Mobile Context-Aware Personal Messaging Assistant

Senaka Buthpitiya*, Deepthi Madamanchi, Sumalatha Kommaraju,
and Martin Griss

Carnegie Mellon Silicon Valley,
NASA Research Park, Bldg. 23, Moffet Field, CA 94035
{senaka.buthpitiya,deepthi.madamanchi,sumalatha.kommaraju,
martin.griss}@sv.cmu.edu

Abstract. A previous study shows that busy professionals receive in excess of 50 emails per day of which approximately 23% require immediate attention, 13% require attention later and 64% are unimportant and typically ignored. The flood of emails impact mobile users even more heavily. Flooded inboxes cause busy professionals to spend considerable amounts of time searching for important messages, and there has been much research into automating the process using email content for classification; but we find email priority depends also on user context.

In this paper we describe the Personal Messaging Assistant (PMA), an advanced rule-based email management system which considers user context and email content. Context information is gathered from various sources including mobile phones, indoor and outdoor locationing systems, and calendars. PMA uses separate scales of importance and urgency to prioritize emails and to decide on an appropriate action, such as SMS to user, defer to later, file or forward. Initial results yield 96% recall and 88% precision in importance classification of emails; 95% recall and 92% precision in urgency classification of emails. PMA shows a 30X reduction in false-negative rates over existing systems. A key contribution of our work has been to leverage an extensible set of context information, gathered in a mobile environment, for the classification of emails and customizable decision making.

Keywords: Context-aware computing, mobile, messaging, email assistant, Jess rules, prioritization, productivity, urgency, priority, SMS.

1 Introduction

Today email is not only used as a messaging tool but also for task management, information storage and archiving, scheduling and social communication [1]. This clearly explains that email has evolved into a primary and very frequently used communication tool. Therefore, the problem of email overload is ubiquitous [2]. There is a strong need for a system which can intelligently prioritize and manage email for busy professionals. On average a professional can receive upwards of 50 email messages per day, including all the email inboxes the user owns; typically only an average

* Corresponding author.

S. Ystad et al. (Eds.): MobiCASE 2009, LNICST 35, pp. 254–272, 2010.

of 23% of these emails are important at a given instant in time based on his context. These important emails either get lost in the flood of unnecessary email or are forgotten even after being marked for later processing because there is no reminder system to notify users about the unprocessed-marked emails. These deferred important emails should instead be presented at the right time. It is more helpful to have a user examine a message when needed, rather than in the order it arrives in the mail box, often interrupting important work. While this is important to all busy professionals, the problem is particularly acute for users who are increasingly mobile, with limited time and dynamically changing situations.

We envision a personal messaging system that addresses many of the concerns described above. For example, a busy anesthetist may want to receive notifications of only the important mails at night. Usually his nurse, who works from 9 a.m. to 5 p.m., has access to his inbox to take care of them. But, during the rest of the day, he may want to have a system which notifies him of important emails. During the night, he wants all urgent emails to be sent to his smart phone, perhaps summarized as SMS. The incoming urgent emails could instantly trigger an alarm. Similarly, a mobile engineer may notice oxidation on some equipment – not yet serious, but something to deal with eventually. The project manager is informed about this oxidation, not when the message is sent, but instead when she is visiting that area. If a worker sends a message to his manager about a repair issue in a manufacturing facility (requiring future attention), the manager may receive it while he is, for example, enmeshed in arranging a conference. Since the information is not time critical, he does not address it at that time, and it sits in the manager's inbox where it is likely to be forgotten; instead it would be better if this information is delivered (or a reminder issued) only when he is planning maintenance for equipment in that area.

We have developed a Personal Messaging Assistant (PMA) prototype, which primarily captures incoming (email) messages, analyzes them based on the email content and the user context and stores or processes the messages for immediate or eventual delivery. The PMA system consists of two parts: an email scanning server and a client on a mobile device. The messages coming into the inbox are scanned by the server periodically at stipulated time intervals and repeatedly prioritized by an intelligent rule-based system that makes decisions based on message content, user preferences and contextual information. Contextual information transmitted from the mobile device includes current location, topics of interest for the user, feedback regarding received messages, current activity, or social context such as who is nearby. The system also uses information from email content, the calendar, history of movement and patterns of information access. The mobile device periodically sends user context information to the server via http; and also receives selected messages delivered via SMS[1]. This system can be customized to suit an individual's preferences and social context, such as a dynamically changing list of important colleagues.

While a multitude of email-content based email sorting/classification programs exist, we believe our system outperforms these systems by taking context-information into consideration. PMA outperforms other context based email sorting programs [3]

[1] Even if the user's mobile phone has a built-in email client, such as Blackberry, prioritization and filtering of email will help reduce clutter and ensure that the most appropriate messages are seen first. We used SMS for convenience to avoid modifying the client email reader, and as a ubiquitous channel available on most phones.

with the use of separate scales to measure the importance and urgency of an email. This allows PMA to classify and decide on an action to take with an email with greater accuracy. The improved performance PMA offers can be seen with the large reduction in false-negatives and the high accuracy and precision results. The rest of the paper is divided into five sections which describe, respectively, other work and programs related to PMA; the design and implementation of the PMA system; performance testing done on the system and an interpretation of the results; the future improvements we hope to make with the system; and finally we conclude with a summary of our work and contributions.

