# E-Mail on the Move: Categorization, Filtering, and Alerting on Mobile Devices with the ifMail Prototype

Marco Cignini, Stefano Mizzaro, Carlo Tasso, and Andrea Virgili

Department of Mathematics and Computer Science,
University of Udine
Via delle Scienze, 206 – Loc. Rizzi – Udine – 33100 Italy
{cignini,mizzaro,tasso,virgili}@dimi.uniud.it

**Abstract.** We propose an integrated approach to email categorization, filtering, and alerting on mobile devices. After a general introduction to the problem, we present the ifMail prototype, capable of: categorize incoming email messages into pre-defined categories; filter and rank the categorized messages according to their importance; and alert the user on mobile devices when important messages are waiting to be read. The second part of the paper describes an extended evaluation of the ifMail prototype, whose results show the high effectiveness levels reached by the system.

## 1 Introduction

Information overload is the main problem for information access users: we are overwhelmed by too much information when we browse the Web, when we analyze the results of a search engine, when we use a directory, when we read the messages in a forum or in a newsgroup, and when we use electronic mail. Electronic mail, historically one of the first services made available by the Internet to the large public audience, is today one of the major activities of Internet users. All of us rely on email as one of the primary communication methods, both at work and at home: email has, at least partially, supplanted paper mail, messages, and telephone conversations.

*Email overload* is an important facet of information overload: the average user receives dozens of messages per day, and the trend is not slowing down at all [32]; some of us are lucky and receive a manageable number of email messages per day, whereas others are completely overwhelmed; unsolicited email, usually called spam or junk-mail, is constantly and worryingly increasing.

Usage of email is a highly personalized activity, and people use email in amazingly different ways [14]. People read emails with different strategies: *archivers* choose strategies that allow them to read everything and not miss anything important, and *prioritizers* want to limit the time spent on email reading to switch to "real work" [11]. Accordingly to Whittaker and Sidner [33], people can be divided into *no filers* (that keep all the messages in their inbox), *frequent filers* (that constantly clean up their inbox), and *spring cleaners* (that clean up their inbox once every few months).

Also, email software tools (Eudora, Outlook, Mozilla, to name just a few) are used not only in the standard ways foreseen by email tools designers, i.e., for reading and answering messages, but also in more "perverted" ways. We refer here to archiving,

F. Crestani et al. (Eds.): Mobile and Ubiquitous Info. Access Ws 2003, LNCS 2954, pp. 107–123, 2004.

managing a personal agenda or serving as a reminder tool: people send mail to themselves as a reminder; people use the inbox message list as an agenda; people use email for task management and delegation; people hit reply for avoiding to type in a long list of addresses; people archive a whole message when the attachment is an important document; people use email as a file transfer mean; and so on. This creative use of email has generated another meaning for the "email overload" expression [32], i.e., the overloading of uses of this tool, and because of this phenomenon, email has been named a serial-killer application [10].

In this scenario, advanced tools for email processing are desperately needed: threading, categorizing, archiving, filtering, alerting, and perhaps more. Today's email clients provide these functions in a rather limited way. Mail tools allow to view the messages sorted by date, by thread, by sender, etc. Users can manually categorize the messages, usually by drag-and-drop in one of a hierarchy of folders. A priority flag can be manually attached to a message by the sender, and shown to the receiver by the mail client. Filters based on pattern matching rules on (mainly) the structured part of messages (i.e., subject, sender, date, priority, size, etc.) can be manually defined by the user to automatically move the received messages in the appropriate folder (and to execute other operations on the message). Automatic anti-spam filters, to filter out spam exploiting some learning techniques, are common in many mail tools. All email tools can notify the user sitting in front of his/her desktop that new mail has arrived by visual and/or sound messages.

