MinEMail: SMS Alert System for Managing Critical Emails

Kyle Rector
DUB Group — University of Washington
Seattle, WA 98195
rectorky@cs.washington.edu

Joshua Hailpern
HP Labs
Palo Alto, CA 94043
joshua.hailpern@hp.com

ABSTRACT
Email is the primary method of digital communication for most people, but the overwhelming quantity has led to a poverty of attention. Existing manual and automatic solutions that aim to save important emails from falling through the cracks have begun to address this problem, but may increase user workload, sacrifice efficiency, or fail to identify high value communications. In response, we developed MinEMail, an alert system that uses a text message (SMS) to remind and notify users of critical emails that may have been missed or forgotten. MinEMail provides an alert infrastructure as well as accurately labeling and predicting which emails are critical, and when and how they need to be addressed. To motivate our system, we also present an up-front study with 777 participants that aims to understand the state and limitations of email and SMS in enterprise. We conduct an experience sampling study of over 3000 emails in order to construct MinEMail's predictive models. Finally, we present the results from a 15 user ecologically valid real-world deployment of MinEMail in enterprise.

Author Keywords
Email; personal information management; information overload; SMS

ACM Classification Keywords
H.5.3. Group and Organization Interfaces: Asynchronous interaction.

INTRODUCTION
Despite the proliferation of social networking both within and beyond enterprise, email continues to be the primary method of digital communication for employees. However, a large quantity of email has led to a poverty of attention [29], particularly for industry professionals [9]. The content of emails can exacerbate email overload when a sender requests information, forces a deadline, or presses for an immediate reply [26]. Missing a critical message could stall progress on a project or cause frustration in a coworker. On the other hand, 64% of incoming emails are not relevant or do not need immediate attention [9]. Therefore, addressing all critical emails while ignoring the irrelevant is an ongoing challenge.

A number of automated filtering [3, 30] and manual management [1, 23, 30] solutions have been created to address this problem. However, existing research has shown that these approaches have limitations which may increase the workload of end-users while adding little benefit [22, 28, 29].

In response, we developed **MinEMail**, an alert system that uses an SMS[1] to remind and notify users of *critical* emails that may have been missed or forgotten. While many emails may have value to a user, we define *critical* emails to be those few emails that are crucially important to a person's career success. The results of our 777 enterprise employee study of email and SMS usage show that text messages are viewed significantly sooner than desktop email notifications. Therefore, SMS-based alerts allow critical notifications to be brought to the user's attention sooner. Further, our survey shows that 1 in 2 of enterprise users who *have* email on their mobile device do not enable push notifications, thus potentially missing incoming time-sensitive emails. MinEMail relies upon predictive models to identify critical emails and calculate when and how they should be addressed. While others have examined aspects of this problem [9, 11], MinEMail is the first end-to-end study, full implementation, deployment and validation of such a system to the authors' knowledge.

The primary contributions of this work are the development of a fully functional SMS alert system for *critical* emails, an ecologically valid real-world deployment study of MinEMail with 15 users, the creation of predictive models from almost 3000 emails collected during an experience sampling study, and a survey of 777 users about the current state and perceptions of email and SMS in enterprise.

BACKGROUND AND RELATED WORK
Email overload is a well established problem [29], with many emails vying for a user's attention based on content, personal utility, and task importance [27]. While there are non-software strategies employed to triage emails [25], including strategies to triage via mobile phone [21], some messages fall through the cracks (e.g. critical emails that arrive when users are away, or forgotten emails that never get addressed).

In the remainder of this section, we review related research, relevant systems, and analytics efforts similar to ours including; *manual* management where a user takes an action to prevent missing email; *automatic* management where systems provide email organization; and *completion* prediction, which helps assess what actions should be taken on a given email.

Manual Email Management

[1]We use the term **SMS** when referring to a text message received on a cellular phone.

CHI 2014, April 26–May 01 2014, Toronto, ON, Canada

http://dx.doi.org/10.1145/2556288.2557182

Several systems and features have been created to help users *manually* manage their email and prevent lost or forgotten messages. This ranges from annotation [23], to reminders or resurfacing [1], to traditional organization such as labels, flags or folders. Studies on these manual approaches show that they are generally unsuccessful [29], inefficient, and do not help in re-finding [28]. Further, regular monitoring of one's email reduces, rather than enhances, a user's productivity [22]. Subsequently, these results suggest that ideal solutions should require none to minimal extra work by the user to manage their inbox or receive reminders or alerts.

Automatic Email Management

Unlike the above manual approaches, many systems have automated the email management process from automatic labeling based on time and user [30], or automatically populated inboxes (e.g. GMail Priority Inbox or Contactually[2]). These tools segregate emails with value based on who you have emailed the most and relevant keywords [3]. While these approaches help address the onslaught of email, users still need to wade through a possibly large subset of messages, allowing critical[3] emails to be forgotten about as they fall lower and lower in any inbox or queue.

Completion Action Prediction for Email

Automatically attaching a completion action to an email [15, 24] or thread of emails [7] has been widely studied. These actions can range from detecting messages that need a reply [12], to be saved [11], or to have an attachment sent [4]. More personalized tasks have been modeled for managers in enterprise [5]. This work suggests that attaching and identifying completion activities to emails is reasonable and predictable, and we build upon this work.

One notable project is the work by Buthpitiya et. al. [9], who proposed an email personal messaging assistant. Their proposal linked email content to the calendar events and physical location of the end-user. However, they did not build their proposed system, deploy a real world test, or observe potential user behavior. The only contribution discussed in their work was a predictive model of high priority email. While the author's model performance is quite good, their rule-based approach is not grounded, justified and may not scale using the inboxes from six users. Further, their models are simulated on only three user inboxes, with 75 messages per user (a very small data set).

These projects add support for the development of an alert system, provide guidance on how to alert users, and validate that finding and remembering critical emails is a real problem. In this regard, MinEMail builds upon and extends the findings of these projects. It should be noted that while MinEMail is performing completion prediction (for replying, forwarding, sending an attachment, or reading the message), there are three key distinctions 1) We are leveraging a much larger data set within an enterprise context, 2) We are building a complete end-to-end SMS based alert system, and 3) We are testing the impact of these models and system using cross validation and a real-world deployment.

[2] `http://www.contactually.com/`

[3] There is a distinct difference between emails with value and those that are critical. This difference is highlighted later in the paper.

SCOPE AND MOTIVATION

MinEMail is a system designed to find critical emails and remind users to take action by leveraging SMS. **We define critical emails as messages that are "too important to miss or forget."** It is important to note that not every email with value is critical. MinEMail uses SMS because text message notifications have a significantly quicker response time than email, are accessible on mobile devices regardless of data network availability, and cover those 50% of mobile device users who do not enable email push notifications. Therefore, by judiciously using SMS to alert users of critical emails, we aim to mitigate the byproduct of email overload and emails falling through the proverbial crack.

MinEMail is *not* a priority inbox, as only a small percentage of emails- those that are critical- are sent to users. Similarly, MinEMail is not an email push notification system, because MinEMail does not push *all* incoming messages; nor are alerts sent immediately upon the arrival of a critical email. MinEMail notifies a user when the email is due, rather than when the message is first received. Lastly, MinEMail is *not* a system for setting reminders or to-do lists for emails.

