RC25243 (W1112-008) December 6, 2011
Computer Science

IBM Research Report

EPIC: A Multi-Tiered Approach to Enterprise Email Prioritization

Jie Lu, Zhen Wen, Shimei Pan, Jennifer Lai
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 704
Yorktown Heights, NY 10598

Research Division
Almaden - Austin - Beijing - Cambridge - Haifa - India - T. J. Watson - Tokyo - Zurich

EPIC: A Multi-Tiered Approach to Enterprise Email Prioritization

Jie Lu Zhen Wen Shimei Pan Jennifer Lai
IBM T. J. Watson Research Center
19 Skyline Drive, Hawthorne, NY 10532 USA
{jielu, zhenwen, shimei, jlai}@us.ibm.com

ABSTRACT
We present Enterprise Priority Inbox Classifier (EPIC), an automatic personalized email prioritization system based on a topic-based user model built from the user's email data and relevant enterprise information. The user model encodes the user's topics of interest and email processing behaviors (e.g. read/reply/file) at the granularity of pair-wise interactions between the user and each of his/her email contacts. Given a new message, the user model is used in combination with the message metadata and content to determine the values of a set of contextual features. Contextual features include people-centric features representing information about the user's interaction history and relationship with the email sender, as well as message-centric features focusing on the properties of the message itself. Based on these feature values, EPIC uses a dynamic strategy to combine a global priority classifier with a user-specific classifier for determining the message's priority. An evaluation of EPIC based on 2,064 annotated email messages from 11 users, using 10-fold cross-validation, showed that the system achieves an average accuracy of 81.3%. The user-specific classifier contributed an improvement of 11.5%. Lastly we report on findings regarding the relative value of different contextual features for email prioritization.

Author Keywords
Email, prioritization, priority inbox, enterprise, user modeling.

ACM Classification Keywords
H.5.2 [Information Interfaces and Presentation]: User Interfaces—User interaction styles; H.4.1 [Information Systems Applications]: Office Automation—Time management.

General Terms
Algorithms, Design, Experimentation.

1. INTRODUCTION

The Radicati Group [14] reports that in 2011 the average number of messages sent and received by a corporate knowledge worker is close to 112 messages per day, and that this number is only growing. Given such a large number of messages and the amount of time required to read and respond to each one, people often seek to optimize the time spent on email processing by scanning the inbox, checking sender names and subjects in order to prioritize some messages for attention over others [15, 17, 19]. When the number of new messages in one's inbox is large, sifting through them to identify high-priority ones quickly becomes a non-trivial and time-consuming task by itself, resulting in a daily feeling of "email overload" [3] and occasionally resulting in the unfortunate result of overlooking key messages since people find it difficult to create an efficient order when sorting based on elements such as sender, subject, or date [7].

To help alleviate this problem, we developed Enterprise Prioritize Inbox Classifier (EPIC), a system that automatically identifies high-priority emails in a user's inbox and displays them in a separate section from other emails in the interface, with the goal of assisting the user in his/her daily triage of incoming messages. The prioritization is informed by a user model that is automatically created from the user's email data along with relevant enterprise information (e.g. organizational structure). Information encoded in this user model is used in conjunction with the message metadata and content to compute the values of a set of contextual features. Based on these contextual features, the system then determines the priority of the incoming message using a multi-tiered approach, which dynamically determines how to combine a global priority classifier (established from labeled training data of multiple users) with a user-specific classifier built from ongoing feedback training.

EPIC differs from previous work on email prioritization in two primary aspects. First, we derive a set of contextual features from each message based on a topic-based user model. This user model encodes granular information about the user's interaction with different people on different topics, and the user's relationship (e.g. direct-report, team member) with each of his/her email contacts in an enterprise environment. Second, we have implemented a multi-tiered approach to priority classification. Instead of simply combining a global classifier and a user-specific classifier with fixed weights (e.g. [1]), our system uses instance-based matching between a new message and messages for which the user has provided one-click feedback to dynamically determine the best strategy to combine the two classifiers. This approach enables the system to quickly incorporate user-specific criteria for

determining email priority without sacrificing the reliability provided by a global classifier, which has been created using pooled training data from multiple users.

In the balance of the paper, we first present related work in Section 2. Then we provide an overview of EPIC in Section 3, including its system architecture and interface. Sections 4, 5 and 6 describe in detail the user model, contextual features, and the prioritization process respectively. Finally we describe the evaluation in Section 7, followed by a discussion in Section 8, and conclude the paper in Section 9.

2. RELATED WORK

Prior studies show that the action a user takes on an email message (e.g. read, reply, file, delete) largely depends on the user-perceived importance of this message [4, 5, 6, 19]. The main goal of email prioritization is thus to identify email messages with a high value of user-perceived importance. The development of EPIC's contextual features is informed by previous study findings regarding which characteristics of an email message affect its perceived importance. Particularly, Dabbish et al. collected data about individuals' actions with their emails through surveys [4, 5] and a think-aloud study [19], and conducted statistical analyses to understand the associations between message features and the actions the recipients take on a message. Their study results indicated that the user's relationship with the email sender, the history of communication between them, and the type of message content (e.g. action or information request, information or attachment, status update, scheduling, reminder, social) were the main characteristics that influenced users' perceptions of message importance. Based on a series of interviews and a survey, Neustaedter et al. found that social information was vital for determining the importance of an email [13]. For example, the user's personal or working relationship with the sender and recency of contact were good indicators of email importance.