2 Related Work

The Email overload has been a problem since the early 1990s [2]. Since then several techniques have been proposed to effectively manage emails. The idea of prioritizing and filtering the emails based on personal attributes is not a novel concept. We list such work and highlight their key features in this section. We also highlight relevant research related to filtering email spam.

2.1 Personal Email Assistant

The Personal Email Assistant [3] prototyped a system which can prioritize, filter, index and re-file all the emails. This system used information retrieval (Lucene) and statistical methods (WEKA, SVM) to classify incoming emails from a Microsoft Exchange email server, and then a rule system to decide on what to do with each class of mail. However, their system did not address mobile users, or changing dynamic context. The system was also envisioned as part of an ensemble of information management agents, rather than as a distinct tool.

2.2 Cool Agent Personal Assistant

The HP Laboratories Cool Agent project [4][5], continued later as the UCSC ScateAgent project [6], developed a series of context-aware personal and team assistant agents to manage information, arrange meetings, help software engineering teams and offer travel advice. Each of these used rule-based systems in conjunction with personal preferences and context, access to calendar information as well as mobile devices. In both systems, the context-aware and preference-aware notification agents play an important role. The user can set his preferences and the notification agent notifies the user whenever there is a change in the system. The notification agent used in this system is very similar to the delivery agent in PMA.

The CoolAgent Personal Assistant (PA) also uses multiple agents which can route messages and notifications. Based on the preferences and context, the notification agent routes the messages via one or more channels (email, voicemail, IM/jabber, or pager). One part of the overall PA vision is for the Personal Email Assistant (PEA) to interact as a peer or child of the other agents. In one direction, the PEA uses the calendar agents, the notification agents and the PA to find information and to communicate with the user. In the other direction, email messages concerning meetings could trigger the meeting agents, or at least monitor, prioritize and route email relevant to specific meetings.

2.3 Conventional Email Rules and Filters

Almost every email client can allow the user to set up some rules which can recognize (or filter) messages which have certain keywords or email header information. The rules can then delete, file or forward the messages. For example, Microsoft Outlook has two levels of rule settings, one on the Microsoft Exchange Server and another on the local outlook client. Microsoft Exchange Server uses rules on mailboxes and other folders to automatically execute actions on objects in the folders. One can use rules to develop applications that carry out predefined or custom actions, even at times when the client application is not running. Rules can be performed on the Exchange store (server-side) or on Microsoft Outlook (client-side). One of the main disadvantages of client-side rules is that they can only run when Outlook is running online. Also, the rules are very rigid and have only two states for any set rule: Yes or No. In our work, the PMA server can also act on the user's behalf even when the email client is not running.

Gmail provides filters to manage the flow of the incoming messages. Using these filters, one can perform only basic operations such as forwarding, deleting, or labeling based on the combinations of keywords, sender, recipients etc. These filters are very rigid and do not take into account contextual information. Also certain useful settings, such as forwarding emails as IM or SMS are not possible.

2.4 Spam Filters

In the past decade, communication through emails has grown exponentially and so has spam. As of 2002, the number of spam messages sent daily was 2.4 billion [7], whereas by mid-2007, it had reached an estimated 100 billion per day [8]. Spam filtering uses a variety of techniques such as rule-based rankings, Bayesian word distribution filters, distributed adaptive blacklists, white list verification filters, Bayesian trigram filters, etc.

Rule-based spam filters like SpamAssassin [9] evaluate a large number of patterns and match the content of an email. SpamAssassin uses techniques such as keyword filters, email header analysis, email content databases, statistical filters, and negative rules. SpamAssassin [10] mainly uses rules and weights (also known as scores). Each rule performs a test on the email and attaches a weight to the email. After an email is processed by all the rules, the final weight, which is an addition of individual weights, is compared to the threshold weight. If the score exceeds the value of the threshold, the email is tagged as spam. This process of assigning weights to the email, comparing to a threshold and tagging has influenced the PMA design. SpamAssassin also lets its users configure their email delivery system. Through this, the users can decide whether the tagged email is spam or not and perform an appropriate action on the email.

Our rule and weights based approach (described more fully below), has a number of unique features, such as the use of a wide variety of extensible context information gathered from a mobile environment, the representation of all user information as context information, the processing of context information from a rule based system, etc. Unlike the earlier systems, the PMA system separately calculates the importance and urgency of an email as two separate indicators which affect priority and therefore to achieve superior email classification results.

3 Approach – Ranking Emails

The primary objective of the Personal Messaging Assistant (PMA) is to effectively prioritize and categorize emails for immediate or delayed delivery, forwarding or filing, by taking context information into consideration. The system decides to deliver specific emails to the user as and when an email (i.e., its content and envelope) becomes important and/or urgent in the current context. It also decides on the appropriate mode of delivery (e.g., SMS, Text-to-Voice on the mobile phone, IM/XMPP) based on the current context of the user. Earlier prototypes of PMA and past work in the field [11] have shown that using a single prioritization metric to achieve this objective was ineffective, as it could not simultaneously account for both urgency and importance of an email.

In our work, we use two separate and independent scales to rate the emails being processed. The first scale represents an email's importance. The second scale represents an email's urgency. The use of these two scales is based on the observation that important emails need not necessarily be urgent and vice-versa. For example, consider: 1) An email from a user's spouse, asking to pick up their son in an hour is both important and urgent; 2) An email from the user's boss, about a deadline that is two weeks away being pushed back another month is important but possibly not urgent; 3) An email from an online auctioning system informing the user about a better bid is urgent but, in most cases, not important; 4) An email invitation from a colleagues is neither urgent or important (see Fig. 1).

