from the participants. We observed and videotaped their activities and comments throughout the experiment, as well as collecting their work products.

First, learning algorithms classified each email message. Then, three explanations of each result were generated: a Rule-based, a Keyword-based, and a Similarity-based explanation. The application of the classification algorithms and the generation of explanations, described in the next section, were all done off-line prior to the experiment.

The experiment followed a within-subject design where each participant experienced all three explanation paradigms. We counterbalanced learning effects in our design by randomizing the order of explanation paradigms that each participant experienced. The participants were 13 graduate and undergraduate students (7 females, 6 males). All had previous experience using computers but did not have computer science backgrounds. All were native English speakers.

Low-fidelity prototypes are important for experiments aiming to encourage participant feedback, because they avoid the impression of a "finished" product [24]. Thus, we used printouts of emails instead of an on-line display. This set-up also allowed for flexibility and ease of feedback. Using pens, printouts, and a big table to support spatial arrangements (Figure 1), participants could move papers around to compare them, scratch things out, draw circles, or write on

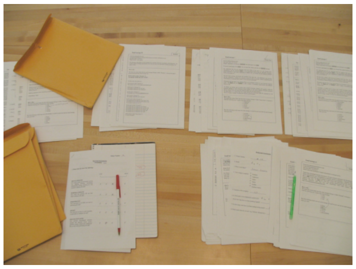


Figure 1: Lo-fi prototype set-up with pens, printouts, table.

them in any way they chose (Figure 2).

The experiment was conducted one participant at a time with a facilitator interacting with the participant and an observer taking additional notes. First, the participant was familiarized with thinking aloud. Next, he or she looked through 40 sample pre-classified email messages to become familiar with the folders and to develop an intuition for how new email messages should be categorized; this sample was kept the same across participants. At this point, the main task began, divided into three 15-minute blocks (one per explanation paradigm) of processing the remainder of email messages from the dataset.

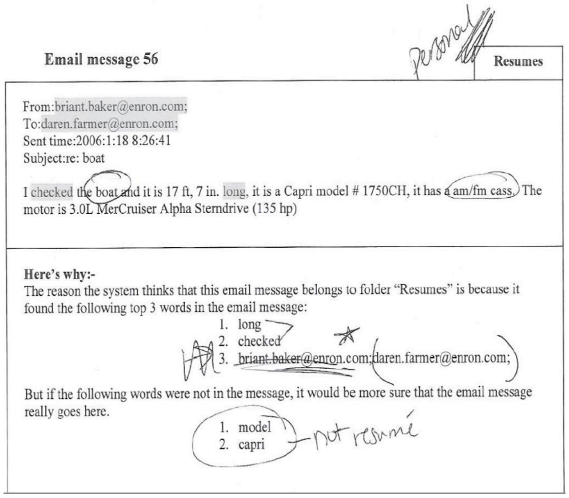
For the main task, we randomized assignments of emails to explanation paradigms to avoid exclusive association of an email with just one paradigm. For each message, the facilitator handed a new printout to the participant, who decided whether the predicted folder classification was correct, reclassified the message if needed, and gave feedback to improve the classification if needed. The participants were told that an evolving "virtual email assistant" had been implemented, and that we wanted their help "in order to get these predictions working as well as they possibly can."

After each paradigm's 15-minute block, participants provided subjective self-evaluations of mental effort, time pressure, overall effort, performance success, and frustration level, based on standard NASA TLX questions [10]. They also took a comprehension test for that paradigm. Finally, at the end of the study, they compared all three explanation paradigms in terms of overall preference, ease of understanding, and ease of feedback.

EXPLANATIONS OF THE LEARNING ALGORITHMS

We generated the explanations under the following additional three design principles:

(P3) Common algorithms: We focused on standard implementations of machine learning algorithms found in Weka [26] that were viable for (a) generating explanations and (b) good performance in the email domain.



Email message 56 personal **Resumes**

From:briant.baker@enron.com;
To:daren.farmer@enron.com;
Sent time:2006:1:18 8:26:41
Subject:re: boat

I checked the boat and it is 17 ft, 7 in. long, it is a Capri model # 1750CH, it has a am/fm cass. The motor is 3.0L MerCruiser Alpha Sterndrive (135 hp)

Here's why:-
The reason the system thinks that this email message belongs to folder "Resumes" is because it found the following top 3 words in the email message:

1. long
2. checked
3. ~~briant.baker~~@enron.com;daren.farmer@enron.com;

But if the following words were not in the message, it would be more sure that the email message really goes here.

1. model
2. capri

not resumé

Figure 2: Example of participant feedback.

(P4) Simplified but faithful explanations: It does not seem reasonable to provide end users a complete explanation of a statistical learning algorithm. Instead, we sought to develop explanations that would be informal, yet accurate enough to engender useful mental models of this reasoning, analogous to "naïve physics" descriptions of qualitative physics.

(P5) Concrete explanations: The explanations were required to be in terms of specific features that were visible in the current email message.

Learning Algorithms and Training

We chose two learning algorithms: the Ripper rule-learning algorithm [7] and Naïve Bayes probabilistic learning algorithm. These algorithms have been widely applied for email classification (e.g., [7, 8, 25]). To obtain a prediction for each of the 122 email messages, we performed a stratified 5-fold cross-validation.