It should be noted that while we develop models to predict when and how critical emails should be addressed, this is not our *primary* contribution: an end-to-end novel system combining two communication mediums, email and SMS.

Scenarios

Consider an enterprise employee, Don, who works in the marketing division of a company. Don is one of the 50% of employees who does not have push notifications on his mobile device and thus has no automated way to be informed about an incoming email message, regardless of its critical nature. On a given day, Don is emailed by his boss Roger with a critical question pertaining to their new marketing campaign. Without a timely answer, the entire campaign could be stalled. The following are two scenarios illustrating how MinEMail can assist Don addressing Roger's message in a timely fashion. With both of the following scenarios, MinEMail is judicious about its alerts, and does not overloaded the end-user with MinEMail notifications.

Not Sitting By Your Computer

Unfortunately, Don has left work[4] and is unaware of Roger's needs. If Roger's email is critical and needs to be addressed right away, MinEMail will send an SMS alert to Don within one hour. This greatly increases the chance that Don sees and takes action on the message. Roger thus received Don's answer within a reasonable time window, allowing the campaign to move forward.

Don't Forget...

Don is at his desk and reads the email from Roger, but he quickly forgets about it because at the same time, a coworker

[4] This scenario also applies to an employee being off-site for lunch, in a meeting, at a conference, etc.

	Question	Mean (SD)	Median [IQ Range]	Histogram
Email	**SUS Score**	80.01 (14.59)	82.50 [70.00,92.50]	
Email	**TLX mental demand**	58.76 (25.35)	64.29 [42.86,71.43]	
Email	**TLX physical demand**	41.03 (23.49)	28.57 [14.29,57.14]	
Email	**TLX temporal demand**	58.58 (22.75)	57.14 [42.86,71.43]	
Email	**TLX frustation level**	46.27 (23.61)	42.86 [28.57,57.14]	
Email	**TLX effort**	46.23 (24.60)	42.86 [28.57,71.43]	
Email	**TLX Total**	41.81 (15.23)	42.86 [30.95,52.38]	
Email	**Value-Benefit Level**	5.59 (1.32)	6.00 [5.00,7.00]	
SMS	**SUS Score**	73.77 (16.24)	77.50 [62.50,87.50]	
SMS	**TLX mental demand**	43.03 (20.43)	42.86 [28.57,57.14]	
SMS	**TLX physical demand**	44.08 (23.53)	42.86 [28.57,57.14]	
SMS	**TLX temporal demand**	52.66 (22.64)	57.14 [28.57,71.43]	
SMS	**TLX frustation level**	40.31 (22.29)	28.57 [28.57,57.14]	
SMS	**TLX effort**	39.09 (21.74)	28.57 [17.86,57.14]	
SMS	**TLX Total**	36.53 (13.76)	35.71 [26.19,45.24]	
SMS	**Value-Benefit Level**	5.07 (1.52)	5.00 [4.00,6.00]	

Table 1: Usability Metrics

SUS [8] scores range 0-100, with higher scores being considered better
NASA-TLX [17] scores range 0-100, with lower being considered better
Value-Benefit Level ranged from 1-7, with higher being better

Question	Mean (SD)	Median [IQ Range]	Histogram
Email High Priority (%)	0.24 (0.23)	0.15 [0.05,0.30]	
SMS High Priority (%)	0.22 (0.26)	0.10 [0.05,0.30]	
Email Late Reply (min)∗	666.10 (10019.51)	60.00 [15.00,120.00]	
SMS Late Reply (min)∗	66.49 (1079.25)	10.00 [5.00,20.00]	
Annoying (Email/hour)∗	16.89 (50.91)	5.00 [2.08,20.00]	
Annoying (SMS/hour)∗	22.68 (74.32)	5.00 [1.35,10.00]	

Table 2: Quantitative Responses

∗ Histogram shown in log scale due to long tailed data

approaches his cubicle to ask a question. After a few days, Don has still not responded to Roger. Meanwhile, MinEMail has predicted that this is a critical request from Roger that should be addressed within a few days. Given Don's inaction, MinEMail sends an SMS to Don, reminding him about Roger's request. Furthermore, MinEMail provides Don with the infrastructure to respond on his mobile device, even if he does not have a corporate email application. In this situation, MinEMail helps remind an end-user of a forgotten email rather than a newly arrived message.

SURVEY OF EMAIL AND SMS USAGE IN ENTERPRISE

To motivate and guide this work beyond the existing literature, we deployed a survey on and perceptions about email and SMS usage within an enterprise setting. The survey was distributed to all HP employees located in Palo Alto, to which 777 employees responded (30.47% Male, 66.92% Female, 2.61% not reported) with a mean age of 39.37 years (sd = 11.45). Participants spanned multiple parts of the company[5] and educational attainment[6]. We strongly believe that this is a broad and well representative set of users spanning a large tech company.

The survey asked basic demographic questions, usability and effort metrics (Table 1), quantitative questions (Table 2) and qualitative questions. These questions attempted to elicit how email and SMS are used in enterprise, and the boundaries between when it usable and when it becomes annoying.

[5]273 Management, 41 research, 272 engineering, 25 finance, 8 legal, 27 administrative assistant, 31 business, 127 other

[6]71 PhD/JD/MFA, 237 MS/MA, 379 BS/BA, 85 AD, 32 HS

Results & Discussion

Overall, results show that email has a significantly higher degree of satisfaction/usability (SUS) and value (Value-Benefit Level), but it is more demanding to use (TLX) across almost all dimensions. These differences are all statistically significant ($p<0.001$) using an independent 2-group Student t-test[7]. Users also reported (Table 2) a willingness to tolerate a higher mean amount of SMS alerts per hour as compared to email before they become annoying ($p=0.004$)[8].

Further, user responses (Table 2) show that email has a higher percentage of high priority messages ($p<0.001$)[9], yet the amount of time before a response is considered late differs by a large degree ($p<0.001$)[10]. An SMS is considered late in minutes, while an email is considered late in hours. This suggests that even though emails have a higher chance to contain highly valuable information, there is a delay in addressing their content.

It is important to note that only 50.56% of respondents use push email on their mobile device, suggesting that if a user is away from their computer, half of the users will likely not be notified if a critical email arrives in their inbox.

Users reported[11] receiving 17.51 messages per day (sd = 29.14)[12], with a median of 10.00 [5.00, 20.00] messages. In addition, 47.76% of users reported to look at an SMS after being notified within "a minute or two," 41.95% "right away," 5.15% "about once an hour," and 5.15% reported various other times greater than one hour. Comparable statistics about email are gathered and discussed later in this paper.

SYSTEM & ARCHITECTURE OF MINEMAIL

The user responses reported above present SMS to be a powerful platform for facilitating critical email reminders. SMS has a quicker response time, higher threshold for annoyance, and lower usability demand. In addition, 50% of mobile device users do not have push notifications on their phones. SMS has a higher degree of accessibility than email: if users travel to a low data-coverage area, or a conference or event where the data network is over-stressed, email access may not be strong or readily available. However, SMS remains an open conduit for communication, allowing users to still receive messages (and, if necessary, get to a computer with good internet access).

Based on the needs outlined in previous work and our user survey, we created MinEMail, a system that automatically reminds users to act on critical messages and meetings by sending an SMS. MinEMail allows users to be notified of a crit-

[7]TLX Physical Demand is significant with p=0.01.