Therefore by rating incoming emails using these two different scales, PMA is able to make more flexible decisions than if it were using a single scale. The importance and urgency values for each email is assigned based on the current context of the user and vary with variations in the user's context.

	Unimportant	**Important**
Non-Urgent	Evite for a BBQ.	From manager: Client visit pushed back by another month.
Urgent	Online auction: you were out bid.	Son missed his bus, pick him up from school.

Fig. 1. Classifications of Importance and Urgency (adapted from [12])

3.1 Context Representation

Context is the differentiating factor that makes PMA unique in the way it handles email. We classify context information into two categories: static context and dynamic context. As their names suggest, static context encapsulates the user preferences, and priorities that either remain relatively constant over time or change gradually. On the other hand, dynamic context might change almost every instant. Static context includes information such as user email contacts, mobile phone book contacts, age, social group, email address, names, and preferred emails. Dynamic context is comprised of information such as user's location and activity that are communicated at regular intervals to the server from the mobile devices. Dynamic context also includes user specific information such as calendar information, topics of interest, people nearby, etc. While static context is mainly gathered from explicit user input and the user's email inbox and managed on the servers, the dynamic context is partly updated from the mobile device at regular intervals and partly fetched from external servers.

For example, to gather location information, we use a combination of exterior and interior locationing. In our prototype PMA, we use an extended Wi-Fi Redpin service [13, 14] that sends indoor location information to the PMA server while outdoor locationing is achieved through a combination of GPS and cell ID information. Fetching context information from external servers is exemplified in the PMA prototype with the use of the Business Meeting Organizer system (also presented at this conference) to retrieve calendar information. Context information received by the PMA system from various sources is stored in the context engine of the system.

The dynamic context information and the static context is represented within the PMA system in a context data object with fixed fields for location, activity and user's email addresses and an extendable set of tag-value fields for representing additional context information. The tag is a textual representation of the type of the context data being represented and the value section contains the value of the context data type. Fixed fields for location, activity and user email addresses were created in the data object as they always represent a meaningful value.

The PMA system stores both types of context (static and dynamic) using the same set of extensible tag-value pairs. By storing both static context (which is generally considered as user information [15]) and dynamic context (which is regarded in most context-aware systems [16] as the only form of context information) the same way allows the PMA rule systems to use a uniform method to access this information. We similarly store user preferences.

The exact context information represented in a context data object at any given moment varies with the context of the user at that point in time. For example – 1) when the user is in a meeting the context data object holds information as to the topic of the meeting, the chair of the meeting, etc. (gathered from the user's calendar), 2) when the user is driving to work the context data object holds information regarding the estimated time of arrival (gathered from the GPS system).

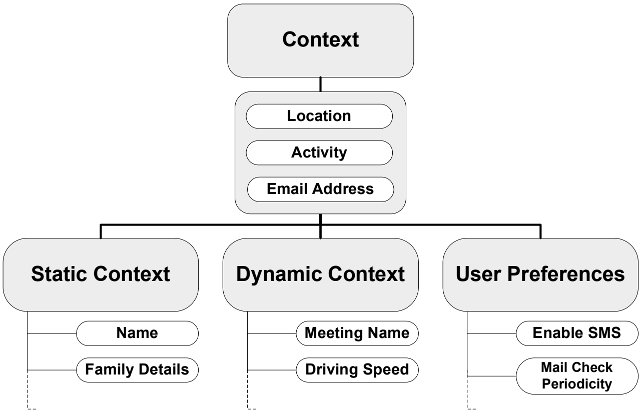

Fig. 2. Types of Context

3.2 Architecture

The PMA architecture consists of five main components, shown in Fig. 3. They are the email preprocessor, the context-generator, the importance processor, the urgency processor and the delivery agent.

PMA processes emails using rules. Furthermore, to make the rule system simpler and more adaptable, the system uses a concept of buckets to classify and prioritize emails. We use the main components of context (currently location and activity) as a tuple to select a bucket into which the emails are placed and then a set of rules specific to that bucket are invoked to process the emails. The rules themselves take various additional aspects of context into consideration. For cases where a location-activity tuple of the user's current context does not have a defined bucket for the rules, a generic bucket is used. The five components are:

- *Context Generator* – retrieves context information from various sources (location information from the user's mobile phone, user's schedule from his online calendar, etc.), and converts the raw context data into the PMA compatible representation described above.
- *Email Preprocessor* – preprocesses emails retrieved from the user's mailbox. Preprocessing involves the removal of non-textual components (images, video clips, etc.) embedded in the emails, and stemming the words in the textual part of the email's body and the email's subject. The preprocessor creates two separate stemmed word lists, one each for the subject and the body of the email which also includes the frequency of each stemmed word.
- *Importance Processor* – selects the appropriate rule-set for this bucket of emails to calculate a numerical value denoting the importance of each email.
- *Urgency Processor* – selects an appropriate rule-set to calculate a numerical value denoting the urgency value of each email.
- *Delivery Agent* – selects appropriate rule-sets for the email buckets for the current location and activity contexts. The delivery rules consider the calculated importance and urgency values and the user's current context to decide on which mails

to deliver to the user immediately, in which order and by which means. The delivery agent can also decide to forward select emails to an address in a predefined set of addresses (belonging to a peer, colleague, family member, etc.). The delivery agent can sort the emails in the inbox into separate folders that allow the user to quickly ascertain which emails have a higher priority. The delivery agent can also keep track of all the delivered emails so as to avoid "re-deliveries". This component is designed in such a way so as to allow fast and easy extensions to delivery methods and sorting options (e.g., add a desktop notification program as a new delivery target). This agent can be customized by the user to allow finer grained control over the entire system.