Prior to training, each email message was preprocessed to remove headers and common "stop" words. The remaining words were stemmed by Porter's method to remove word endings [22]. Each email message was then represented as a Boolean vector with a Boolean feature for each observed email sender (the From field), one Boolean feature for each observed set of email recipients (the union of the From, To, CC, and BCC fields)[1], and one Boolean feature for each distinct word observed in the Subject and Body fields.

Ripper learns a set of classification rules. The rules are ordered by class but unordered within class. Hence, to make a prediction, Ripper first applied the rules for the least frequent class (Bankrupt in our dataset). If one of these rules matched the email message, it was classified as Bankrupt. Otherwise, Ripper moved on to the rules for the next most frequent class (Resume), and so on. There were no rules for the most frequent class (Enron News); it was treated as the default if none of the rules for the other classes matched.

Naïve Bayes can be viewed as learning a weight (positive or negative) w_{jk} for each word j and each email folder k. Hence, to predict the email folder, it computed a score for folder k as

$$score(k) = \sum_j w_{jk} \cdot x_j$$

where x_j was the jth Boolean feature in the email message. It then predicted the folder with the highest score.

The overall accuracy of the predictions was not particularly

[1] In contrast to using a separate feature for each recipient, our approach identifies to which "team" of people the message relates [25].

high: 60% for Naïve Bayes and 75% for Ripper when used to classify the entire set of emails. We would have preferred higher accuracy and equal accuracy between algorithms. Still, high accuracy was not required to answer our experiment's research questions, and our analysis takes accuracy differences into account.

Generating Explanations

The Rule-based explanations (Figure 3) were generated by highlighting the rule that made the classification, and listing it above all other possible rules.

The Keyword-based and Similarity-based explanations were both generated from the learned Naïve Bayes classifier. Consistent with our fifth design principle ("visible words only"), the Keyword-based explanations (Figure 4) were generated by listing up to five words *present in the email message* having the largest positive weights as well as up to five words *present in the email message* having the most negative weights. (As we will discuss later, users found this latter set of "negative" words counter-intuitive. They are the words whose presence in the message *reduces* the certainty of the classifier in the sense that the classifier would be more confident if these words did *not* appear.)

The Similarity-based explanations (Figure 5) were generated by computing the training email message that, if deleted from the training set, would have most decreased the score. This was typically the training example most similar to the email message being classified. This example was then displayed and up to five words with the highest weights that appeared in both the example and the target email message were highlighted.

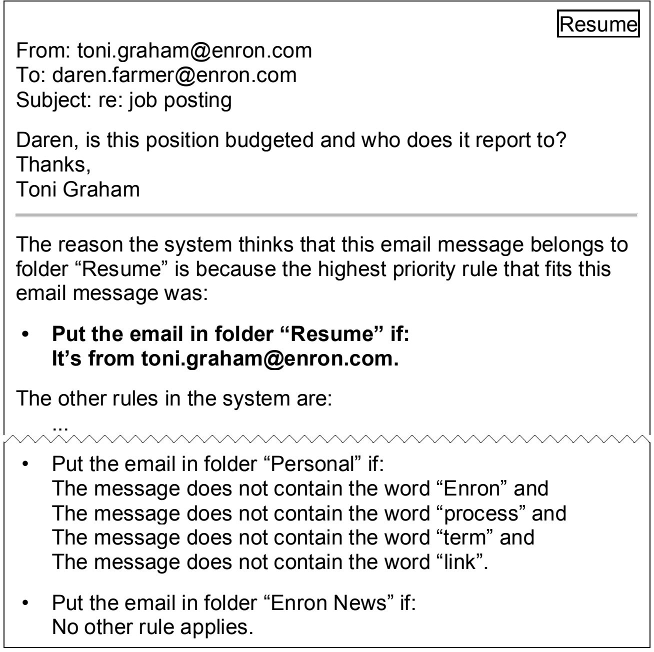
METHODOLOGY FOR QUALITATIVE ANALYSIS

Derivation of Codes

The data were analyzed using two coding schemes. In the main coding scheme, we coded all user utterances with the goal of determining the reaction of users to the explanations. In the second coding scheme, we performed a more detailed analysis of only the utterances that constituted negative comments about the explanations or suggested changes to the learning algorithms. The goal of the second coding scheme was to assess how much of the users' feedback could be assimilated by the learning algorithms and the background knowledge required to do so.

For both schemes, to ensure consistency in interpretation of the codes and when to use them, two researchers independently coded a small subset. They then iterated on this subset, further refining the codes and developing norms about how to apply them. For the main coding scheme, the total agreement value was 81% for the first subset at the end of these iterations, which indicates high coding reliability. For the second coding scheme, the agreement was 82% after the iterations. At this point, the schemes were deemed robust enough, and the remaining data were then coded.