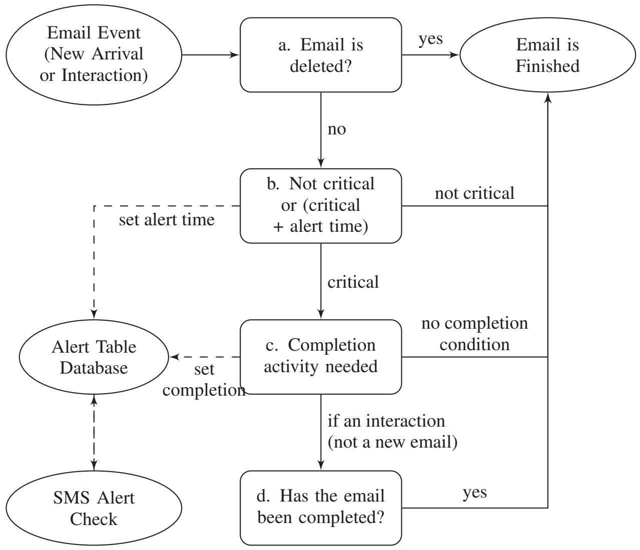
[8]Because these distributions were not normally distributed, Wilcoxon Rank-Sum was used.

[9]See footnote 8

[10]See footnote 8

[11]While developing predictive models for MinEMail, we logged existing email interaction patterns. Given the wide variety of mobile platforms in use, it is not feasible to collect comparable data about SMS usage. We therefore asked similar questions about SMS within this survey, and compare responses to email later.

[12]The distribution of incoming messages was not normally distributed, therefore we also mention the median.



(a) *State Diagram*

a) **Email is deleted?** If the incoming event is a deletion of an email, consider email finished.

b) **Not critical or (critical + alert time)** Using the model referred to as "main" in Table 3, each email is classified as either *not* critical, critical immediately, critical in a few hours, critical this evening, critical tomorrow, critical in a few days, or critical in one week. These predicted completion times are store in "alert table" database.

c) **Completion activity needed** Using the other 6 models in Table 3 (one for each potential completion condition), each critical email is assigned one or more completion conditions. If assigned multiple completion conditions, only one is required to be completed for that email to be considered finished. Should no model positively label a critical email, we capture this failure condition and consider the email "finished." New completion conditions will *replace* any previously existing conditions set in the "alert table."

d) **Has the email been completed?** The current state of the email is compared to the assigned completion conditions (e.g. email marked as needing to be forwarded, was forwarded). For conditions not trackable by the OAI, emails are considered completed if message is opened or previewed.

(b) *Explanation of Steps*

Figure 1: Email flow chart in the Server.
Solid lines are logic flow, while dashed lines are actions. — Steps a-d do not include emails that are "finished." — SMS alert check runs every 10 minutes

ical email and act upon it before a key deadline is missed or a coworker is let down. While an SMS may increase an employee's workload, missing a critical message may prompt follow ups by co-workers that could be as or more disruptive than SMS. MinEMail therefore aired on not sending an alert, only sending 1 SMS per critical message, and kept total alerts well under the annoyance threshold (Table 2).

We built a server-client architecture with a Microsoft **O**utlook **A**dd-**I**n (**OAI**) as our client. The OAI sends user email data and email interactions to the MinEMail server. The server ranks and prioritizes each message, and decides when to send end-user notifications over SMS. Given MinEMail is an OAI, only actions in Outlook can be detected. Thus, if another medium is used our system we cannot detect the action and an unnecessary alert may be sent. However, even in this case, only 1 alert will be sent, minimizing any potential disruption.

In the remainder of this section, we describe the OAI and the MinEMail server architecture. In the following section, we detail the process used to create the predictive models used in the MinEMail architecture.

MinEMail OAI Client

To monitor and log user emails and email interactions, we created an OAI. This small augmentation to Outlook is completely transparent to the end-user with the exception of requesting a user's cellular phone number during installation (which is sent to the server, and is used to send SMS alerts).

The OAI actively monitors a user's inbox for incoming messages and user interactions with said emails. When a new message is received, the OAI is triggered causing metadata to be extracted and sent to the MinEMail server. When users interact with emails (e.g. preview, open, reply), those events are also sent to the MinEMail server. The models used, and features extracted are discussed in the following section.

The MinEMail OAI is exclusively a monitoring and reporting tool. All computations, modeling and alerts are controlled on the MinEMail server. This reduces the local-machine storage cost, footprint, and resources needed to run the OAI.

MinEMail Server

The MinEMail server is a Java-based Apache server that stores and processes email data and sends SMS alerts. Every time a "new" email arrives, or a user interacts with an email, that data is sent to the server by the OAI. This causes the updated email to pass through several evaluation steps detailed in Figure 1. Through these evaluations, emails are segregated between critical and non critical, when they should be addressed, and what activity should be done so they can be considered completed. The probabilistic models used in these evaluation steps (and how they were generated) are covered in the following section.

For those emails that are labeled critical and "unaddressed," a pending alert is stored in an alert table. Every 10 minutes, the MinEMail server iterates over the alert table and determines which critical emails are overdue. When an email comes due, the MinEMail server sends the end-user an SMS alert. These alerts include the sender's email address, the subject line, and a unique randomly generated link to the MinEMail server. This link, if clicked, redirects the user's browser to a "mailto" link hosted on the MinEMail server. This "mailto" link will cause the local email client (on a mobile device or desktop) to create a new email, pre-populated with the original email's content as a reply message (including To, CC, subject line, and body). This gives end-users the option to read in full or respond to the critical email on their mobile device even if their corporate email account is not on their phone. Users also have the option to handle the message offline, or to respond to the message on their computer.

Once an alert has been sent to a user's phone, that message is considered finished, and no additional alerts are sent. Likewise, if a user completes a critical email by addressing its predicted completion activity, no alert is sent. Lastly, if an email is determined to be non-critical, it is also considered finished and not assessed again. For data security and privacy, once an email is deemed not critical or finished, private information and the email's content is expunged from any database.

The central nature of the MinEMail server provides multiple benefits over integrating the models and SMS alerts into a local client. First, alerts can be sent to a user even if that individual's computer is not available (e.g. asleep, off, no network). Second, for users with multiple computers, MinEMail will sync email interactions so that redundant alerts are not sent. Lastly, this reduces local overhead on local machines, minimizing impact on end user's computer performance.

Scenario Walkthrough Of Architecture

To illustrate the inner-workings and process of MinEMail's architecture (specifically Figure 1), we present two walkthrough scenarios. We will revisit the enterprise employee Don and an email being sent by his boss Roger.

Scenario 1. Roger sends Don a question via email about their marketing campaign. Don's OAI sends the server the data from the new email:

1. The server receives the data extracted by the OAI, and stores it in a email lookup table. This entry (with the email's meta data) is used to process the email through the predictive models.
2. (step a) The email has just arrived (has not been deleted).
3. (step b) The email is run through the a model that determines the message *is* critical, and that it should be addressed later "this evening."
4. (step b) The an alert for this email with an evening due date is added to the Alert Table database.
5. (step c) The email is processed through a series of models that predict what potential completion conditions would address this email. The email is determined to need a reply to be completed.
6. (step c) The alert table is updated with the new information.
7. (step d) Because the email is new, step d is skipped and the evaluation is complete.