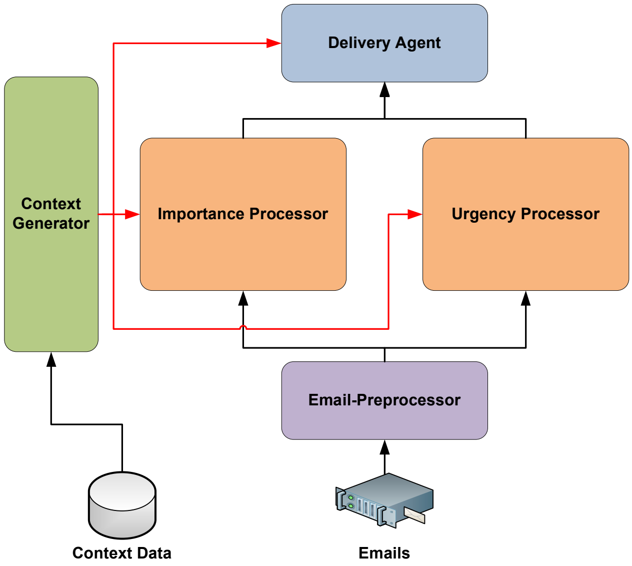

Fig. 3. Architectural overview of the PMA system

3.3 Rule System

The rule systems used in the Importance Processor, Urgency Processor and the Delivery Agent are written using the JESS rule engine for Java [17]. In the current implementation of the rules, only forward-chaining rules are used.

As the rule systems are not able to handle context information directly (i.e., – context data in its raw form is cumbersome to match against features of an email), the Java modules containing the rule engines convert context values to appropriate JESS facts for the rule engine.

The JESS rule engine matches facts to select the rules to be fired. JESS ignores rules which contain facts that have not been defined. This allows the PMA designer to dynamically add or remove (especially remove) the extensible context fields from its context representation without the need to alter the rule-sets.

3.4 Email Content Recognition

PMA uses three different techniques of content analysis to classify an email. The first type is to match the words in the subject and body of the email with lists of words (keywords) denoting varying levels of importance and urgency. The lists in which words occur add or subtract from a weight denoting the email's importance or urgency.

The second type is to compare words occurring the subject and body of an email with dynamic context information. This allows the system to identify emails which become relevant due to the user's current context (e.g., – when the user is heading to a meeting and an email related to the subject of the meeting arrives, the system is able to assign a higher importance and urgency to it).

The third type is to compare the contents and other metadata of an email with the user's static context information. The exact comparisons made vary depending on the bucket holding the email, for example the system may check if the email contains a reference to a user's family member, or it may check if the mail is addressed to the user rather than being a carbon-copy or a blind-carbon-copy.

3.5 Stemming and Lexical Frequency Algorithm

To allow PMA to recognize all equivalent forms of a word in an email subject line or body all textual input from emails to the PMA are stemmed, using the Porter Stemmer Algorithm [18]. The implementation of the algorithm processes a body of text and returns the stemmed words list with only a single instance of each stemmed word and a count of the number of occurrences of the word. For example, the words "work", "worked", "works" and "working" are stemmed to the word "work", whereas the word "worker" is stemmed to the word "worker".

3.6 Structure of the Email Rule System

Each of the components that use the JESS rule system (i.e., – the importance processor, the urgency processor and the delivery agent), employ a flat-rule-architecture.

The flat rule architecture consists of an arbitrary number of rules, each of which performs a single content recognition operation (belonging to one of the three types of content recognition categories described above). In the urgency and importance processors, the rules increment, decrement or scale the importance/urgency values, maintained by the PMA, for each email. The delivery agent employs a simple set of threshold rules (acting on the urgency and importance values) to determine the action for each email in the current context.

An alternative to this method, used in previous iterations of PMA, is to use a layered model for rules. In this type of architecture the firing of one or more rules in a lower layer creates facts that trigger a rule on a higher layer. The current PMA rule systems moved away from this type of implementation for two main reasons. I.e., – 1) to keep the rule system simple enough to allow ordinary users to edit the rules themselves 2) in a flat rule architecture it is easier to implement mechanisms to provide feedback to the user to explain why decisions were made within the system (to improve user confidence [19]).

The customization of the rule systems in the importance processor and the urgency processor can be performed in three different ways. The first involves the addition of new rules to match content of the emails with one instance of the keywords, the static context or the dynamic context. The second is to vary the impact of each rule on the final outcome of deciding whether an email is important/urgent or not. The impact of each rule on the overall decision is adjusted by changing the amount by which the rule adjusts the importance/urgency value of an email when the rule is triggered. The third is to alter the list of keywords understood by the PMA system. This could be done either by the addition or deletion of keywords or by changing the category to which a particular word belongs.