These activities are both time consuming and rather ineffective: manually defining a filter and managing a set of several filters puts a higher cognitive load to a user engaged in other activities and, often, the decision whether a message is interesting, junk, belonging to a certain topical category, and so on cannot be taken only on the basis of the structured part of the message but it has to be taken also on the basis of message body, attachment, meaning, and even context (i.e., the thread to which the message belongs, the current situation in which the user is, and so on). Also, alerting is rather neglected: having only a visual and/or acoustical "You have new mail" notification on our desktop is a rather poor way of communication, that ignores both the cognitive situation of the user, like his/her current task or degree of attention, and features of the message like its urgency, the sender, the topic, and so on.

The coming of portable devices (cell phones, PDAs, pagers, and so on), that are enabled to various network connection modes (GSM, GPRS, UMTS, Wi-Fi, Bluetooth, etc.), is a new and important variable to add in the above sketched scenario. There are several issues that need to be addressed. The new environment implies both limitations to be taken into account and opportunities to be exploited; therefore, simply replicating the non-mobile approach in the mobile world would lead to far from ideal solutions. For instance, using a mobile device to access one own email inbox via standard protocols like POP or IMAP is an unsatisfying solution that neglects both the always-on modality of a user empowered with a mobile device, and the cost usually implied by data transmission on a wireless connection. The usually complex user interfaces of mail tools cannot be replicated on small-screen devices, so it is much more difficult to have ease of reading and user's feedback (e.g., explicit feedback of relevance, categorization, importance, and urgency of a message is likely to be replaced by more implicit kinds of feedback, perhaps exploiting the time that a message waits in the "unread" status). The interaction modes requiring continuous attention (e.g., drag-and-drop), that are common for desktop-based tools, are not adequate for devices used out there in the real world, with several sources of distraction.

Notifications could and should be delivered on the nowadays widely available smaller and portable devices with the most appropriate modality (WAP-push, SMS, etc.). Notifications should be done depending on features of the received messages like their number, their importance, the category they pertain to, and so on. The well known limitations on bandwidth, screen size, and user cognitive load (time, distraction level, and so on) make extremely important to have a *selective* alerting functionality, capable of notifying the user only when really important messages arrive: not only the notification of a spam message would be very unpleasant for the user, but also the notification of a "normal" message when the user is in a particular context (e.g., while driving, or engaged in a meeting, or in an important phone conversation) can be unpleasant as well. The mobile world requires an integrated solution, exploiting categorization, filtering and alerting.

Moreover, in the mobile world, categorizing, filtering, and alerting will have an increased importance, since accessing email by a mobile device is more critical in many respects. People carry with themselves their mobile devices, that are therefore much more intrusive than a standard desktop: the "new mail" sound that might be an acceptable interruption when sitting in front of a desktop computer, is likely to be very annoying while engaged in real-world critical activities.

Turning our attention to more technical issues, we notice that new mail tools and protocols might be designed to allow the user (both as a sender and as a receiver) to specify (manually, semi-automatically, or automatically) the alerting modalities of certain message categories. Complex engineering solutions are needed because the limited storage and computational power available today on the mobile devices, and the bandwidth limitations, suggest a server side based solution, in which most of the computation takes place on the server and the data transmission on the mobile device is limited.

Also, the integration of all the devices that one can use to read his/her own email messages (desktop PC, mobile devices, internet points, etc.) is another interesting, and difficult problem, and reinforces the requirement for server side based solutions. A further kind of integration is that among all the different kinds of messages that the user of a mobile device can receive: besides email-like messages, we have SMS, EMS, and MMS (and perhaps more in the future). The integration of all these message services is a difficult problem as well.

Finally, the increased email access by mobile devices will change the people usage of email: nobody can predict all the range of new "perverted" or "creative" uses that mobile device users could imagine and adopt when mobile email tools will be broadly available (e.g., the sending of email to oneself as a remainder is likely to become much more frequent).