Scenario 2. Don opens a message in Outlook. The OAI then sends a notification to the server that a specific message was opened, and at what time:

1. The server receives the notification from the OAI, and stores the open event (and its corresponding timestamp) in said email's row in the lookup table.
2. (step a) The email in question has not been deleted.
3. (step b) The email is reassessed for being critical, and when it may be due. The model uses both the email metadata (as in Scenario 1) and any interactions (e.g. the open event/timestamp).
4. (step b) The alert for this email is updated, with an "tomorrow" due date.
5. (step c) The email is reprocessed through completion condition prediction. The email is determined to still need a reply to be completed. The Alert Table is not updated.
6. (step d) The email is an interaction, as is checked if any completion condition has been satisfied. Because no reply has been sent, the alert is kept in the Alert Table and the evaluation is complete.

If Don did not reply by the deadline, an SMS would be sent, and the alert would be removed from the Alert Table. As a result, the message would no longer run through the flow chart (regardless of any updates).

PREDICTIVE MODELS DEVELOPMENT

A critical component of the MinEMail system is the ability to identify and classify emails as being critical, when they need to be addressed, and by which method. While others have approached this problem through ungrounded a-priori rules [9] and qualitative interviews [11], we wanted to ground our approach in observable user behavior and both create and verify our predictive models via machine learning techniques. Within a corporate environment, such as HP, employee privacy is important. Thus, to further scope and guide any potential solution, we created models that would not leverage an email's semantic content. Rather, our solution uses on lower level features such as attachment or word count.

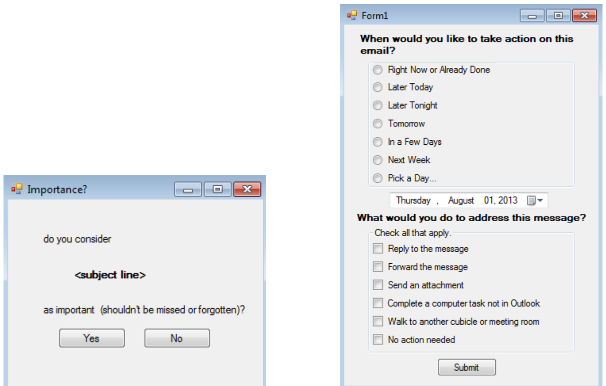
In order for MinEMail to classify emails, we required a gold standard by which to train and test our models. To this end, we followed an experience sampling approach similar to [16], minimally disrupting a user's workflow. An OAI would monitor and log low-level information about the individual emails, while experience sampling prompts would periodically ask individual users to label messages along three dimensions: 1) identify critical emails, 2) calculate when a user must act on the email, 3) determine what action would "complete" the email (whether or not said action is detectable by the OAI).

Using the labeled data, we modeled 7 distinct classification problems; one model determines if a message is critical and when it should be addressed (referred to as *main* in Table 3), while the other 6 models predict the six potential activities needed to "complete" a message. Six models are needed (one per activity) because the completion activities are not mutually exclusive. Thus the seven classification problems are:

- **If Critical & When (Main)**: If an email is not critical, no further action is needed. If it is critical, determine which of the 7 user specified time frames in which it needs to be addressed[13].
- **Forward**: If forwarded, email has been addressed
- **Reply**: If replied to, email has been addressed
- **Attach**: If an attachment is sent, email has been addressed
- **Offline**: If viewed, email has been addressed
- **Computer Task**: If viewed, email has been addressed
- **None**: No completion activity needed

Two sets of seven models were built. The first set is for **"new" emails**, those that have arrived and not been acted upon (used

[13]We opted to create one model determining if and when a message would need to be addressed, rather than two distinct models (separating if critical and when), because we felt that messages that required immediate action would likely look different from those that would be addressed in a day or week. Thus, by treating each time frame as a separate class in the classification task, we could ensure higher performance.



(a) *"Is Important" Prompt* (b) *Actionable Information*

Figure 2: Experience sampling message boxes.

in Scenario 1 in the previous section). These emails are completely missed (and thus no interactions have been taken on them). Subsequently, a set of models are created that have a limited set of features (no interaction variables). The second set of models are applied to **"old" emails** (used in Scenario 2 in the previous section), or those in which have a user interaction (e.g. previewed, moved to a folder, etc). By creating two sets of models, we can explicitly leverage user interactions on emails*only* when that data is present. If a user has not interacted with an email, we can optimize model performance based on the metadata that has been logged.

The remainder of this section details the experimental methodology and data mining results from generating the predictive models used by MinEMail.

Logging & Experience Sampling Architecture

To facilitate logging and experience sampling, a modified version of the OAI client for MinEMail was used with experience sampling prompts (and without requesting a phone number). Email metadata and user responses to experience sampling prompts were stored on a secure server. Extracted metadata included raw information (e.g. number of attachments, time that the message was sent, etc.), content-based information (e.g. cue phrases [19], request words [2]) and corporate information (e.g. direction in workspace hierarchy [14]). A full list of all 90 distinct features extracted is included as a supplemental document (Appendix A). To ensure data privacy, the body of the email, senders, or receivers were *not* logged on the server. User's emails were assigned a unique hash, thus allowing a message and user to be tracked without associating any behavior with an actual person.

Experience sampling prompts occurred on 30% of emails, as it was twice the median (and the upper quartile bound) of critical emails reported in the initial survey. Prompts were limited to those emails which a user had already interacted, ensuring that the email and its content were seen by the user. Prompts appeared immediately after a user closed, replied, or forwarded a message. If they previewed or opened a message, users would see a prompt after 10 seconds. For example, if a user previewed a message, the odds that an experience sampling prompt would appear would be 30%.

Experience sampling prompts (Figure 2) consisted of two steps. First, a high level binary choice (Figure 2a) asking if a message should not be missed or forgotten. If an email was marked as critical, a second prompt would appear with two questions (Figure 2b). This allowed users to specify the amount of time before the email would need an action taken, and what action (or lack thereof) would be required to address the email. For emails marked as unimportant, no further prompt was presented. Responses were sent to the server, updating the database about the email in question.

It should be noted that there is a bias with experience sampling. By only prompting users on emails that are being interacted upon, there is a high likelihood that said email has a modicum of value, and may be critical. Subsequently, a large percentage of experience sampled messages will be labeled as critical, missing those messages which are not. We augment the experienced sampled emails by treating email that is deleted without opening or previewing as not critical emails in our models. The user action of removing an unread message has the implication of not being critical.

Participants, Study Duration, & Data

From our original survey, we asked participants to indicate if they were interested in participating in a followup study. Those participants were emailed an invitation to participate with the data collection study, and 55 responded by filling out a demographic survey. They were then sent instructions on how to enroll. However, our only 19 unique users submitted data. Due to privacy concerns, we are unable to track which users participated. We can only report data from all 55 participants (mean age 48.04, σ= 11.11 years), with a reasonable gender split (41% Female) from a variety of backgrounds[14].

Participants were asked to participate for 1-2 weeks (5-10 business days). However, the actual duration was up to each individual. Our study was conducted in a real-world setting so duration could not be controlled. Subsequently, users participated on average for 8.68 days (σ=3.83) with a median participation length of 9 days [6,12]. This resulted in 7090 emails being logged[15].