Shown below are three separate rules in the importance processor of the PMA when the user's context tuple is outdoor-driving. The first rule, "driving-rule-1", checks if an email is from an important person (important persons are a group maintained by the context generator); if so the importance of that email increases by 50 (this value can be customized). The second rule, "driving-rule-2", checks if an email was received as a carbon copy; if so the importance of that email is reduced, by 10. The third rule, "driving-rule-3", checks the subject section of an email for words defined as high importance keywords; if any are found in the subject line of an email the importance of the email is increased for each occurrence, by 20.

Examples of three simple rules in the importance processor

```
(defrule driving-rule-1
“Increase email importance if from an 'Important Person'"
 (declare ( no-loop TRUE))
 (and
  (context (othertags $?before GROUP_06_IMP_PPL$?after))
  (email (messageID ?message_id) (from ?from))
  (test (isSubstring ?from ( implode$ (first$ ?after)))))
=>
 (incrementImportancePacketValue ?message_id 50 ))

(defrule driving-rule-2
"Reduce email importance if email is received as a CC"
 (declare ( no-loop TRUE))
 (and
  (context (users_email_addrs $? ?owners_addr $?))
  (email (messageID ?message_id) (cc $? ?owners_addr$?)))
=>
 (incrementImportancePacketValue ?message_id -10 ))

( defrule driving-rule-3
"Increase email importance if highly-important keywords
found in subject"
 (and
  (email (messageID ?id)(subject $? ?h_imp_wrd $?))
  (subject_keywords (h_imp_wrds $? ?hgh_imp_wrd $?)))
=>
 (incrementImportancePacketValue ?id 20 ))
```

Shown below is part of the definition of the Context class, which contains representations of context within the PMA system. The class contains a timestamp field, a location field, an activity field, a field for the user's email address(es), and an arraylist is used to hold the extensible tag-value pairs of context

Part of the context representation class in Java

```
public class Context {
      Date        date_current;
      String      location, activity, owners_email_addrs;
      ArrayList<ContextTag>   additional_tags;
      .
      .
}
```

4 Testing and Results

4.1 Baseline

Initially statistical data from several user mailboxes was collected and analyzed to observe how many emails were received each day, and the number of emails that were left unread. Emails in the main inbox were counted separately from emails filed manually or automatically filed in folders or the Trash folder. The statistical data regarding the amount of emails deleted by users could not be accurately discovered, since the Trash folder is periodically cleared by the email system. Therefore the baseline is an underestimate of the actual complexity. Data was collected for a 120 day period except for Trash folder data which was collected for a 30 day period. Table 1 shows these results.

These results and previous email usage research [1] indicate users receiving high quantities of emails tend to leave larger percentages of emails unread. Based on these statistics, the PMA system was designed with the users receiving in excess of 40 emails per day in mind.

Table 1. Summary per-day baseline email statistics

	Received	Deleted	Filed	Unread
Mailbox 1	79.4	0	59.9	12.3
Mailbox 2	38.1	3.6	23	3.1
Mailbox 3	14.3	4.1	4.5	1
Mailbox 4	12.3	0.2	0.1	1.8
Mailbox 5	10.5	0	8.4	0
Mailbox 6	175.7	44.7	56.6	77.6

4.2 Effectiveness Testing

The first stage of testing was directed at discovering the effectiveness of PMA, by executing it in three different configurations.

The three configurations are: 1) as is without any customization of the system (PMA-a), 2) with customizations done using separate email inboxes as data sources for the customization (PMA-b), 3) using the target email inboxes as data sources for customization, in a 4-fold cross validation (PMA-c). In all three configurations, the static component of the user's context information (e.g. - user's name, email addresses) were provided to the PMA system. In the customized configurations the keyword collection used by the system was increased using a data source (configuration 2 - the data source was a separate inbox, configuration 3 - the data source was a section of the inbox used in the test).

This stage of testing was carried out on three users' inboxes, each with approximately 75 emails. The context was generated synthetically (for repeatability and time considerations) while the test was being carried out. Classifications done by the PMA system was checked for correctness against manual classification performed by the owner of the inbox, for each of the context scenarios.

The results of the importance prioritization in this stage of testing are shown in Fig. 4 along with the sorting provided by Gmail Labels and Gmail Rules and random assignment as benchmarks. The Gmail Rules and Labels were created manually by the PMA team analyzing 20% of the mail box as a guide. Approximately 25 rules were created.

The results of the urgency prioritization in this stage of testing are shown in Fig. 5 with random urgency prioritization as a benchmark. In each test the results were validated against manual classification of the entire inbox by the inbox owner, separately for importance and urgency for a series of contexts.