For both coding schemes, we calculated the inter-rater agreement as the percentage of intersection of the codes divided by the union of all codes applied. For example, if one researcher gave the codes {Breakdown, Suggest Change} for one email and another researcher gave the



Resume

From: toni.graham@enron.com
To: daren.farmer@enron.com
Subject: re: job posting

Daren, is this position budgeted and who does it report to?
Thanks,
Toni Graham

The reason the system thinks that this email message belongs to folder "Resume" is because the highest priority rule that fits this email message was:

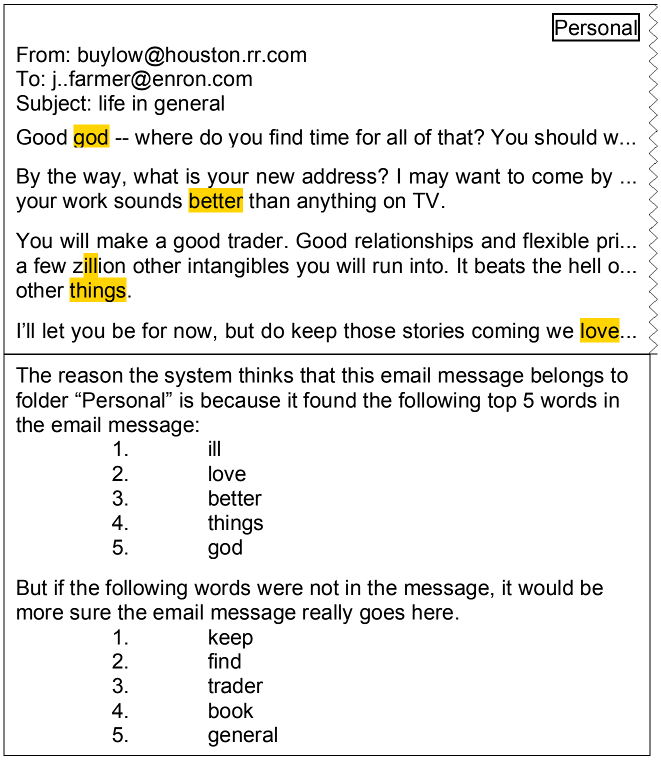
- **Put the email in folder "Resume" if:**
  It's from toni.graham@enron.com.

The other rules in the system are:

...

- Put the email in folder "Personal" if:
  The message does not contain the word "Enron" and
  The message does not contain the word "process" and
  The message does not contain the word "term" and
  The message does not contain the word "link".
- Put the email in folder "Enron News" if:
  No other rule applies.

Figure 3: (Top): Email.
(Bottom): Rule-based explanation excerpt.



Personal

From: buylow@houston.rr.com
To: j..farmer@enron.com
Subject: life in general

Good god -- where do you find time for all of that? You should w...

By the way, what is your new address? I may want to come by ...
your work sounds better than anything on TV.

You will make a good trader. Good relationships and flexible pri...
a few zillion other intangibles you will run into. It beats the hell o...
other things.

I'll let you be for now, but do keep those stories coming we love...

The reason the system thinks that this email message belongs to folder "Personal" is because it found the following top 5 words in the email message:

1. ill
2. love
3. better
4. things
5. god

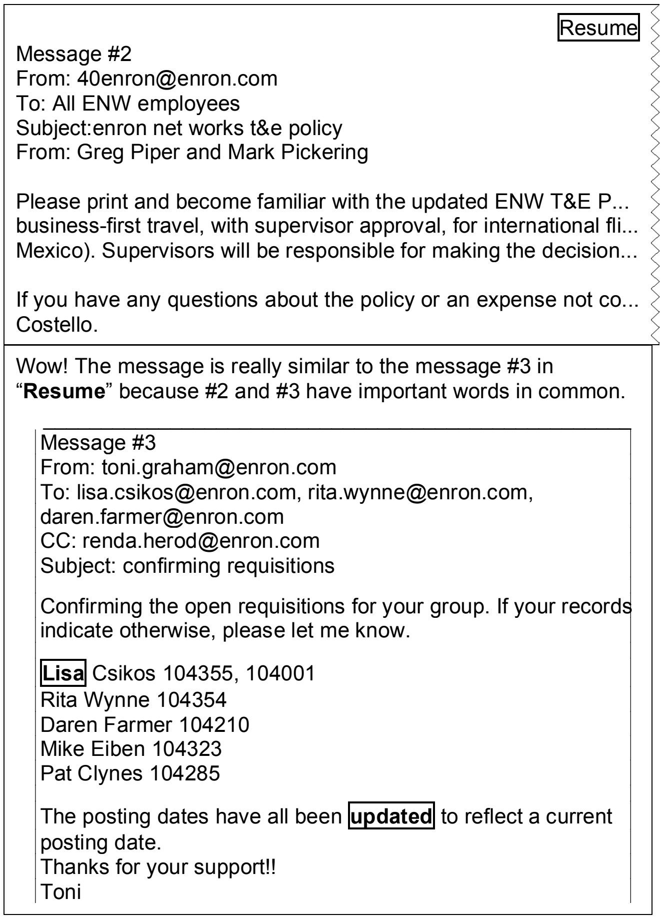
But if the following words were not in the message, it would be more sure the email message really goes here.