There have been several works on redesigning email interfaces to help users quickly identify important emails in their inbox. For example, the REMAIL interface [10], the conversation-based email client described in [18], and Google Gmail automatically group emails into conversational threads. DriftCatcher displays at the interface the social metadata associated with each email, including relationship tie strength between the sender and the user, the average time taken by the user to respond to the sender, and the social context classification of the email content (e.g. Inquire, KeepInTouch, Interest, Planning, Inform/Share) [11]. The SNARF prototype email tool allows users to sort emails based on social metrics that measure the social importance of the user's email correspondents [13]. In comparison, EPIC automatically groups emails into high-priority vs. normal categories displayed at the interface, and incorporates threading and social information into the contextual features it uses for determining email priority.

Because EPIC automatically prioritizes email messages in the user's inbox, it is closely related to prior work on automatic email prioritization. Existing approaches mostly prioritize emails based on the classifier learned using supervised learning algorithms [1, 8, 20]. For example, Google Gmail Priority Inbox uses linear logistic regression models with a variety of social, content, thread, and label features to prioritize users' incoming messages [1]. The PRIORITIES system uses a Support Vector Machine (SVM) classifier over word-based, phrase-based, and meta-level features (e.g. message sender, recipients, length, time, presence of attachments) to determine the importance of new unread emails [8]. The work of Yang et al. also uses SVM classifiers, but with additional social importance features computed based on each user's personal social network derived from email data [20]. The content-based features used by these approaches for classifier learning are simply words that occur in email content, which may not work well for very brief messages with too few words (sparse data) or long messages with too many words (noisy data). By contrast, EPIC includes high-level features such as the importance of the topic discussed in email content (inferred based on a topic-based user model) to increase the robustness of prioritization. Regarding user feedback, existing approaches either only learn the classifiers through one-time batch processing of labeled training data and do not consider dynamic user feedback [8, 20], or simply use user feedback to incrementally update the feature weights of the existing classifiers [1]. Aggressively updating feature weights based on user feedback reduces the robustness of email prioritization, while conservatively updating feature weights results in a slow response to user feedback. In comparison, EPIC dynamically combines a global classifier learned from training examples that are aggregated across multiple users with a user-specific classifier created based on ongoing user feedback to achieve a balance between robustness and responsiveness.

3. SYSTEM OVERVIEW

In this section we provide an overview of Enterprise Priority Inbox Classifier, first discussing its system architecture, and then its user interface.

3.1 System Architecture

Figure 1 illustrates the system architecture, which consists of five main modules. The *user modeling* module creates a model for each user based on his/her email data and relevant enterprise organizational information. The user model encodes information such as the user's interaction behaviors with his/her contacts through emails, what topics they discuss, and the type and strength of their relationship within the enterprise. Section 4 describes the user model in more detail.

The *feature extraction* module takes an email message and the user model as input, and computes the values of a set of contextual features. These contextual features describe the context associated with the message, including interaction and relationship information associated with the email

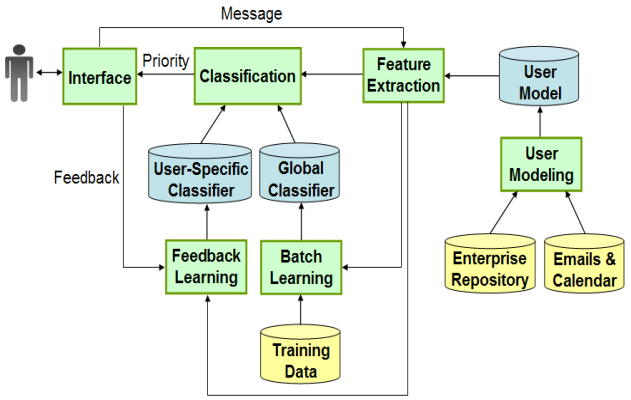

Figure 1. EPIC system architecture

sender (retrieved from the user model), and characteristics of the message that are deemed as influencing user-perceived message importance. Section 5 provides a list of all contextual features.

The *batch learning* module creates a global priority classifier using supervised learning based on labeled messages from multiple users (Section 6.1), and the *feedback learning* module collects user feedback from the interface about the priority of individual messages, and analyzes these messages to create a user-specific classifier (Section 6.2). Both of these modules call the feature extraction module to extract contextual features from the messages that are used for learning.

Finally, the *classification* module determines the priority of a message based on its contextual features. It combines the result from applying the global classifier and that from the user-specific classifier to classify each message as either high-priority or not (we do a binary classification). The description of the algorithm can be found in Section 6.3.

3.2 User Interface

Figure 2 shows the interface of EPIC developed as an extension to an IBM Lotus Notes 8 email client. Lotus Notes 8 supports two types of inbox view: a sort-ordered view with messages sorted by sender name, subject, date, or size, and a grouped view in which messages are automatically organized into different categories ("High Importance" for messages marked highly important by senders, "Calendar Events" for messages related to calendar activities, "Normal" for all other messages). All messages within each category are ordered by date.

EPIC extends this email interface by adding a new category of "High Priority" (Figure 2a) to display messages classified as high-priority by its automatic email prioritization. In addition, a "High Priority" icon is added to