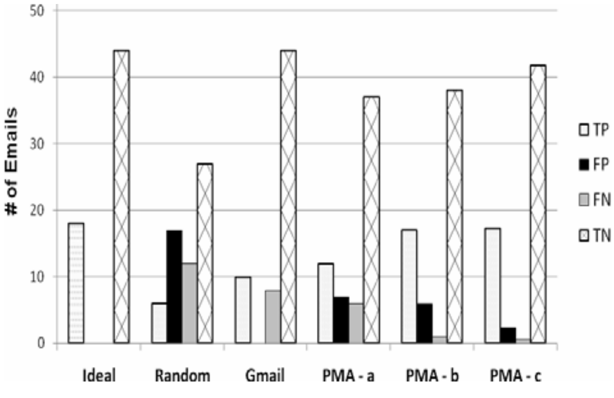

Fig. 4. Importance Prioritization

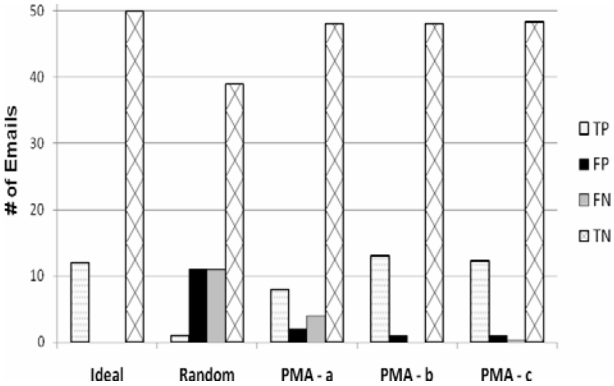

Fig. 5. Urgency Prioritization

In both figures, true-positives (i.e., – correct classifications as important and urgent respectively) are displayed in the left most (speckled) columns, false-positives (i.e., – incorrect classifications as important and urgent respectively) are displayed in the second (black) columns, false-negatives (i.e., – incorrect classifications as unimportant and non-urgent respectively) are displayed in the third (gray) columns and true-negatives (i.e., – correct classifications as unimportant and non-urgent respectively) are displayed in the fourth, right most (diagonal-pattern) columns.

The importance prioritization data shows that Gmail labels perform slightly better than the PMA system (PMA - c) at labeling unimportant emails (depicted by the high true-negative count), yet the high false-negative count shows that Gmail would classify a large number of emails as unimportant where as the PMA system tends to be more cautious and reduce the false-negatives to a minimum while allowing a low number of false positives. The reduction in false positives is approximately 30x over the Gmail system. The PMA results are advantages on the premise that it is better to receive some un-important mails along with the important ones, rather than missing some of the important mails.

The urgency classification data shows that the PMA system is very adept at urgency classification even in an un-customized state, while the effectiveness is increased (high number of true-positives/negatives and extremely low number of false-positives/negatives) when the PMA system is customized.

Table 2. Summary of precision and recall of importance classification

	Random	Gmail	PMA - c
Recall	33.3%	55.6%	96.3%
Precision	26.1%	98.2%	88.2%

Table 3. Summary of precision and recall of urgency classification

	Random	Gmail	PMA - c
Recall	8.3%	N/A	94.8%
Precision	8.3%	N/A	92.6%

4.3 Variation of Effectiveness with Customization

The second stage of testing was directed at discovering the variation of effectiveness of PMA's sorting with the increase of customization. This first level of simple customization of PMA was done solely by increasing the set of keywords understood by the system. While simple, it is quite powerful. The basic keyword set of the PMA system presently consists of approximately 125 keywords, which were manually selected by the research team. The tests were carried out on a constant set of inboxes while the customization was increased by roughly adding 15 keywords between each test. The keywords used in increasing customization were selected randomly from the keyword bank created manually by the research team.

Fig. 6 show the variation of importance classification accuracy with the number of keywords defined. The solid line in the graph shows the variation of true-positives (TP) a.k.a. emails properly classified as important, while the dashed line in the figure plots the false-positives (FP, emails improperly classified as important). Fig. 7 plots the accuracy of unimportance classification with the number of keywords defined. In this case the solid line represents the true-negatives (TN, i.e., – emails correctly classified as un-important) and the dash line represents emails incorrectly classified as unimportant (false-negatives – FN).

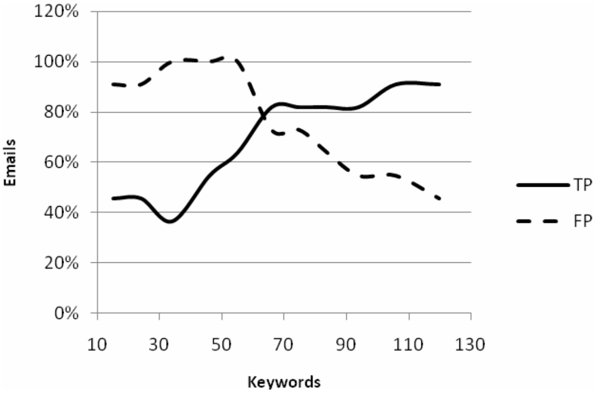

Fig. 6. Variation of importance classification accuracy with the number of keywords

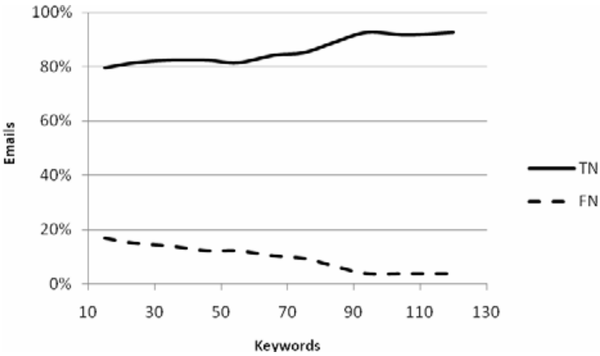

Fig. 7. Variation of unimportance classification accuracy with the number of keywords