1. keep
2. find
3. trader
4. book
5. general

Figure 4: (Top): Excerpt from email.
(Bottom): Keyword-based explanation, supplementing the highlights in the email.

codes as {Emotion, Suggest Change} for the same email, then the agreement was calculated as 1/3 (33%) as follows:

$$\frac{|\{\mathit{Breakdown}, \mathit{SuggestChange}\} \cap \{\mathit{Emotion}, \mathit{SuggestChange}\}|}{|\{\mathit{Breakdown}, \mathit{SuggestChange}\} \cup \{\mathit{Emotion}, \mathit{SuggestChange}\}|}$$



Message #2
From: 40enron@enron.com
To: All ENW employees
Subject:enron net works t&e policy
From: Greg Piper and Mark Pickering

Please print and become familiar with the updated ENW T&E P...
business-first travel, with supervisor approval, for international fli...
Mexico). Supervisors will be responsible for making the decision...

If you have any questions about the policy or an expense not co...
Costello.

Wow! The message is really similar to the message #3 in "**Resume**" because #2 and #3 have important words in common.

Message #3
From: toni.graham@enron.com
To: lisa.csikos@enron.com, rita.wynne@enron.com, daren.farmer@enron.com
CC: renda.herod@enron.com
Subject: confirming requisitions

Confirming the open requisitions for your group. If your records indicate otherwise, please let me know.

Lisa Csikos 104355, 104001
Rita Wynne 104354
Daren Farmer 104210
Mike Eiben 104323
Pat Clynes 104285

The posting dates have all been **updated** to reflect a current posting date.
Thanks for your support!!
Toni

Figure 5: (Top): Excerpt from email.
(Bottom): Its Similarity-based explanation.

The main codes, along with a description and example are shown in the first three columns of Table 1. The second coding scheme is discussed in Results (Part 2).

RESULTS (PART 1): EXPLAINING TO USERS

Analyzing the video transcripts and questionnaires using the coding scheme just described produced the counts shown in the final column of Table 1. (We will not discuss further the codes making up less than 1% of the total.)

Explaining to Users: Understandability

If the behavior of any system is not understandable to users, they cannot form an accurate mental model of how the system works and cannot use it well. Further, if they do not understand how it works, their feedback is less likely to contain information useful for enhancing the algorithm.

Which Paradigms Did They Understand?

According to the participants' responses to the questionnaires, the Rule-based explanation paradigm was the most understandable (Table 2). This was corroborated by their verbal remarks: Rule-based explanations generated three times as many remarks indicating understanding and less than a tenth the remarks indicating breakdowns as either Keyword-based or Similarity-based explanations.

Differentiating between shallow and deep understanding reveals further insights. "Shallow understanding" in this context means that participants were simply able to make the classification decision the same way as the explanation paradigms. In the questionnaires, we included an email without a classification or explanation, and asked the participants to categorize it to a folder based on what the paradigm would predict. Nearly all participants categorized it correctly in the case of Rule-based explanations and Key-

Code	Description	Example from data	Count (% of total)
Breakdown	Expressing confusion or lack of understanding with the explanation of the algorithm.	I don't understand why there is a second email.	41 (8%)
Understand	Explicitly showing evidence of understanding the explanation of the algorithm.	I see why it used "Houston" as negative	85 (17%)
Emotion	Expressing emotions.	It's funny to me.	15 (3%)
Trust	Stating that he or she trusted the system.	I would probably trust it if I was doing email.	1 (<1%)
Expectation	Expressing an expectation for the system to behave in a certain way.	I hope that eventually the intelligent assistant would learn to give more reasons.	2 (<1%)
Suggest change	Correcting the explanations or otherwise suggesting changes to the system's reasoning.	Different words could have been found in common, like "Agreement," "Ken Lay."	161 (32%)
Negative comment	Making negative comments about the explanation (without suggesting an improvement).	…arbitrary words: "energy" especially bad.	100 (20%)
Positive comment	Making positive comments about the explanation.	The Resume rules are good.	94 (19%)

Table 1: The main coding scheme.

word-based explanations, but only 4 of the 13 participants categorized the email correctly in the Similarity-based case.

"Deep understanding" implies understanding the *reasoning behind* the classification decision of the explanation paradigms. The questionnaires included an email with a classification but without the explanation, and participants were asked *why* the paradigm would classify an email the way it did. For the Rule-based explanation paradigm, a majority of participants answered by giving a rule, and some even managed to reconstruct a close version of the actual rule that was applied. For Keyword-based explanations, nearly all participants answered with keywords, even managing to identify correctly some of the keywords used in the actual example. However, only three participants answered even close to correctly for the Similarity-based case.

The evidence is thus quite strong that the Similarity-based explanations had a serious understandability problem.

What Factors Affected Understanding?

We investigated the factors that contributed to understanding via the Understand, Breakdown, and Negative Comments codes from the video transcripts and written questionnaire comments. Three factors stood out in affecting understanding of the system's behavior: understanding of the general idea of the algorithm, the Keyword-based explanations' negative keyword list, and appropriateness of word choices.

Regarding understanding of the algorithm, some participants expressed understanding of the algorithm by describing the essential strategy, as in the following two quotes. This enabled them to predict system behavior.

> *P6 (on Rule-based):* "I understand why it would just default to Enron, since that's what the rule is."
> *P1 (on Similarity-based):* "I guess it went in here because it was similar to another email I had already put in that folder."

In the case of the Keyword-based paradigm, some problems in understanding were caused by the negative keyword list. Nobody had anything positive to say about the inclusion of negative keywords in the explanation:

> *P6 (on Keyword-based):* "So what does this mean (referring to 2nd set of words)?"
> *P8 (on Keyword-based):* "I guess I really don't understand what it's doing here. If those words weren't in the message?"