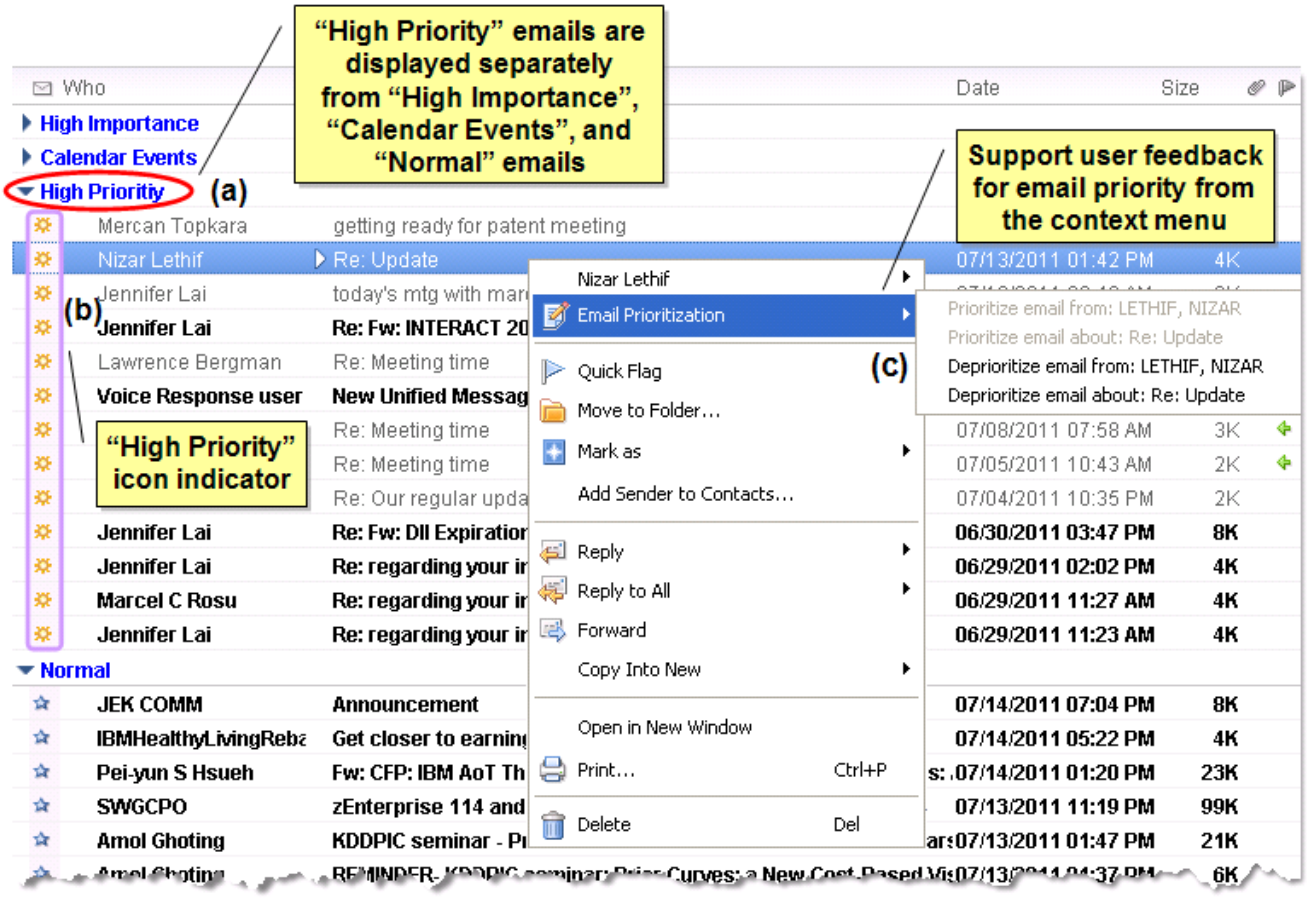

Figure 2. EPIC user interface

the display of each high-priority email in the inbox view (Figure 2b) so that even when a user chooses a sort-ordered view instead of a grouped view, s/he can still easily identify high-priority emails based on the icon indicator.

Having both a "High Importance" section and a "High Priority" section may seem redundant at first until one realizes that the messages in the "High Importance" category are marked as 'urgent' by the sender. This does not mean they are of high priority to the receiver. Quite to the contrary, these messages often linger unread once the receiver sees that they are from a support person about a calendar event that is still weeks away, or from other administrative staff that want forms completed or updated.

To support user feedback, an email prioritization menu item was added to the context menu triggered when a user right mouse buttons down on a message highlighted in the inbox view. The user can indicate to the system to de-prioritize a high-priority message or to prioritize a message currently in the "Normal" category while supplying the reason for such de-prioritization or prioritization (e.g. whether it is because of the sender or the subject of the message, see Figure 2c).

4. USER MODEL

EPIC creates a user model for each user. It encodes information about the user's interaction history with each email contact and their relationship, two characteristics that have been shown to be influential of the user's assessment of message importance [4, 5, 6, 19]. It extends the multi-tiered, topic-based user model proposed in [12] to record finer-grained information about the user's behaviors in his/her interactions with different people, and the user's relationships with them in an enterprise environment. Below we describe in detail *interaction* and *relationship*, two of the main data structures in EPIC's user model.

4.1 Interaction

The interaction between the user and an email sender consists of the set of messages exchanged between them (sent to and copied on), the statistical topic model generated from the aggregate content of this set [12], and the relevant statistics derived from these messages and associated user actions. Specifically, the following statistics are recorded in an interaction between the user and an email sender:

- *incoming_count*: the number of incoming messages from this person.
- *outgoing_count*: the number of outgoing messages to this person.
- *read_count*: the number of incoming messages from this person that have been read by the user.
- *reply_count*: the number of incoming messages from this person that have been replied by the user.
- *reply_lapse*: the average time taken by the user to reply an incoming message from this person.
- *file_count*: the number of incoming messages from this person that have been flagged or saved by the user.
- *most_recent_interaction_time*: the time of the most recent message exchanged between the user and this person.