All these issues constitute a research agenda for the years to come, and need to be tackled from an interdisciplinary standpoint: user modeling, information retrieval and filtering, human computer interaction, software engineering, are all disciplines that can contribute to the development of more effective email tools for the mobile and wireless world. In this paper we do not present a final and general solution. Rather, our aim is twofold: (i) to show how to improve and make (at least partially) automatic the tasks of email categorization, filtering, and alerting; and (ii) to show how to integrate these new and more effective tools in the mobile scenario, where people access email while on the move. The paper is structured as follows. In Section 2 we highlight the main issues related to email categorization and filtering. We also survey the literature, briefly describing the relevant work that has been proposed so far. In Section 3 we describe the ifMail prototype, from both conceptual and technical perspectives. In Section 4 an extended experimental evaluation of the effectiveness of our approach is presented. Section 5 closes the papers and sketches future developments.

## 2 Categorization and Filtering of Email Messages

Text categorization (or classification) is the grouping of documents into predefined categories [28]. State-of-the-art classifiers automatically built by means of machine learning techniques show an effectiveness comparable to manually built classifiers.

Email messages are very heterogeneous. Examples of variables that can range over rather wide set of values are: length, language(s) used, importance of the contained information, presence/absence of attachments of various kinds, formal/informal tone, emoticons, jargon. Also structured data contained in the header like date, sender, subject, number of recipients, are bound to wide variations. Given the peculiar nature of email messages, email categorization is a very particular case of general text categorization.

Various approaches, mainly derived from the experiments on generic text categorization, have been applied to email categorization [9]: Cohen [7] uses the RIPPER algorithm; Payne and Edwards [24] compare CN2 (a rule induction algorithm) with IBPL1 (a modified version of K-nearest Neighbor algorithm using memory based reasoning); Rennie [25] exploits naïve Bayes classifiers; Segal and Kephart [29] develop a system for semi-automatic categorization (i.e., the system proposes to the user three alternative folders for each message) based on TF-IFD; Brutlag and Meek [4] compare Linear Support Vector Machine, TF-IDF, and Unigram Language Model, and obtain that no method outperforms the others; McCreath and Kay [14] show how the combination of hand crafted and learnt rules is more effective than either approach working alone. All these approaches show rather similar results, with accuracy (percentage of messages classified in a correct way) around 70%-80%. An even more difficult problem, the clustering of email messages (i.e., given a set of email messages, extract the categories and classify the messages in the found categories), is tackled in [13].

Spam (or junk) email filtering has seen an increasing interest in last years, due to the increasing amount of unsolicited emails: Pantel and Lin [19] and Sahami et al. [27] exploit naïve Bayes classifiers; Adroutsopoulos et al. [1] use a memory-based (or instance-based) approach, implemented as a variant of the K-nearest neighbor (K-nn) algorithm; Carreras et al. [5] rely on the boosting algorithm AdaBoost to find a highly accurate classification rule by combining many weak rules.

Anti-spam filtering has been approached as a separate problem from email categorization, even if, at first glance, it seems just a 2-categories categorization problem. However, anti-spam is an easier problem than categorization not only because it handles just two categories, but also because the two categories are rather well defined (it is rather easy to define spam), clear-cut (it is rather easy to sort out spam from non-spam), and objective (usually, what is spam for one user is spam for everybody). In turn, email categorization is highly subjective: each user can choose rather different criteria for creating the categories (e.g., some users divide messages on the basis of the sender, others on the basis of the topic, others on the basis of their a-priori categorization of their job activity, and so on); the number of categories can vary a lot among users; the categories are sometimes not well defined (users can be very well organized or completely chaotic); and so on. Therefore, it is quite likely that a single fit-for-all email categorizer is not feasible, and that hybrid approaches are needed. Indeed, even if it is difficult to have a definitive comparison between the effectiveness of anti-spam filters and of email categorizers because of the high differences in the collections used, in the number and features of categories, and so on, it is evident that anti-spam filters effectiveness is rather higher (95% precision) than the more general email categorization problem.

The alerting problem is much less studied than email categorization and filtering: further research in terms of notification modalities, prototype implementation and evaluation, and user studies is needed. It seems anyway obvious that only important messages should be notified on mobile devises, to avoid high cognitive loads and distraction on the user. Therefore, an integrated solution, comprising categorizing, filtering and alerting is required.