Finally, appropriateness of the word choices seemed to have an effect on understanding, especially if they were felt by participants to be common words or topically unrelated:

> *P1 (on Similarity-based):* "'Day', 'soon', and 'listed' are incredibly arbitrary keywords."

Discussion: Understanding.

In addition to the clear evidence of understandability problems for Similarity-based explanations, we note three results of particular interest.

First, although Rule-based explanations were consistently understandable to more than half the participants and, at least for this group of participants, seemed to "win" over the other two paradigms, note that about one-third of the participants preferred one of the other explanation paradigms. This implies that machine learning systems may need to support *multiple* explanation paradigms in order to effectively reach all of their users.

Second, Keyword-based explanations seemed to be reasonably understandable except for the negative keyword list, which our results suggest was a problem. There are several potential remedies. One possibility is that the negative keyword list could be explained in some different way to give users a better understanding of how the algorithm works. For example, instead of drawing attention to words with negative weights that are present in the email, the explanation could make use of the strongest negative weights associated with words that are *absent* from emails, since their absence increases the confidence of the learning algorithm. Another possibility is that the negative keyword list should be omitted from the explanation altogether.

Third, the *topical* appropriateness of word choices seemed particularly critical to participants' ability to predict and understand system behavior. This knowledge is too complex to be learned from only 122 email messages, but it could be possible in larger document collections; we will return to this point in the Results (Part 2) section.

Explaining to Users: Preferred Paradigms and Why

Following the understanding trends, participants' ratings favored the Rule-based explanations over the other two (Table 3). Still, nearly 50% of the participants chose a paradigm other than Rule-based as their favorite, so the Rule-based paradigm did not receive a clear mandate.

We expected preference to closely follow understanding trends, but analysis of the Positive Comments, the positive Emotion codes, and the questionnaire responses provided additional useful insights into factors that seemed to affect participants' preferences in positive ways. These remarks fell into four categories, three of which (approval of reason-

Explanation	Rank 1	Rank 2	Rank 3
Rule-based	9	2	2
Keyword-based	3	6	4
Similarity-based	1	5	7

Table 2: Participants' rankings from the written questionnaires. (Rank 1 is "understood the most.")

Explanation	Rank 1	Rank 2	Rank 3
Rule-based	7	4	2
Keyword-based	3	4	6
Similarity-based	3	5	5

Table 3: Participants' rankings from the written questionnaires. (Rank 1 is "preferred the most.")

ing soundness, clear communication of reasoning, and perceived accuracy) tie at least somewhat to understandability.

Participants' approval of soundness of reasoning was remarked upon often. Also, clear *communication* of reasoning, which is distinctly different from the mere presence of sound reasoning, mattered to a number of our participants. For example, Participant 1's comment below is fairly representative of several about the reasoning itself, whereas Participant 10's comment exemplifies several comments specifically about communication of the reasoning:

> *P1 (on Keyword-based):* "The reasons seem like perfectly good reasons…this is a good reason why it shouldn't be in Personal."
> *P10 (on Similarity based)*: "I like this one because it shows relationship between other messages in the same folder rather than just spitting a bunch of rules with no reason behind it."

High accuracy, as perceived by the participants, was remarked upon often. (We will return to the influence of *actual* accuracy shortly). For example:

> *P11 (on Rule-based):* "I think this is a really good filter. Put in Resume if it's from toni.graham"
> *P2 (on Similarity-based):* "Similarity was my favorite - seemed the most accurate, and took email addresses into account."

The fourth category was unexpected: Several participants appreciated Similarity-based explanations' less technical style of expression, a characteristic we inadvertently introduced in our wording that emphasizes similarity ("Wow!"). This introduction of informality in the form of slang produced a number of Positive Comments for that explanation paradigm, pointing out possible benefits from relaxing the language style used in explanations. For example:

> *P1 (on Similarity-based): "*I also appreciate how the computer is excited about its decision... It's funny to me ... Told you, in conversational form, why it was similar."
> *P10 (on Similarity-based):* "This is funny... (laughs) ... This seems more personable. Seems like a narration rather than just straight rules. It's almost like a conversation."

Accuracy

As we have mentioned, the algorithms performed at different accuracy rates, with Ripper outperforming Naïve Bayes. (We define "accurate" as being in agreement with the original Enron user who owned the email.) It surprised us that Ripper was so much more accurate than Naïve Bayes. This suggests that Ripper may be a better choice than Naïve Bayes for accuracy in this type of situation. As to our experiment, accuracy rate was not statistically predictive (via linear regression) of any of the ratings provided by participants, did not result in differences in participants' willingness to provide feedback, and did not affect their accuracy in doing so.

When the participants disagreed with the machine (28% of the time), participants were usually right (22%), but not always (6%). Ultimately, the participant corrections brought the accuracy rates for all paradigms to almost identical levels: 71-72%. Also, both the machine and the participants disagreed with the original user 22% of the time, suggesting some knowledge possessed only by the original user and perhaps even some misfiling by the original user.

As the preceding paragraph points out, the participants were not perfect oracles. The error rate is consistent with earlier findings regarding end-user programmers' accuracy in serving as oracles when debugging, which have reported error rates of 5-20% (e.g., [21]). This error rate seems fairly robust across studies, and suggests a similar level of "noise" that users' judgments would introduce into the learning algorithm's data.