4.2 Relationship

The relationship between the user and an email sender consists of one or more connections between the two. A connection is a particular type of link between two people. The connections between the user and his/her email contacts fall into four categories:

- *Communicational*: connections derived from senders and recipients of emails as well as participants of calendar meetings.
- *Organizational*: connections based on organizational structure (e.g. managing, managed, same manager).
- *Social*: connections derived from activities in enterprise online social networks, such as community co-membership, wiki co-editing, file sharing.
- *Academic*: connections as a result of academic activities such as paper/patent co-authoring.

Different connections are assigned different weights to reflect their inherently different strengths (e.g. organizational connections are typically stronger than social connections). The overall relationship strength between the user and an email contact is the weighted sum of all their connections.

5. CONTEXTUAL FEATURES

The contextual features used for prioritization are developed based on previous study findings as well as our own observations about the influential characteristics in determining message importance. They fall into two categories: people-centric and message-centric. Below we describe these features and how their values are calculated based on the user model as well as the message metadata and content.

5.1 People-Centric Features

People-centric features represent aggregate information about the user's interaction and relationship with the email sender, including:

- *incoming_freq*: the normalized frequency of incoming messages from this sender, which is calculated as $max(incoming\_count, T) / T$, where T is the interaction frequency threshold, empirically set to 50.
- *outgoing_freq*: the normalized frequency of outgoing messages to this sender, which is calculated as $max(outgoing\_count, T) / T$.
- *read_rate*: the percentage of incoming messages from this sender that have been read by the user (*read_count* divided by *incoming_count*).
- *reply_rate*: the percentage of incoming messages from this sender that have been replied by the user (*reply_count* divided by *incoming_count*).
- *reply_lapse*: *reply_lapse* from the user model, measured in days.

- *file_rate*: the percentage of incoming messages from this sender that have been flagged or saved by the user (*file_count* divided by *incoming_count*).
- *interaction_recency*: the recency of interaction, calculated as 1.0 / (*log*(*t* + 1.0) + 1.0), where *t* is the time lapse measured in days between current time and *most_recent_interaction_time* from the user model.
- *relationship_type*: the connection between the user and this sender that has the highest associated weight (Section 4.2).
- *relationship_strength*: the overall relationship strength between the user and this sender, which is calculated as the weighted sum of all their connections.

5.2 Message-Centric Features

Message-centric features focus on the properties of the message itself, which include:

- *message_scope*: whether the message is sent exclusively to the user, or to a small group of people, or to a large group.
- *message_type*: whether the message is a regular mail, or a meeting notice that requires user action (e.g. invite, reschedule), or a meeting notice that does not require user action (e.g. confirm), or something else (e.g. automatic message like out-of-office reply).
- *content_type*: if the message content contains a request (empirically determined based on lexical heuristics), time-critical words (e.g. deadline), keywords pre-specified by the user, and/or one or more file attachments.
- *threading*: if the message belongs to an email thread, if the user has taken any action on previous messages from the same thread.
- *topic_likelihood*: the likelihood that the message content is about the topic inferred by the system, which is calculated based on LDA's document-topic distribution as LDA [2] is used to derive the topics contained in the user model.
- *topic_importance*: the importance of the topic (from the user's perspective) inferred from the message content. Because the topics derived by LDA are not ranked, information about topic importance cannot be directly obtained from LDA. There has been prior work on inferring topic importance based on criteria such as topic coverage and variance, topic distinctiveness, topic mutual information, topic similarity and redundancy [16]. For the email application domain, our experience is that user actions associated with email messages provide a better indicator of user-perceived topic importance. Therefore, topic importance is computed using a weighted combination of the following factors: the percentage of the user's emails that are about this topic, the percentage of the emails about this topic that were read, and the percentage of the emails about this topic that were forwarded, replied, saved, or flagged.

6. PRIORITIZATION

6.1 Global Priority Classifier

EPIC uses linear regression (chosen for its efficiency and robustness) to create a global priority classifier based on labeled training messages collected from multiple users. The priority score s_g of an email message is a linear combination of the message's contextual features:

$$s_g = \sum_{i=1}^{n} a_i f_i$$

where f_i is the value of the message's i^{th} contextual feature as defined in Section 5, and a_i is the regression parameter representing the automatically learned weight for this feature.

Before learning, feature selection is performed to remove features that lack variations in the training data since such features do not have prediction power and will cause the regression to fail. Specifically, features with variance less than a threshold of 0.001 are filtered out. During learning, s_g is set to 1 for high-priority and −1 for low-priority.

To determine the priority of a message using this classifier, the system calculates the message's s_g value based on the values of its contextual features, and classifies it as high-priority if s_g is greater than a classification threshold t_c. The value of t_c can be determined based on application needs or user preferences. For example, if low false-negative rate is required or desired (e.g. users do not want to miss important emails) and high false-positive rate (unimportant emails mistakenly labeled as high-priority) can be tolerated, a smaller t_c value is used.

6.2 User-Specific Priority Classifier

A user-specific classifier is dynamically learned based on ongoing user feedback. Although implicit feedback derived from user actions such as reading or replying an email message may imply the priority of this message, it is often inaccurate and unreliable ([1, 5]). Therefore, currently EPIC focuses on using explicit user feedback to obtain more accurate and reliable ground truth data for creating user-specific classifiers.

When a user provides priority feedback for a message (e.g. through the email prioritization context menu item described in Section 3.2), the system records the priority label (e.g. high-priority or low-priority because of the sender and/or the subject) along with the contextual features of the message to create a user-specific feedback *instance*. When enough feedback instances are collected for this user, linear regression is used to learn a user-specific classifier:

$$s_u = \sum_{i=1}^{n} b_i f_i$$

The classifier is updated as the number of feedback instances grows.