Results: Email & Data Set Statistics

Users received a mean of 38.262 (σ= 23.969) and median 33.00 [20.00,47.80] messages per day. This is well above the number of SMS's reported by users. 24.5% (σ=24.10) of emails are opened by a user each day (median 14.5% [4.5,45.4]), while 50.8% (σ=29.4) are previewed by a user each day (median 47.4% [30.8,80.9]).

Of all 7090 emails, 1293 were experience sampled (55.45% as critical). In addition, 1597 emails were deleted without

[14] 21 Management, 5 Business, 2 Marketing, 4 Legal, 6 IT, 12 Engineering, 2 Finance, 3 Administrative Assistant

[15] While all of the data was included in our data set for model generation, statistics reported are normalized on a per-user basis. In addition, partial days only containing data from the hours of AM or PM were excluded to accurately report daily rates of usage.

	Model	Kappa	TP	FP	Precision	Recall	Accuracy
main	SMO	0.01 (0.01)	0.75 (0.01)	0.74 (0.01)	0.61 (0.07)	0.75 (0.01)	0.75 (0.01)
	TREE	0.25 (0.04)	0.78 (0.01)	0.59 (0.03)	0.74 (0.02)	0.78 (0.01)	0.78 (0.01)
	RFST	**0.95 (0.01)**	**0.98 (0.00)**	**0.06 (0.01)**	**0.98 (0.00)**	**0.98 (0.00)**	**0.98 (0.01)**
	SVM	0.80 (0.02)	0.91 (0.01)	0.04 (0.01)	0.93 (0.01)	0.91 (0.01)	0.91 (0.01)
forward	SMO	0.03 (0.07)	0.92 (0.02)	0.90 (0.04)	0.86 (0.06)	0.92 (0.02)	0.92 (0.02)
	TREE	0.00 (0.00)	0.92 (0.02)	0.92 (0.02)	0.85 (0.04)	0.92 (0.02)	0.92 (0.02)
	RFST	0.85 (0.12)	0.98 (0.01)	0.21 (0.17)	0.98 (0.01)	0.98 (0.01)	0.98 (0.01)
	SVM	**0.98 (0.02)**	**1.00 (0.00)**	**0.03 (0.04)**	**1.00 (0.00)**	**1.00 (0.00)**	**1.00 (0.00)**
reply	SMO	0.35 (0.09)	0.70 (0.05)	0.36 (0.05)	0.70 (0.04)	0.70 (0.05)	0.70 (0.05)
	TREE	0.57 (0.08)	0.79 (0.04)	0.21 (0.04)	0.79 (0.04)	0.79 (0.04)	0.79 (0.04)
	RFST	**0.97 (0.01)**	**0.98 (0.00)**	**0.02 (0.00)**	**0.98 (0.00)**	**0.98 (0.00)**	**0.98 (0.00)**
	SVM	0.95 (0.03)	0.98 (0.01)	0.03 (0.02)	0.98 (0.01)	0.98 (0.01)	0.98 (0.01)
attach	SMO	0.04 (0.08)	0.96 (0.01)	0.94 (0.06)	0.94 (0.01)	0.96 (0.01)	0.96 (0.01)
	TREE	0.00 (0.00)	0.96 (0.01)	0.96 (0.01)	0.93 (0.03)	0.96 (0.01)	0.96 (0.01)
	RFST	0.79 (0.09)	0.99 (0.01)	0.32 (0.12)	0.99 (0.01)	0.99 (0.01)	0.99 (0.01)
	SVM	**0.98 (0.05)**	**1.00 (0.00)**	**0.00 (0.00)**	**1.00 (0.00)**	**1.00 (0.00)**	**1.00 (0.00)**
offline	SMO	0.00 (0.00)	0.95 (0.02)	0.95 (0.02)	0.91 (0.04)	0.95 (0.02)	0.95 (0.02)
	TREE	0.00 (0.00)	0.95 (0.02)	0.95 (0.02)	0.91 (0.04)	0.95 (0.02)	0.95 (0.02)
	RFST	0.88 (0.08)	0.99 (0.01)	0.19 (0.13)	0.99 (0.01)	0.99 (0.01)	0.99 (0.01)
	SVM	**0.93 (0.06)**	**0.99 (0.01)**	**0.00 (0.00)**	**1.00 (0.00)**	**0.99 (0.01)**	**0.99 (0.01)**
comp. task	SMO	0.05 (0.05)	0.79 (0.04)	0.76 (0.04)	0.80 (0.09)	0.79 (0.04)	0.79 (0.04)
	TREE	0.14 (0.07)	0.80 (0.04)	0.70 (0.06)	0.82 (0.04)	0.80 (0.04)	0.80 (0.04)
	RFST	0.90 (0.06)	0.97 (0.02)	0.11 (0.07)	0.97 (0.02)	0.97 (0.02)	0.97 (0.02)
	SVM	**0.97 (0.01)**	**0.99 (0.00)**	**0.01 (0.01)**	**0.99 (0.00)**	**0.99 (0.00)**	**0.99 (0.00)**
none	SMO	0.08 (0.07)	0.77 (0.05)	0.71 (0.08)	0.77 (0.12)	0.77 (0.05)	0.77 (0.05)
	TREE	0.08 (0.04)	0.76 (0.06)	0.71 (0.07)	0.77 (0.09)	0.76 (0.06)	0.76 (0.06)
	RFST	0.90 (0.04)	0.96 (0.01)	0.11 (0.05)	0.97 (0.01)	0.96 (0.01)	0.97 (0.01)
	SVM	**0.98 (0.02)**	**0.99 (0.01)**	**0.02 (0.02)**	**0.99 (0.01)**	**0.99 (0.01)**	**0.99 (0.01)**