There was an odd relationship between actual accuracy and participants' speed (efficiency). With Keyword-based and Similarity-based explanations, there was no significant relationship between accuracy and participants' speed, but in the case of Rule-based explanations, the predictive effect was negative! More specifically, an interaction test of the number of processed emails predicted by accuracy, participant, and the interaction of accuracy and participant, showed an interaction effect of participant and accuracy predicting the number processed with Rule-based explanations (linear regression, p=0.0314, F[3,9]=2.515, R^2=0.456). As expected, there was also a main effect of participant predicting the performance speed in the case of Rule-based explanation paradigm (linear regression, p=0.0299, F[3,9]=2.515, R^2=0.456), which simply says that some participants were faster than others in this paradigm. This says that the number of emails processed depended, in the Rule-based paradigm, on the participant and the accuracy rate he/she experienced, in a negative direction: the higher the accuracy rate for some participants, the *lower* the number of emails processed.

Discussion: Accuracy.

The odd relationship between accuracy and participant efficiency suggests the possibility of a point of diminishing return. That is, the more accurate the algorithms (and their explanations), the harder it may be for a user to spot the flaw when the algorithm makes a mistake. Thus, there may be a threshold accuracy point beyond which users may view the cost of guiding the algorithm further as exceeding the benefit of doing so.

RESULTS (PART 2): THE USERS EXPLAIN BACK

To what extent could our participants' feedback be assimilated by machine learning algorithms? For all three paradigms, we coded participants' feedback (Negative Comment and Suggest Change) along two dimensions. The rows of Table 4 identify the type of change and the columns identify the knowledge needed to handle the change.

The types of feedback were not independent of the explanation paradigms: some paradigms seemed to encourage participants to think along the lines of particular types of feedback. We will point out these influences along the way whenever a paradigm represents at least 50% of a category.

Participants' Suggestions by Type

Type code 1: Adjust weight or feature importance.

Participants' reactions to Keyword-based explanations generated 69% of the feedback of this type, perhaps because of the feature-focused nature of the Keyword-based paradigm. Some participants' suggestions for changing feature weight or importance involved adjusting the weight on features in a general sense, such as:

> *P8 (on Keyword-based)*: "The second set of words should be given more importance."

Other participants flipped a weight from negative to positive (or vice versa), or focused on the frequency of the word occurrence in the email, akin to term weighting in information retrieval:

> *P7 (on Keyword-based)*: "Keyword 'include' is not good for the second set. It should be in the first set of words."
> *P1 (on Rule-based)*: "'Bankruptcy' is here over and over again, and that seems to be the obvious word to have in here."

Other comments concerned relative importance:

> *P4 (on Keyword-based)*: "I think that 'payroll' should come before 'year'."

These suggestions could easily be incorporated into existing machine learning algorithms. A general capability could be added for accepting this type of user constraint (e.g., "there should be a positive weight on feature *X*") (c.f. [1]). From the constraints, the corresponding weights could be automatically modified in the Naïve Bayes classifier, or the appropriate scoring functions for feature selection and pruning could be automatically modified in Ripper.

	KB-English	KB-common-sense	KB-domain	KB-other	Total	%
1. Adjust weight	11	11	4	13	39	12%
2. Select different features (words)	70	64	25	16	175	53%
3. Parse or extract in a different way	7	17	10	0	34	10%
4. Employ feature combinations	9	5	2	1	17	5%
5. Relational features	0	9	5	0	14	4%
6. Other	3	12	4	33	52	16%
Total	100	118	50	63	331	
%	30%	36%	15%	19%		

Table 4: Types of participants' changes (in rows) that required various background knowledge (in columns).

Type code 2: Select different features.

This was the most widespread type of feedback, for all three explanation paradigms. More than half of all feedback referred to either adding a new feature for the algorithm to consider or removing a feature from consideration:

> *P13 (on Rule-based)*: "It should put email in 'Enron News' if it has the keywords 'changes' and 'policy'. I put down some keywords that I noticed."

As with feature weights, it would be fairly easy to incorporate feature addition/removal into Naïve Bayes and Ripper given a user constraint entry capability. Using this, feature removal can be accomplished by deleting the feature from the training data. Feature addition is more challenging. Ripper's scoring functions for feature selection and pruning could be modified to prefer the recommended features; Naïve Bayes could be given a Bayesian prior distribution that preferred the recommended features.

Type code 3: Parse/extract features in a different way.

Some participants suggested a different form of text parsing, such as:

> *P1 (on Similarity-based):* "Different forms of the same word must be looked at."

In the simplest case, this could be achieved by an improved stemming procedure. In some cases, however, the suggested extraction operates on a structure such as a URL:

> *P6 (on Similarity-based):* "I think it would be good to recognize a URL."

Either the system would already need to know about URLs or else the user would need to define them (perhaps by giving examples).

Participants also suggested using the structure of the email to extract features, such as the "From" and "Subject" field:

> *P13 (on Rule-based):* "Yea, I mean it has 'job' in the subject line (for sorting into Resumé folder)"

Finally, some participants suggested new kinds of informative cues:

> *P6 (on Keyword-based):* "I think that it should look for typos in the punctuation for indicators toward Personal."