6.3 Priority Classification

In theory, a user-specific classifier should perform better than a global classifier since the criteria used by different users for determining message priority are likely not the same. However, in practice, few users will provide sufficient amount of feedback needed to train a comprehensive user-specific classifier, especially during the initial period of using our system. Therefore, priority classification is based on a combination of the global classifier and the user-specific classifier. We propose three methods, described below.

Basic Linear Combination (BASIC)

This method linearly combines the priority scores from the global classifier and the user-specific classifier as:

$$s = w \times s_g + (1 - w) \times s_u$$

We experimented with different values of w to investigate how w affects prioritization performance. The results are reported in Section 7.2.

Dynamic Linear Combination with Instance Matching on Contextual Features (DYNAMIC+FEATURES)

Given a new message M, this method dynamically assesses the quality of the user-specific classifier to decide if it is a reliable choice for determining M's priority. The quality is estimated based on the matching between M and previous feedback instances which are used to train the user-specific classifier. Specifically, the method computes the shortest feature-based distance as:

$$d = \min_j(\sum_{i=1}^{n}(f_{M,i} - f'_{j,i}))$$

where $f_{M,i}$ is the value of M's i^{th} contextual feature as defined in Section 5, and $f'_{j,i}$ is the value of the i^{th} contextual feature in the j^{th} feedback instance.

If d is below a certain threshold (empirically set to 0.5), it indicates that previous feedback instances contain message(s) similar to M, which implies that the user criteria of determining the priority of a message similar to M have been encoded in the classifier.[1] As a result, the user-specific classifier is expected to perform well in predicting M's priority, so the system sets the weight of the global classifier to 0 in linear combination, essentially utilizing the user-specific classifier only. If d value is above the threshold, the basic linear combination of the two classifiers, as described in the first method, is used.

[1] The goal is to find at least one similar feedback instance, which is measured by the shortest distance between the current message M and previous feedback instances. Requiring M to be similar to at least X% of the feedback instances in order to use the user-specific classifier alone might improve the accuracy for the applicable cases, but there would be fewer applicable cases when compared to requiring at least one similar feedback instance.

Dynamic Linear Combination with Instance Matching on Sender/Subject (DYNAMIC+SENDER/SUBJECT)

Wainer et al. investigated which inbox-level cues are used to select which email to open, and reported in [19] that email priority is a function of the predicted utility of message content inferred from sender and subject. Given this finding and our observation that it is difficult for users to articulate prioritization criteria along the dimensions of contextual features, we decided to solicit user feedback only on sender and subject. This design avoids a complex interface that might confuse users or even discourage them from providing feedback, but still enables the system to collect valuable feedback utilized by this priority classification method. If a new message matches one previous feedback instance on the main factor(s) indicated by the user (exact match for sender and substring match for subject), the message is given the same priority as this feedback instance. If it matches multiple feedback instances, then the message is assigned the priority of the most recent feedback instance. Otherwise, the basic linear combination method is used to determine the message's priority.

7. EVALUATION

In this section, we first describe our evaluation methodology, followed by an analysis of the evaluation results.

7.1 Methodology

We recruited eleven people within a large organization for the evaluation, including one manager, three product software engineers, two research software engineers, and five researchers. All of them use emails extensively for communication with others. A user model was created for each study participant based on the messages from his/her inbox, sent folder, and any user-created folders s/he selected. Each participant was then asked to annotate the priority for a set of messages randomly selected from his/her previously received messages. There were six labels to choose from:

- high-priority because of sender,
- high-priority because of subject,
- high-priority because of both (sender and subject),
- low-priority because of sender,
- low-priority because of subject,
- low-priority because of both.

A total number of 2,064 labeled messages were collected, for which each participant contributed a minimum of 100

	# High-priority	# Low-priority
Because of sender	257 (22.80%)	193 (20.60%)
Because of subject	209 (18.55%)	410 (43.76%)
Because of both	661 (58.65%)	334 (35.64%)
Total	1,127	937

Table 1. Summary statistics of the labeled messages used for evaluation

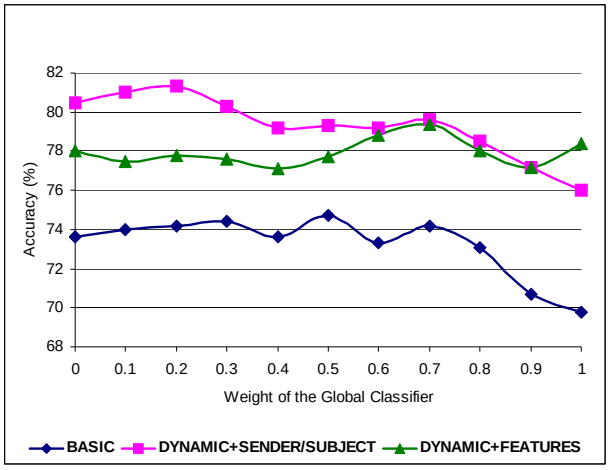

Figure 3. Accuracy results of three priority classification methods in different weight settings

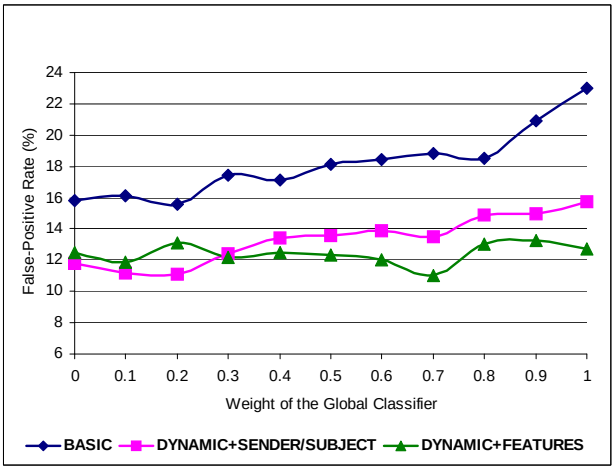

Figure 4. False-positive rates of three priority classification methods in different weight settings