(a) **"New emails"** Truncated Models without Email Interaction Variables

	Model	Kappa	TP	FP	Precision	Recall	Accuracy
main	SMO	0.20 (0.01)	0.77 (0.01)	0.62 (0.02)	0.73 (0.01)	0.77 (0.01)	0.77 (0.01)
	TREE	0.49 (0.06)	0.81 (0.01)	0.32 (0.05)	0.78 (0.02)	0.81 (0.01)	0.81 (0.01)
	RFST	0.85 (0.03)	0.94 (0.01)	0.16 (0.02)	0.95 (0.01)	0.94 (0.01)	0.94 (0.01)
	SVM	**1.00 (0.01)**	**1.00 (0.00)**	**0.00 (0.01)**	**1.00 (0.00)**	**1.00 (0.00)**	**1.00 (0.00)**
forward	SMO	0.02 (0.05)	0.92 (0.03)	0.91 (0.05)	0.87 (0.04)	0.92 (0.03)	0.92 (0.03)
	TREE	0.00 (0.00)	0.92 (0.03)	0.92 (0.03)	0.85 (0.05)	0.92 (0.03)	0.92 (0.03)
	RFST	0.75 (0.15)	0.97 (0.02)	0.34 (0.17)	0.97 (0.02)	0.97 (0.02)	0.97 (0.02)
	SVM	**1.00 (0.00)**	**1.00 (0.00)**	**0.00 (0.00)**	**1.00 (0.00)**	**1.00 (0.00)**	**1.00 (0.00)**
reply	SMO	0.41 (0.08)	0.72 (0.04)	0.33 (0.04)	0.73 (0.05)	0.72 (0.04)	0.72 (0.04)
	TREE	0.32 (0.05)	0.69 (0.03)	0.39 (0.04)	0.72 (0.02)	0.69 (0.03)	0.69 (0.03)
	RFST	0.93 (0.03)	0.97 (0.01)	0.04 (0.02)	0.97 (0.01)	0.97 (0.01)	0.97 (0.01)
	SVM	**0.99 (0.01)**	**0.99 (0.01)**	**0.01 (0.01)**	**0.99 (0.01)**	**0.99 (0.01)**	**0.99 (0.01)**
attach	SMO	0.08 (0.18)	0.96 (0.01)	0.91 (0.10)	0.93 (0.03)	0.96 (0.01)	0.96 (0.01)
	TREE	0.00 (0.00)	0.96 (0.01)	0.96 (0.01)	0.93 (0.01)	0.96 (0.01)	0.96 (0.01)
	RFST	0.81 (0.11)	0.99 (0.01)	0.29 (0.14)	0.99 (0.01)	0.99 (0.01)	0.99 (0.01)
	SVM	**1.00 (0.00)**	**1.00 (0.00)**	**0.00 (0.00)**	**1.00 (0.00)**	**1.00 (0.00)**	**1.00 (0.00)**
offline	SMO	0.12 (0.17)	0.96 (0.01)	0.88 (0.09)	0.93 (0.04)	0.96 (0.01)	0.96 (0.01)
	TREE	0.00 (0.00)	0.95 (0.01)	0.95 (0.01)	0.91 (0.02)	0.95 (0.01)	0.95 (0.01)
	RFST	0.66 (0.23)	0.98 (0.01)	0.44 (0.22)	0.98 (0.01)	0.98 (0.01)	0.98 (0.01)
	SVM	**1.00 (0.00)**	**1.00 (0.00)**	**0.00 (0.00)**	**1.00 (0.00)**	**1.00 (0.00)**	**1.00 (0.00)**
comp. task	SMO	0.14 (0.07)	0.80 (0.04)	0.71 (0.03)	0.84 (0.02)	0.80 (0.04)	0.80 (0.04)
	TREE	0.30 (0.10)	0.81 (0.03)	0.55 (0.08)	0.79 (0.04)	0.81 (0.03)	0.81 (0.03)
	RFST	0.79 (0.04)	0.94 (0.01)	0.23 (0.05)	0.94 (0.01)	0.94 (0.01)	0.94 (0.01)
	SVM	**1.00 (0.00)**	**1.00 (0.00)**	**0.00 (0.00)**	**1.00 (0.00)**	**1.00 (0.00)**	**1.00 (0.00)**
none	SMO	0.16 (0.07)	0.78 (0.04)	0.67 (0.07)	0.81 (0.04)	0.78 (0.04)	0.78 (0.04)
	TREE	0.39 (0.13)	0.81 (0.05)	0.49 (0.07)	0.81 (0.05)	0.81 (0.05)	0.81 (0.05)
	RFST	0.82 (0.09)	0.94 (0.03)	0.18 (0.07)	0.94 (0.03)	0.94 (0.03)	0.94 (0.03)
	SVM	**0.98 (0.02)**	**0.99 (0.01)**	**0.02 (0.02)**	**0.99 (0.01)**	**0.99 (0.01)**	**0.99 (0.01)**

(b) **"Old" emails** Full Models with Email Interaction Variables

Table 3: Model Performance (5-Way Cross Validation)

"main" is the model that predicts if a message is not critical vs. the seven time intervals indicating when to address a critical email

being read or previewed. Thus the final data set used to model and predict user behavior consisted of 717 critical emails, and 2173 non-critical emails. 33% of this data was critical, which is slightly higher than what users self reported. This rate of critical emails is similar to previous work [9].

Of those emails marked as critical, users indicated that 90.95% needed to be addressed "right away," 5.57% "in a few hours," 0.23% "this evening," 1.01% "tomorrow," 1.86 % "in a few days," 0.31% "next week," and for only one email did a user manually select a calendar date. Overall, 408 messages could be completed by sending a reply, 177 required no action but were important enough to read, 155 required a computer task outside of Microsoft Outlook, 57 could be completed by forwarding said email, 34 required offline interaction, and 27 required sending an attachment[16]. Clearly, not all of these completion conditions can be detected by an Outlook add-in. We therefore consider non-detectable completions of critical emails satisfied if the user has read the message, because they would be aware of its content.

Results: Model Performance

In order to create a system that can automatically alert users to critical emails before they are due, we constructed a series of predictive models. We generated these distinct models using Weka: a J48 decision tree, SMO (sequential minimal optimization), RFST (Random Forest) and SVM (Table 3). Models for "new" emails (without interaction data such as Previewing a Message) are shown in Table 3a, while performance of "old" emails (a user *has* interacted) are in Table 3b.

While we expected a rigorous competition between these four distinct approaches, both J48 and SMO performed extremely poorly with Kappa statistics hovering around 0.4 and 0.1 respectively. However, SVM performed quite well with extremely high Kappa Scores that could be considered Excellent[10, 13], Very Good [6] or Almost Perfect[20]. Full results from a 5-fold cross validation are shown in Table 3[17]. It is intriguing, though not unsurprising, that models performed better when modeling "old" emails that included user interactions (Table 3), given that user behavior seems to be an important indicator of the critical nature of emails, and how to best address them. Subsequently, we used the models with the highest Kappa scores in MinEMail.

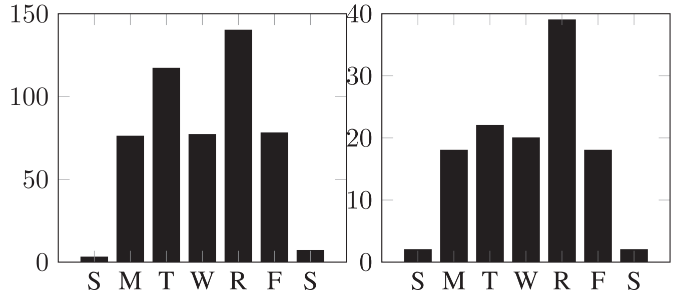
Unlike higher level NLP analysis, our lower level features (Appendix A) are easy to calculate and have little system overhead. Given the model performance had nearly perfect Kappa scores, it appears that the low level features are highly predictive – a similar finding to Hailpern et al. [16].

STUDY: MinEMail DEPLOYMENT

In order to study the impact of MinEMail in an enterprise setting, we conducted an ecologically valid study to observe user behavior and system performance. Since MinEMail is predicated upon alerting users to emails in their day-to-day

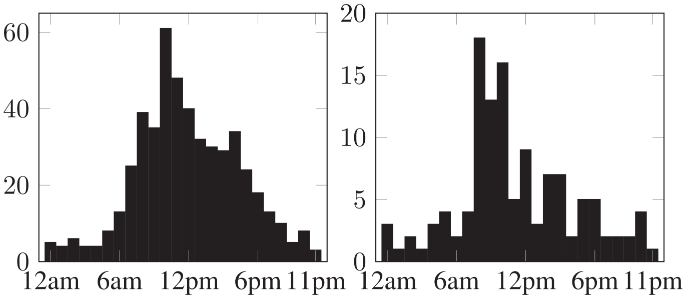
[16]When examining user responses as to how to "complete" a critical email, users were allowed to indicate as many potential completion conditions as they wished. Therefore the total number of completion condition counts sums to 859 instead of 717.