For any individual suggestion of this kind, it is easy to imagine manually engineering a method for assimilating this feedback (run a spelling checker and count misspelled words), but hard to imagine a general-purpose mechanism.

Type code 4: Feature combinations.

Participants pointed out that two or more features taken together could improve the prediction, especially when they were working with Similarity-based explanations, which generated 63% of the suggestions of this type:

> *P6 (on Similarity-based):* "I think it would be better if it recognized a last and a first name together."
> *P12 (on Keyword-based):* "I would think like 'authorize signature' or 'w-2 form'."

It would be easy to include specific combinations as new features in the learning algorithms. However, defining abstract sets (e.g., first and last names) would be more challenging in general.

Type code 5: Relational features.

An interesting type of participant suggestion concerned the use of relations, such as:

> *P6 (on Rule-based):* "I think maybe it should use the response and automatically put it in the folder with the message that was responded to."
> *P8 (on Keyword-based):* "This message should be in 'Enron-News' since it is from the chairman of the company."

In these cases, the participants used relationships between messages (threading) or organizational roles (chairman of Enron) to define a new feature. Incorporating relational features typically requires major changes to the architecture of the learning system, because standard email classification methods only consider a single email message at a time and only consider the contents of that email (i.e., as opposed to job title information in the address book). Some recent work has incorporated relational features [9, 15], but only in special-purpose implementations. Building a learning system that can be easily extended to incorporate new relations is an open research problem.

Type code 6: Other.

Most of the remaining feedback concerned changes to the learning algorithm itself. These included suggestions such as adding logical NOT to the rule language, eliminating the default rule in Ripper, requiring an equal number of positive and negative keywords in Keyword-based explanations, and so on. These changes would all require fundamental algorithm changes, and hence, cannot be automatically assimilated by existing learning algorithms.

There were also cases in which the real problem lay with the way the explanation was constructed, rather than with the learning algorithm:

> *P13 (on Similarity-based):* "Having 'points' being the only keyword, I mean that kind of sucks."

The true classifier used all of the keywords in the messages, but the explanation only highlighted the shared words with the top weights. This suggests that an automated method for assimilating user feedback would need a component that could diagnose whether the perceived problem is due to approximations in the explanation or design decisions in the learning algorithm.

Participants' Suggestions by Knowledge Source

Knowledge Code KB-English.

Almost a third (30%) of the participations' suggestions relied on knowledge of English:

> *P8 (on Rule-based):* "Does the computer know the difference between 'resumé' and 'resume'? You might have email where you're talking about 'resume' but not in a job-hiring sense."
> *P5 (on Keyword-based):* "Last names would be better indicators."

We coded feedback in this category if the necessary knowledge could be learned from analysis of large document collections or obtained from other online resources (e.g., Wordnet [17] or named-entity recognizers [27]).

Knowledge Code KB-Commonsense.

Some knowledge might need to be manually encoded, but it could then be reused across many different organizations and applications. For example, participants indicated that there are "families" of words that are work- or business-related, and also suggested topic-related words:

> *P4 (on Rule-based):* "'Policy' would probably be a good word that would be used a lot during business talk."
> *P1 (on Keyword-based):* "'Qualifications' would seem like a really good Resume word, I wonder why that's not down here."

Recent advances in topic modeling [3] might allow this kind of knowledge to be automatically discovered.

Knowledge Code KB-Domain.

Some participant suggestions relied on knowledge specific to Enron:

> *P11 (on Similarity-based):* "Different words could have been found in common like 'Agreement', 'Ken Lay'."

In this case, the machine leaning system would need to know that Ken Lay was the CEO of Enron, and that as such he carries several types of special importance to Enron employees, the implications of being an Enron employee, and so on. Such knowledge would need to be encoded separately for each organization, and therefore would be difficult to incorporate in a general machine learning approach.

Knowledge Code KB-Other.

All remaining feedback was coded KB-Other. This usually occurred when a participant's comment was not specific enough to suggest the underlying knowledge source.

Discussion: Incorporating the Suggestions

These two coding dimensions allowed us to estimate what fraction of the participant suggestions could be incorporated into Ripper or Naïve Bayes without significant algorithm changes. Specifically, type codes 1, 2, and 4 (weights, features, and feature combinations) supported by KB-English are the most promising for direct assimilation in these algorithms. These accounted for 27% of the 331 suggestions. Slightly more challenging would be these same change types but supported by KB-Commonsense, because of the difficulty of obtaining the relevant common sense topic models. These accounted for an additional 24% of the suggestions. The remaining suggestions appear to require substantial additional machine learning research in feature extraction, relational learning, and user interfaces for providing application-specific feature specifications.

CONCLUDING REMARKS

We have reported two classes of results. From a user perspective, some of these results were:

Human error: Participants were more accurate than the machine, but they were not perfect. This points out the likelihood of users introducing errors into the data.

Understanding: Rule-based explanations were the most understandable. Keyword-based were next, but the negative keywords list interfered. Similarity-based had serious understandability problems.

Preference: Some factors winning participant approval were reasoning soundness, clear communication of reasoning, and informal wording ("Wow!").