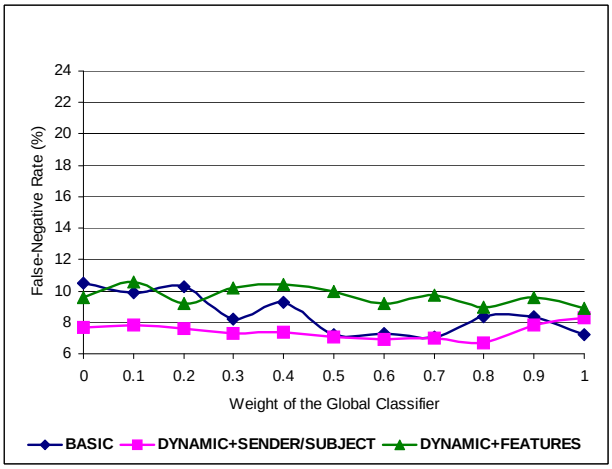

Figure 5. False-negative rates of three priority classification methods in different weight settings

and a maximum of 303 messages. In comparison, the average number of messages a user received was 5,050. Table 1 shows summary statistics of these messages.

We compared the three priority classification methods described in Section 6.3. To determine message priority based on the priority score, the default classification threshold (see Section 6.1) was 0 (a positive score means high-priority). We conducted 10-fold cross-validation to randomly partition the data into 10 equal folds with 9 folds used for training and the remaining fold for testing (repeated 10 times with a different fold for testing each time). In the next section we report the average performance (over all users and folds) for the following measurements: *accuracy* (percentage of messages that were correctly classified); *false-positive rate* (percentage of low-priority messages that were classified as high-priority); and *false-negative rate* (percentage of high-priority messages that were classified as low-priority).

7.2 Results

Figure 3 shows the accuracy results for the three methods when different weights were used to combine the global classifier and the user-specific classifier. For the BASIC method, its highest accuracy (74.7%) was achieved when the two classifiers were combined with equal weights (w = 0.5). Using the global classifier alone (w = 1) resulted in the lowest accuracy, which indicates that different participants did not use the same criteria for determining email priority. Using the user-specific classifier alone (w = 0) had better performance than using the global classifier only, demonstrating the importance of personalization in email prioritization. However, using the user-specific classifier alone did not yield the best accuracy, which suggests that the quality of the user-specific classifier for some participants might not be high due to insufficient training data.

For DYNAMIC+FEATURES, its best accuracy (79.4%) was produced when the global classifier was given a higher weight (w = 0.7). This was because when the system decided to use a combination of the two classifiers instead of the user-specific classifier alone (for ~60% of the test data in our experiments), it was for the cases when the user-specific classifier did not perform well by itself, thus relying more on the global classifier yielded better accuracy. This result demonstrates that the instance matching-based quality assessment of the user-specific classifier worked reasonably well.

For DYNAMIC+SENDER/SUBJECT, a lower weight for the global classifier (w = 0.2) yielded the highest accuracy (81.3%). This optimal weight (which favors the user-specific classifier) is different from the optimal weight (0.7) for DYNAMIC+FEATURES. The reason was because only a small portion (~10%) of the test data had matching sender/subject in the training data. Among the rest of the test data, the user-specific classifier performed well in a lot of cases. If the global classifier were given a higher weight,

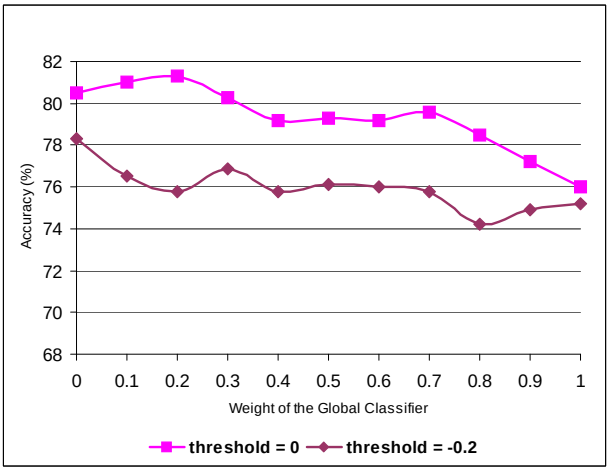

Figure 6. Accuracy results of DYNAMIC+SENDER/SUBJECT with different classification threshold values

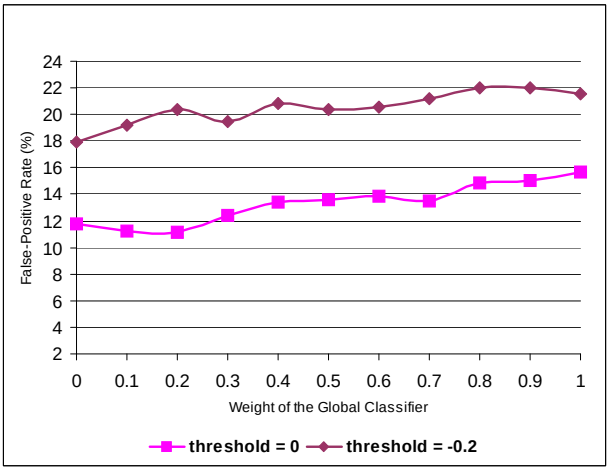

Figure 7. False-positive rates of DYNAMIC+SENDER/SUBJECT with different classification threshold values

it would decrease the accuracy for these cases, resulting in an overall worse performance.