[17]The data shown does not include the features of reading level. This feature consumed a large degree of system resources on the end-user's computer. When the models were built both with and without reading level variables, the was almost no difference in performance. Therefore, reading level was omitted to reduce OAI impact.



(a) *Number of critical emails* (b) *Number of alerts scheduled*

Figure 3: Freq. of critical emails and assigned alerts per day.



(a) *Number of critical emails* (b) *Number of alerts scheduled*

Figure 4: Freq. of critical emails and assigned alerts per hour.

activity, it would be extremely difficult to validate MinEMail through a controlled lab study. Subsequently, our validation study has both the benefits and challenges of a real-world deployment study. To the authors' knowledge, this is the first study of a real system (ecologically valid or otherwise) hybridizing SMS and email.

Participants & Study Duration

An email was sent to employees of HP in Palo Alto inviting them to use MinEMail. 49 employees expressed initial interest and filled out a demographic survey. These respondents were then sent instructions on how to enroll. Due to privacy concerns, especially over email, we were unable to track which specific users participated[18] We thus report demographic data from all 49 participants that signed up. 41% were Female, the mean participant age was 42.75 (sd = 10.52 years), and respondents represented a broad cross-section of HP (14 Management, 6 Business, 3 Marketing, 4 IT, 14 Engineering, 1 Finance, 3 Administrative Assistant). Respondents were asked to participate for 1-2 weeks (5-10 business days). However, the actual duration was up to each individual.

By the time of the study completion, 15 unique users had chosen to participate in our study, with an average participation time of 11.46 days (sd=3.40) and a median of 13 days [9,14]. During the study, MinEMail parsed 4027 emails, which is more emails than previous work has used for initial training [9, 11]. Of those emails, MinEMail identified 498 as critical. In the following sections, we will discuss user behavior and MinEMail usage within this study.

Results & Discussion: User Behavior

Data collected about the critical emails, as identified by MinEMail, provide an interesting insight into the work patterns of enterprise employees and the behavior of critical activities (Figures 3 and 4).

For instance, Tuesday and Thursday contained the most critical emails (see Figure 3a). However, this data also appears to illustrate the fatigue of the workweek generally taking its toll. While Tuesday and Thursday both had a high volume of critical emails, users seemed more aware of their email messages on Tuesday than on Thursday. This discrepancy is so great that Tuesday had the lowest ratio of alerts scheduled per critical email for the weekdays (18.8%), while Thursday had the largest (26.0%). Tuesday and Thursday also had the most sent alerts (see Figure 3b). We cheekily suggest that should a critical issue arise, it's best to send the relevant email on a Tuesday, and to avoid Thursday communication, if possible.

We further break down user interaction patterns by hour (Figure 4). The pattern of critical emails and alerts scheduled per hour reflect a normal working schedule, with a peak of critical emails and alerts scheduled in the morning. Upon close examination of Figure 4a, we see an influx of email right before lunch, with a corresponding alert peak occurring right after lunch (Figure 4b) as many of those messages may be missed. There also appears to be a similar dual email and alert humps at the end of the day, as critical thoughts are sent before heading home. It is worth noting that any alerts scheduled between 7pm-9am were then sent at 9am the next morning.

Results & Discussion: MinEMail

As stated earlier, MinEMail identified 498 emails as critical (12.37%); this is a slightly more conservative number than those labeled as critical in the initial survey and previous work. While some critical emails may not have been identified, we aim to send fewer alerts that correctly identify a missed important email, rather than spam users with a large quantity of messages. Out of the 498 critical emails, 178 had no detectable completion condition assigned to them. We therefore focus our discussion on the remaining 320 critical emails which had the potential to have an SMS alert sent.

As expected, users were generally on top of their critical emails, addressing 62.19% of them in advance of the predicted alert time. Subsequently, only 121 alerts were actually sent over the entire study period. This represents only 37.81% of emails, and an average of 0.70 alerts being sent daily to each user. Thus, alert disruption was kept extremely low, contributing minimally to any annoyance from an increased SMS rate. Furthermore, we prohibited text messages from being sent during the hours of 7:00pm-8:59am. For a real system, we could imagine individual users setting their own "Do Not Disturb" time frame.

Once the alerts are sent, we are unable to track user behavior, given that users' cell phones are not instrumented. We cannot

[18] Anonymized user behavior was tracked with unique hash-codes.

track how many SMS alerts were viewed by end-users, or after how long. Further, users can pursue multiple courses of actions after receiving a message[19] or the sender might take the initiative to follow up with the receiver.

We can only track the actions of clicking the link in the SMS or replying to a message within an email client, as directly associated with the critical email for which the alert was sent. With this in mind, the majority of the sent alerts (53.72%) prompted an action to be taken directly in Outlook or via the link provided in the SMS. Out of these alerts that sired activity, 84.61% prompted the user to return to their desktop Outlook client to address the message, and 18.46% of the alerts resulted in users clicking the "mailto" url contained in the SMS and sending the populated email using their cell phone mail client. Given the limitations of tracking the resulting user behavior, these results are very supportive of the utility of MinEMail and its alert system. Given that almost half of the received alerts resulted in an email client interaction, it is suggestive that MinEMail had an influence.

Followup Survey

In an attempt to better understand the impact of MinEMail and alert quality, we deployed a follow-up survey after the study. However, following an ecologically valid deployment at HP, we were unable to determine which 15 participants, out of the 49 employees who signed up actually participated. Subsequently, our survey was sent to all 49 individuals and we received an extremely low (only 5) number of responses. Though this sample size is too small to conduct formal statistics (due to low confidence intervals) or generalize these opinions, we do report responses to illustrate user reactions and provide a broader view of usage statistics above.

Quantitative survey responses (5-point Likert Scale) suggest that MinEMail was easy to use: Users did not need support (μ=4.4), MinEMail was easy to learn (μ=3.8), and did not need to learn a lot of information before using (μ=4). In fact, one participant explicitly stated that *its designed to be easy to use.* In addition, the majority of survey respondents felt that they were receiving the right amount of text messages (3 people). Qualitative responses further indicated that participants enjoyed the tool because they received *alerts on an important email and ability to respond without having to dig through email to find the message*, and receiving *alerts on critical missed emails without having to sort through my inbox.* However, users were not all positive in their responses, and saw room for improving MinEMail for future use. They were interested in making it a more adaptive system to their preferences ($\mu = 4.4$), include a desktop dashboard of pending MinEMail alerts (μ= 3.8), and have the ability to manually refine alert rules (μ=4.2). Overall, this suggests that MinEMail is a useful tool, one that has room to grow into a major component of enterprise email management.

FUTURE WORK

Overall, MinEMail successfully tackled the missing or forgotten critical email challenges uncovered earlier in this research. Throughout our validation study, MinEMail repeatedly identified critical emails for notification, and when messages were missed or forgotten, sent SMS alerts to users' cellular phones. While it was not possible to track all resulting interactions from the SMS alerts (e.g. online communication), we observed a high degree of activity from users after alerts were sent (subsequently addressing the critical messages). However, like any research, there are avenues for future growth and improvement.