From an algorithm perspective, some results were:

Types of improvements: Among the suggestions were reweighting features, feature combinations, relational features, and even wholesale changes to the algorithms.

Assimilation difficulty: Roughly half of the suggestions for improvement appear to be amenable to automated assimilation with existing methods.

Open questions for machine learning: The results open new questions for research on methods for assimilating complex user suggestions for feature extraction, relational features, and incorporating constraints on solutions found by learning algorithms.

These results provide evidence that machine learning systems can explain their reasoning and behavior to users, and that users in turn can provide rich, informative feedback to the learning system. This suggests rich user-machine collaboration as a promising direction for intelligent user interfaces to learn more effectively, by better harnessing of the intelligence of users.

ACKNOWLEDGMENTS

We thank the participants of our study. We also thank Kandace Bangert for her assistance. This work was supported by Intel, by NSF IIS-0133994, and by DARPA grant HR0011-04-1-0005, contract NBCHD030010.

REFERENCES

1. Altendorf, E., Restificar, E., and Dietterich, T. Learning from sparse data by exploiting monotonicity constraints. *Conf. Uncertainty in Artificial Intelligence*, 2005.
2. Billsus, D., Hilbert, D. and Maynes-Aminzade, D. Improving proactive information systems. *ACM IUI 2005*, 159-166.
3. Blei, D., Ng, A. and Jordan, M. Latent Dirichlet allocation. *J. Machine Learning Res.*, 3, 2003, 993-1022.
4. Blythe, J. Task learning by instruction in Tailor. *ACM IUI 2005*, 191-198.
5. Brutlag, J., Meek, C. Challenges of the email domain for text classification. *Intl. Conf. Machine Learning 2000*, 103-110.
6. Chklovski, T., Ratnakar, V. and Gill, Y. User interfaces with semi-formal representations: A study of designing argumentation structures. *ACM IUI 2005*, 130-136.
7. Cohen, W. Learning rules that classify e-mail. *AAAI Spring Symp. Information Access*, *1996*.
8. Dalvi, N., Domingos, P., Sanghai, M. S. and Verma, D. Adversarial classification. *ACM Intl. Conf. Knowledge Discovery and Data Mining, 2004*, 99-108.
9. Getoor, L., Friedman, N., Koller, D. and Pfeffer, A. Learning probabilistic relational models. S. Dzeroski and N. Lavrac (eds.) *Relational Data Mining*. Springer-Verlag, 2001.
10. Hart, S. and Staveland, L. Development of a NASA-TLX (Task load index): Results of empirical and theoretical research. P. Hancock and N. Meshkati (eds.)*, Human Mental Workload, 1988*, 139-183.
11. Herlocker, J., Konstan, J. and Riedl, J. Explaining collaborative filtering recommendations. *ACM CSCW 2000*, 241-250.
12. Kissinger, C., Burnett, M., Stumpf, S., Subrahmaniyan, N., Beckwith, L., Yang, S. and Rosson, M. B. Supporting end-user debugging: What do users want to know? *ACM Advanced Visual Interfaces 2006*, 135-142.
13. Klimt, B. and Yang, Y. The Enron corpus: A new dataset for email classification research. *European Conf. Machine Learning 2004*, 217-226.
14. Lieberman, H. and Kumar, A. Providing expert advice by analogy for on-line help. *IEEE/WIC/ACM Intl. Conf. Intelligent Agent Technology 2005, 26-32*.
15. Marx, Z., Rosenstein, M. T., Kaelbling, L. P., Dietterich, T. G. Transfer learning with an ensemble of background tasks. *NIPS 2005 Workshop on Transfer Learning*.
16. McCarthy, K., Reilly, J., McGinty, L. and Smyth, B. Experiments in dynamic critiquing. *ACM IUI 2005*, 175-182.
17. Miller, G. WordNet: A lexical database for English. *Comm. ACM* 38(11), 1995, 39-41.
18. Myers, B., Weitzman, D., Ko, A. Chau, D. Answering why and why not questions in user interfaces. *ACM CHI 2006*.
19. Oblinger, D., Castelli, V. and Bergman, L. Augmentation-based learning. *ACM IUI 2006,* 202-209.
20. Pazzani, M. J. Representation of electronic mail filtering profiles: a user study. *ACM IUI 2000*, 202-206.
21. Phalgune, A., Kissinger, C., Burnett, M., Cook, C., Beckwith, L. Ruthruff, J. Garbage in, garbage out? An empirical look at oracle mistakes by end-user programmers, *IEEE Symp. Visual Languages and Human Centric Computing 2005*, 45-52.
22. Porter, M. An algorithm for suffix stripping. *Program*, 14(3), 1980, 130-137.
23. Pu, P. and Chen, L. Trust building with explanation interfaces. *ACM IUI 2006*, 93-100.
24. Rettig, M. Prototyping for tiny fingers. *Comm. ACM* 37(4), 1994, 21-27.
25. Shen, J., Li, L., Dietterich, T. Herlocker, J. A hybrid learning system for recognizing user tasks from desk activities and email messages**.** *ACM IUI 2006*, 86-92.
26. Witten, I., Frank, E. *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd Ed., Morgan Kaufmann, 2005.
27. Zhou, G., Su, J. Named entity recognition using an HMM-based chunk tagger. *ACM Assoc. Comp. Linguistics 2002*.