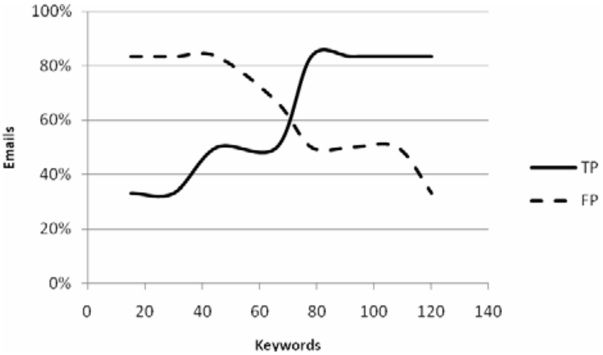

Fig. 8. Variation of urgency classification accuracy with the number of keywords

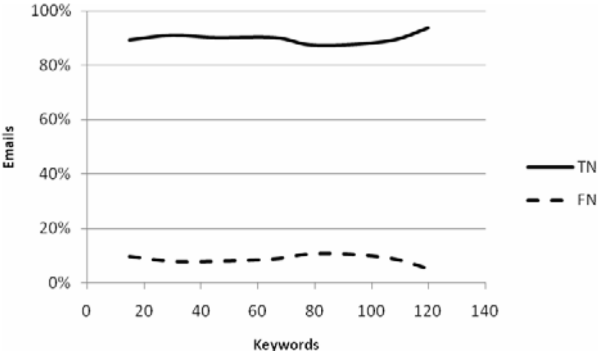

Fig. 9. Variation of non-urgency classification accuracy with the number of keywords

Fig. 8 show the variation of urgency classification accuracy with the number of keywords defined. Fig. 9 plots the accuracy of non-urgency classification with the number of keywords defined.

From the data depicted in figures 6, 7, 8 & 9 it is clear that the accuracy of importance and urgency ratings improve with the increase in keywords.

4.4 PMA Action Sampling

The third stage of testing was to evaluate the percentages of various actions taken on emails (i.e., SMSed to the user, filed for later viewing, forwarded to a peer) and the effectiveness of these actions in different context situations. The PMA makes these decisions based on the delivery agent rules. For testing, the PMA system was customized and used on an inbox with a 75 email sample. The results of the test are shown in figure 15, where the actions taken by PMA for each location-activity pair is represented by the column on the left with the distribution of ideal actions represented to the right (note: 50 mails on which no actions were taken have been omitted from the graph).

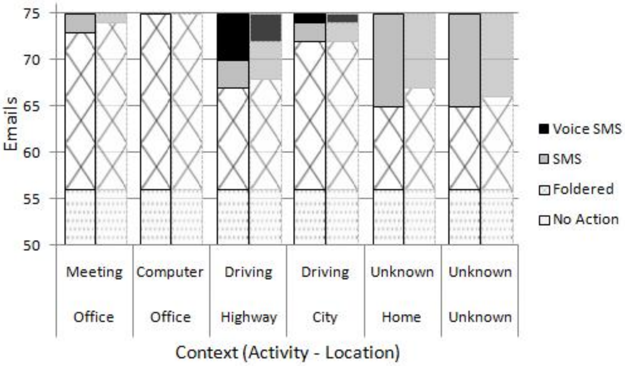

Fig. 10. Variation of actions taken on emails with change in context

The action sampling data shows that SMS as a delivery method is given less priority when the user is deemed to be in a less disturb-able state, as can be seen by the number of SMS sent to the user when he is in a meeting or driving through busy streets within the city. When the user is deemed to be in state where operation of the mobile phone might be difficult, i.e., driving, the PMA decides to use text-to-voice-SMS with the highest priority emails while some of the other emails are delivered (in summary form) as standard SMS. In cases where the PMA infers that the user has ready access to his inbox, PMA does not send SMS but rather files all high priority emails for faster viewing. These decisions are made by a set of rules that take not only the emails content but also the user's context into consideration. The context used in this test was generated synthetically for repeatability and time considerations.

5 Future Work

Performance, generalization and personalization are three dimensions that could be improved in the future versions of the system, which is the major focus for our future work.

The performance of the system is two dimensional. It can be improved in terms of scanning email content and learning the user context information. Advanced machine learning schemes could be used to automate the learning of keywords from user feedback; Naïve-Bayesian methods [20] are under consideration for this purpose. Also, during the initial iterations of the development of PMA, the system run-time was not a major concern, as more emphasis was placed on using context information for email sorting. Going forward, system performance could be improved to consume less system time on larger mail boxes. Lexical analysis or thesaurus based canonic word form analysis on email content to replace the word stemming method used in the current prototype is under consideration.

The next steps to generalize the PMA are to make the system work with other email client accounts like Yahoo! mail and Hotmail and adding handling support for additional message types like SMS, IM, RSS, HTTP and Voice.

Personalization of the system is another area that would be addressed in future research. This includes the creation of a user interface to allow users to create/edit custom rules. Also planned is a user interface on the mobile device to allow user feedback regarding actions taken on emails. Determining the costs of data transfer and processing on the mobile device in terms of power and bandwidth are areas of research under consideration for future work.

We also plan to conduct a larger scale usability test to study the effectiveness of the PMA system in comparison to human-based sorting. The study will also attempt to gather users' requirements for customization.