With broader and longer deployments, MinEMail could become an adaptive system – learning each users' unique "take" on their critical email. This could be done with a proactive learning approach (akin to our experience sampling study), or more subtly by observing user behavior and response times. Another opportunity for future work could be to allow MinEMail to better respond to a user's current activity [18], scheduling alerts based on calendar events or physical location. Thus alerts might be sent to a user early if a meeting with the sender is imminent. Perhaps most exciting would be to develop MinEMail's SMS system to become interactive. Thus users could get more information about a critical email and respond entirely over SMS. This further leverages the benefits of SMS (over data network interactions), and, as the barrier to interaction is far lower for SMS than email (as shown in our initial study), could allow for easier/quicker critical email responses.

We believe that MinEMail alone will not solve the problem of email overload. While additional factors, including social solutions, are needed to resolve all related issues, it is outside the scope of this paper. While our system may not appeal to all users, just as not all users have their work email on their mobile phones, our system evaluation shows that people continued to use MinEMail and they found benefit.

CONCLUSION

We present MinEMail, an SMS alert system for critical emails. We ground our research in previous work, a 777 person survey, and an experience sampling study of 3000 email messages. Our system can assist an enterprise employee to become aware of a critical email they may have missed or forgotten, and give them the ability to address the message. Through an evaluation of 15 employees of HP with varying backgrounds, we have shown that our system directly motivates users to take action in response to MinEMail's messages. Additionally, that the data we have acquired and analyzed to motivate our system provides general insights into email and SMS usage in enterprise.

REFERENCES

1. Mailbox - Put Email In Its Place. `http://www.mailboxapp.com/`, 2013. [Online; accessed 28-August-2013].

2. Ways of asking questions and making requests – MacMillan Dictionary. `http://www.macmillandictionary.com/us/thesaurus-category/british/`

[19] e.g., clicking the link in the SMS, replying to the original message on their computer or mobile phone, calling a coworker, sending a text message, starting a new email using their phone, starting a new email with Microsoft Outlook, creating a new calendar event, or walking to another location.

`Ways-of-asking-questions-and-making-requests`, 2013. [Online; accessed 28-August-2013].

3. Aberdeen, D., Pacovsky, O., and Slater, A. The learning behind gmail priority inbox. In *LCCC: NIPS 2010 Workshop on Learning on Cores, Clusters and Clouds* (2010).
4. Albakour, M.-D., Kruschwitz, U., and Lucas, S. Sentence-level attachment prediction. In *Advances in Multidisciplinary Retrieval*. Springer, 2010, 6–19.
5. Alberts, I., and Forest, D. Email pragmatics and automatic classification: A study in the organizational context. *Journal of the American Society for Information Science and Technology 63*, 5 (2012), 904–922.
6. Altman, D. G. *Practical statistics for medical research*, vol. 12. CRC Press, 1991.
7. Bellotti, V., Ducheneaut, N., Howard, M., and Smith, I. Taking email to task: the design and evaluation of a task management centered email tool. In *Proceedings of the SIGCHI 2003*, ACM (2003), 345–352.
8. Brooke, J. SUS-A quick and dirty usability scale. *Usability evaluation in industry* (1996).
9. Buthpitiya, S., Madamanchi, D., Kommaraju, S., and Griss, M. Mobile context-aware personal messaging assistant. In *Mobile Computing, Applications, and Services*. Springer, 2010, 254–272.
10. Byrt, T. How good is that agreement? *Epidemiology 7*, 5 (1996), 561.
11. Dabbish, L. A., Kraut, R. E., Fussell, S., and Kiesler, S. Understanding email use: predicting action on a message. In *Proceedings of the SIGCHI 2005*, ACM (2005), 691–700.
12. Dredze, M., Wallach, H. M., Puller, D., Brooks, T., Carroll, J., Magarick, J., Blitzer, J., Pereira, F., et al. Intelligent email: Aiding users with ai. In *AAAI* (2008), 1524–1527.
13. Fleiss, J. L., Levin, B., and Paik, M. C. The measurement of interrater agreement. *Statistical methods for rates and proportions 2* (1981), 212–236.
14. Gilbert, E. Phrases that signal workplace hierarchy. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work*, ACM (2012), 1037–1046.
15. Gwizdka, J. Reinventing the inbox: supporting the management of pending tasks in email. In *CHI'02 extended abstracts on Human factors in computing systems*, ACM (2002), 550–551.
16. Hailpern, J., Jitkoff, N., Subida, J., and Karahalios, K. The clotho project: predicting application utility. In *Proceedings of the 8th ACM Conference on Designing Interactive Systems*, ACM (2010), 330–339.
17. Hart, S. G., and Staveland, L. E. Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. *Human mental workload 1*, 3 (1988), 139–183.
18. Horvitz, E., Jacobs, A., and Hovel, D. Attention-sensitive alerting. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, Morgan Kaufmann Publishers Inc. (1999), 305–313.
19. Knott, A. *A Data-Driven Methodology for Motivating a Set of Coherence Relations*. PhD thesis, The University of Edinburgh: College of Science and Engineering: The School of Informatics, July 1996.
20. Landis, J. R., and Koch, G. G. The measurement of observer agreement for categorical data. *biometrics* (1977), 159–174.
21. Pierce, J. S., Bunde-Pedersen, J., and Ford, D. A. Triage and capture: Rethinking mobile email. *interaction 5* (2010), 7.
22. Renaud, K., Ramsay, J., and Hair, M. " you've got e-mail!"... shall i deal with it now? electronic mail from the recipient's perspective. *International Journal of Human-Computer Interaction 21*, 3 (2006), 313–332.
23. Rohall, S. L., Gruen, D., Moody, P., Wattenberg, M., Stern, M., Kerr, B., Stachel, B., Dave, K., Armes, R., and Wilcox, E. Remail: a reinvented email prototype. In *CHI'04 extended abstracts on Human factors in computing systems*, ACM (2004), 791–792.
24. Scerri, S., Handschuh, S., and Davis, B. The path towards semantic email: Summary and outlook. In *Proceedings of EMAIL-08: the AAAI Workshop on Enhanced Messaging* (2008).
25. Tang, A., Siu, N., Lee, I., and Fels, S. Go with the flow: How users monitor incoming email. Tech. Rep. TR-2005-22, Department of Computer Science University of British Columbia, 2005.
26. Thomas, G. F., King, C. L., Baroni, B., Cook, L., Keitelman, M., Miller, S., and Wardle, A. Reconceptualizing e-mail overload. *Journal of Business and Technical Communication 20*, 3 (2006), 252–287.
27. Wainer, J., Dabbish, L., and Kraut, R. Should i open this email?: inbox-level cues, curiosity and attention to email. In *Proceedings of the CHI 2011*, ACM (2011), 3439–3448.
28. Whittaker, S., Matthews, T., Cerruti, J., Badenes, H., and Tang, J. Am i wasting my time organizing email?: a study of email refinding. In *PART 5———Proceedings of the 2011 annual conference on Human factors in computing systems*, ACM (2011), 3449–3458.
29. Whittaker, S., and Sidner, C. Email overload: exploring personal information management of email. In *Proceedings of the SIGCHI '96*, ACM (1996), 276–283.
30. Yiu, K. S., Baecker, R., Silver, N., and Long, B. A time-based interface for electronic mail and task management. *ADVANCES IN HUMAN FACTORS ERGONOMICS 21* (1997), 19–22.