Among all three methods, the two DYNAMIC methods consistently outperformed the BASIC method in all weight settings. DYNAMIC+SENDER/SUBJECT worked better than DYNAMIC+FEATURES in most settings, which demonstrates that the additional information gathered in priority feedback from users (i.e. because of the sender, the subject, or both) helped improve prioritization performance.

Figure 4 and Figure 5 depict the results of false-positive rate and false-negative rate respectively for different methods. From the curve of the BASIC method in Figure 4, we can see that using the global classifier alone (w = 1) yielded a very high false-positive rate (23.0%). This was related to the fact that some participants labeled a much smaller percentage of their messages as high-priority compared to the average (for example, the percentage of high-priority messages from one participant was 27% while the average was 55%). The discrepancy implies that different participants had different standards when determining high vs. low priority, some "stricter" than others. As a result, while an email was considered low-priority by one user, similar emails were regarded as high-priority by other users. Not surprisingly, using the global classifier created based on the training data from all participants produced high false-positive rate for those with "stricter" standards.

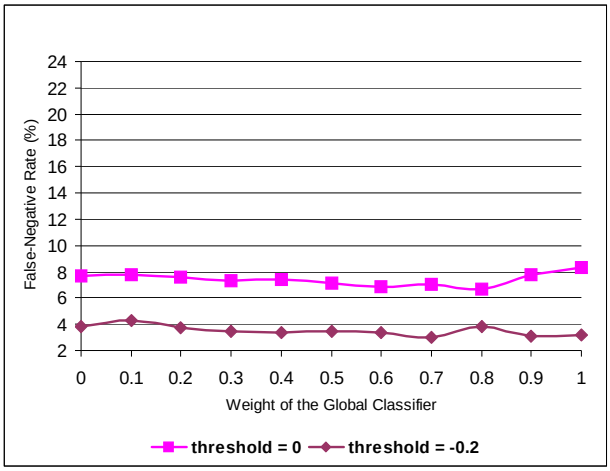

Figure 8. False-negative rates of DYNAMIC+SENDER/SUBJECT with different classification threshold values

For all the methods, the values of false-negative rate were consistently lower than those of false-positive rate, indicating that the classification threshold value of 0 is a reasonable choice for the (common) situations where low false-negative rate is preferable to low false-positive rate. Both DYNAMIC methods had lower false-positive rate than BASIC, but DYNAMIC+FEATURES did not perform well measured in false-negative rate. On the other hand, DYNAMIC+SENDER/SUBJECT consistently performed well for both false-positive rate and false-negative rate. DYNAMIC+SENDER/SUBJECT was also the least sensitive to different weight settings, especially for false-negative rate.

To examine how a different classification threshold value would affect these rates, we also tested the threshold value of –0.2 for DYMAMIC+SENDER/SUBJECT, the best performer among the three methods. Figures 6, 7 and 8 compare the results of using 0 vs. –0.2 for the classification threshold. The smaller threshold value reduced false-negative rate to less than 4%, at the cost of increased false-positive rate and reduced overall accuracy.

To investigate the relative value of different contextual features for email prioritization, we compared the weights of different features in the learned (global and user-specific) classifiers. A large absolute weight value for a feature indicates the predicting power of this feature in determine email priority. A feature that has a small variance in its weights across multiple user-specific classifiers indicates its

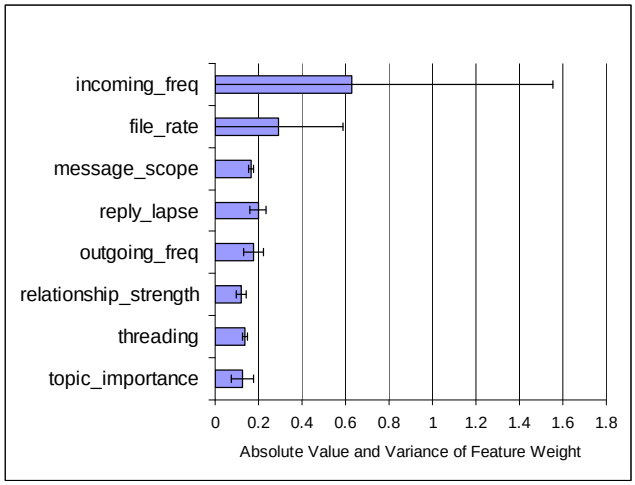

Figure 9. Most important contextual features for email prioritization.

robustness in determining email priority. Based on these two criteria (large absolute value and small variance), we selected eight "most important" contextual features, as shown in Figure 9. We can see from the figure that they include both people-centric features (*incoming_freq*, *outgoing_freq*, *relationship_strength*, *reply_lapse*, and *file_rate*) and message-centric features (*message_scope*, *threading*, and *topic_importance*). Feature *incoming_freq* had the highest mean absolute weight, indicating its high predicting power in general for email prioritization. However, it also had a big variance in its weights for different user-specific classifiers, which means that its value in determining email priority was highly user-dependent. In contrast, feature *outgoing-freq*, which is similar to *incoming_freq* in nature, not only provided good predicting power (large mean absolute weight), but also was stable across different users (small variance).

8. DISCUSSION

One of the main goals of our evaluation was to understand the value added by the user-specific classifier and the dynamic strategy for combining classifiers. As a result, we compared the performance of using either type of priority classifier (global or user-specific) alone (considered a baseline) with those of using different methods to combine the global classifier and the user-specific classifier. The evaluation results indicate that using either type of classifier alone does not achieve as high accuracy for email prioritization as using a combination of both. Using a dynamic strategy to combine a global classifier with a user-specific classifier improves performance in specific, but common, cases. In cases when the user feedback is insufficient for creating a reliable user-specific classifier, or when the context covered by the existing feedback is not applicable to the new message, the global classifier is utilized to augment the system's robustness. In cases when the user feedback is highly relevant to the context of the new message, the system is able to directly apply the user-specific criteria it has learned from such feedback to provide better personalized prioritization.