6 Conclusion

The key goal of PMA is to make email processing for users not just easy but more accurate. PMA considerably reduces time spent by users on filtering emails by sorting and delivering messages that are relevant to the user in his current context. Unlike other email filtering systems that depend solely on email content for filtering and sorting, PMA takes into account the content of emails and the contextual information of the user. Another unique aspect of the system is the consideration of urgency and importance of an email as separate dimensions for classification; thereby PMA is able to integrate a temporal aspect into email sorting. By combining the use of context-information and email content in classification with the idea of separate scales for importance and urgency, PMA is able to intelligently decide whether an email is to be delivered immediately via SMS, deferred for later delivery forwarded to another address, filed, etc.

It is scalable for all inbox sizes and types and offers better performance in terms of identifying email priorities. It could be easily personalized to suit the requirements of any user for better accuracy. The system is highly efficient over continuous usage compared to discontinuous usage.

This system saves ample time for users struggling with email flooding, by filtering emails and decision making on behalf of the user. It gives better performance in terms of filtering emails compared to existing rule based systems. It goes beyond simple email filtering to integrate context-awareness, in an extendable manner, to perform advanced email management.

Acknowledgements

This research was supported by a grant from Nokia Research Center and by CyLab at Carnegie Mellon under grant DAAD19-02-1-0389 from the Army Research Office. The views and conclusions contained here are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either express or implied, of ARO, CMU, or the U.S. Government or any of its agencies.

We also acknowledge the efforts of Patricia Collins, Anind K. Dey, Pei Zhang and the anonymous referees for their valuable feedback and suggestions on earlier drafts of this paper.

References

1. Dabbish, L., Kraut, R., Fussell, S., Kiesler, S.: Understanding Email Use: Predicting Action on a Message. In: Proceedings of the SIGCHI conference on Human factors in computing systems, pp. 691–701. ACM, New York (2005)
2. Whittaker, S., Sidner, C.: Email Overload: Exploring Personal Information Management of Email. In: Proceedings of the SIGCHI conference on Human factors in computing systems, pp. 276–283. ACM, New York (1996)
3. Bergman, R., Griss, M., Staelin, C.: A Personal Email Assistant. Technical report, HPL-2002-236, ACM HP Laboratories, Palo Alto (2002)
4. Griss, M., Letsinger, R., Cowan, D., Sayers, C., VanHilst, M., Kessler, R.R.: CoolAgent: Intelligent Digital Assistants for Mobile Professionals - Phase 1 Retrospective. Technical report, HPL-2002-55(R1), HP Laboratories, Palo Alto (2002)
5. Barton, J., Kindberg, T.: The CoolTown User Experience. Technical report, HPL-2001-22, ACM HP Laboratories, Palo Alto (2001)
6. Yin, M., Griss, M.: SCATEAgent: Context-Aware Software Agents for Multi-Modal Travel. In: Applications of Agent Technology in Traffic and Transportation, pp. 69–84. Birkhäuser, Basel (2004)
7. The Big Business of Fighting Spam, `http://www.ecommercetimes.com/story/20702.html`
8. Anti Spam SpamUnit – Spam Statistics, `http://www.spamunit.com/spam-statistics`
9. Seewald, A.K.: Combining Bayesian and Rule Score Learning: Automated Tuning for SpamAssassin. Intelligent Data Analysis. Technical report, TR-2004-11 Austrian Research Institute for Artificial Intelligence, Vienna, Austria (2004)
10. McDonald, A.: SpamAssassin: A Practical Guide to Integration and Configuration. Packt Publishing, Birmingham (2004)
11. Kiritchenko, S., Matwin, S.: Email Classification with Co-Training. In: Proceedings of the 2001 Conference of the Centre for Advanced Studies on Collaborative research, pp. 8–18. IBM Press, Indianapolis (2001)

12. Covey, S.R.: The Seven Habits of Highly Effective People. Free Press, New York (1989)
13. Bolliger, P.: Redpin - adaptive, zero-configuration indoor localization through user collaboration. In: Proceedings of the first ACM international workshop on Mobile entity localization and tracking in GPS-less environments, pp. 55–60. ACM, New York (2008)
14. Lin, H., Zhang, Y., Griss, M., Landa, I.: WASP: An Enhanced Indoor Locationing Algorithm for a Congested Wi-Fi Environment. Technical report, MRC-TR-2009-04, Carnegie Mellon Silicon Valley, Moffett Field (2009)
15. Carrizo, C., Hatalkar, A., Memmott, L., Wood, M.: Design of a Context Aware Computing Engine. In: Proceedings of the IET 4th International Conference on Intelligent Environments, pp. 1–4. IEEE Press, New York (2008)
16. Salber, D., Dey, A.K., Abowd, G.D.: The Context Toolkit: Aiding the Development of Context-Enabled Applications. In: Proceedings of the SIGCHI conference on Human factors in computing systems, pp. 434–441. ACM, New York (1999)
17. Friedman-Hill, E.: Jess in Action: Java Rule-based Systems. Manning Publications, Greenwich (2003)
18. van Rijsbergen, C.J., Robertson, S.E., Porter, M.F.: New Models in Probabilistic Information Retrieval. Technical report, no. 5587, British Library Research and Development (1980)
19. Li, B.Y., Dey, A.K., Avrahami, D.: Why and Why Not Explanations Improve the Intelligibility of Context-Aware Intelligent Systems. In: Proceedings of the 27th international conference on Human factors in computing systems, pp. 2119–2128. ACM, New York (2009)
20. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques, 2nd edn. Morgan Kaufmann, San Francisco (2005)