One of our design goals was to favor false positives over false negatives since there is less potential harm in presenting users one message too many to review than causing them to miss a key message. Currently this is achieved by adjusting the classification threshold value manually. We envision that longer term the classification threshold values can be dynamically adjusted based on users' behavior and feedback, thus adapting the system to suit a user's specific needs and requirements. In addition, there may be other ways to reduce false positives without negatively affecting false negatives. For example, the user can label certain messages as "can't miss", and the system can create a model to encode the characteristics of these messages and use that information in combination with the classifiers to determine email priority.

Currently EPIC only uses explicit user feedback because implicit feedback is noisy and unreliable. A user may only preview a message s/he considers high-priority instead of opening to read it. User actions in response to an email may be taken in other communication channels (e.g. phone call, face-to-face meeting, instant message, etc.). Previous studies on enterprise email communication discovered that only a small portion of the incoming messages (e.g. ~14%) were replied [9]. It has also been reported in prior work that users read mail they acknowledge is not important ([1]), and the influence of user-perceived importance is small in the decision to reply a message ([5]). However, we understand that users don't always provide much explicit feedback and implicit feedback can provide valuable information which can be utilized by the system to improve its prioritization. For example, the system can learn from user behavior such as skipping certain messages in the "High Priority" category, while taking immediate action on messages in the "Normal" category. The challenge is to identify the set of user behaviors that could be considered reliable indicators of message priority since these can include a single specific action, a combination of multiple types of actions, or even action patterns.

9. CONCLUSION

In this paper, we present Enterprise Priority Inbox Classifier, a system that automatically prioritizes a user's incoming email messages in his/her inbox using a topic-based user model and a multi-tiered approach to priority classification. It uses a variety of contextual features to determine email priority, whose values are computed based on the message metadata, content and a fine-grained user model. The user model encodes information about the user's interaction history and relationship with each of his/her email contacts and the topics they discussed. It is dynamically built from the user's previous emails as well as other relevant data (e.g. organizational structure). To determine the priority of a message based on the feature values, the system dynamically combines a global classifier created using labeled training data from multiple users and

a user-specific classifier built for each user based on his/her ongoing feedback. The global classifier helps alleviate the cold start problem and improve the robustness of priority prediction, while the user-specific classifier increases the system's adaptability and enables quick response to user feedback. The evaluation results showed the effectiveness of our approach for personalized email prioritization.

REFERENCES

1. Aberdeen, D., Pacovsky, O., & Slater, A. The learning behind Gmail Priority Inbox. In *NIPS 2010 Workshop on Learning on Cores, Clusters and Clouds*.
2. Blei, D., Ng, A., & Jordan, M. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.
3. Dabbish, L., & Kraut, R. Email overload at work: an analysis of factors associated with email strain. In *CSCW 2006*.
4. Dabbish, L., Kraut, R., Fussell, S., & Kiesler, S. To reply or not to reply: predicting action on an email message. In *Analysis 2004*.
5. Dabbish, L., Kraut, R., Fussell, S., & Kiesler, S. Understanding email use: predicting action on a message. In *CHI 2005*.
6. Dredze, M., Brooks, T., Carroll, J., Magarick, J., Blitzer, J., & Pereira, F. Intelligent email: reply and attachment prediction. In *IUI 2008*.
7. Faulring, A., Myers, B., Mohnkern, K., Schmerl, B., Steinfeld, A., Zimmerman, J., Smailagic, A., Hansen, J., & Siewiorek, D. Agent-assisted task management that reduces email overload. In *IUI 2010*.
8. Horvitz, E., Jacobs, A., & Hovel, D. Attention-sensitive alerting. In *UAI 1999*.
9. Karagiannis, T. & Vojnovic, M. Behavioral profiles for advanced email features. In *WWW 2009*.
10. Kerr, B. & Wilcox, E. Designing REMAIL: reinventing the email client through innovation and integration. In *CHI 2004*.
11. Lockerd, A. Understanding implicit social context in electronic communication. MIT Master's Thesis.
12. Lu, J., Pan. S., Lai, J., & Wen, Z. Information at your fingertips: contextual IR in enterprise email. In *IUI 2011*.
13. Neustaedter C., Brush, A., Smith, M., & Fisher, D. The social network and relationship finder: social sorting for email triage. In *CEAS 2005*.
14. The Radicati Group. http://www.radicati.com
15. Siu, N., Iverson, L., & Tang, A. Going with the flow: email awareness and task management. In *CSCW 2006*.
16. Song, Y., Pan, S., Liu, S., Zhou, M. & Qian, W. Topic and keyword re-ranking for LDA-based topic modeling. In *CIKM 2007*.
17. Venolia, G., Dabbish, L., Cadiz, J., & Gupta, A. Supporting email workflow. MSR Tech Report MSR-TR-2001-88.
18. Venolia, G. & Neustaedter, C. Understanding sequence and reply relationships within email conversations: a mixed-model visualization. In *CHI 2003*.
19. Wainer J., Dabbish, L., & Kraut, R. Should I open this email?: inbox-level cues, curiosity and attention to email. In *CHI 2011*.
20. Yang, Y., Yoo S., & Lin, Frank. Personalized email prioritization based on content and social network analysis. *IEEE Intelligent Systems*, 25(4):12−18, 2010.