The evaluation of the effectiveness of an email tool is not simple at all. The most naïve approaches show several limitations. Relying on general test collection like TREC (http://trec.nist.gov/) is not adequate, since the peculiar nature of email makes an email message different from a generic document. Usenet news seem more similar, but again differences do exist: for instance, an email message body usually starts with the name of the recipient, whereas this is obviously less frequent for Usenet messages.

Privacy is also an important issue: since email messages contain private data, few people are willing to make public their messages; perhaps those people will anyway clean some of the more compromising and confidential messages, thus making available only a portion of their message archive, that is not a good sample at all; anyway, people willing to make public their email archives are not a good sample for sure, since people that are more reserved are completely left out; and relying on messages archives of mail lists leads again to a biased sample.

## 3 The ifMail Prototype

At the Udine University we have started to study some of the above described issues and, on the basis of our work in the last 10 years, we have developed the ifMail prototype. ifMail handles, with a content based approach, categorization, filtering of email messages, and alerting on mobile devices. ifMail overall operation is shown in Fig. 1. The messages in the incoming stream are processed to extract the internal representations used in subsequent steps. The internal representation contains term/weight (weight representing the importance of each term) pairs, corresponding to both the structured part and the body of the email message. Categorization is obtained on the basis of a profile attached to each user-defined folder and dynamically updated by means of user's feedback. The profile contains two parts: a frame for the information included in the structured part of email messages, and a semantic network for the conceptual content of the body of messages [16]. The profile is matched with the internal representation of the incoming messages and the message is classified accordingly to its content. The matching takes into account both the structured and unstructured parts of email messages. Filtering, performed by re-using the evaluation made in the categorization phase, singles out the most relevant messages in each folder and alerting takes charge of notifying these messages to the user's mobile device. Our notion of filtering is therefore more general than just anti-spam filtering: ifMail tries to associate to each message a numeric figure representing the importance that the message has for the user.

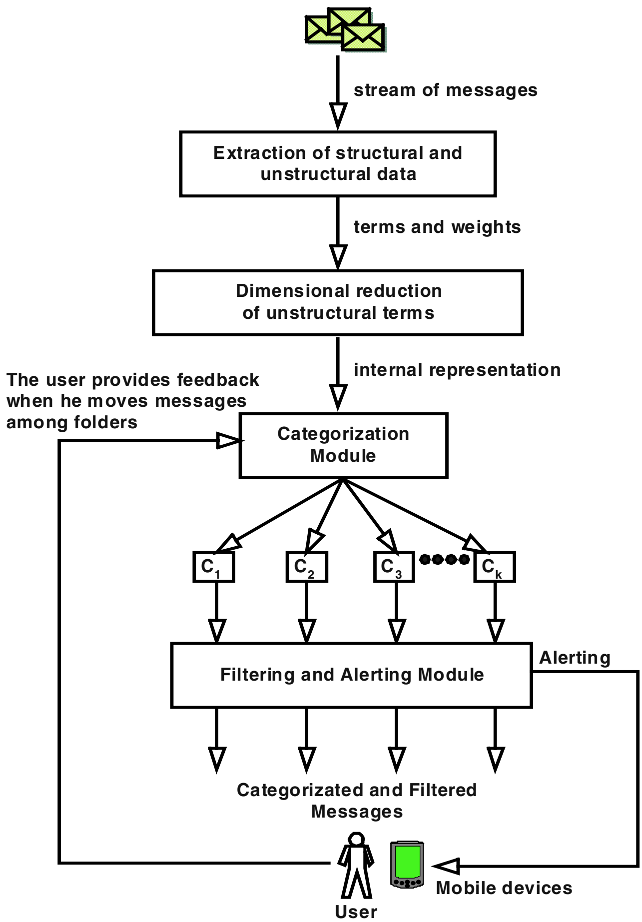


**Fig. 1.** Conceptual model of ifMail operation.

ifMail categorization and filtering are based on the IFT (Information Filtering Tool) system [14,16], capable of profile building, storing, and matching. IFT has been developed on the basis of the UMT (User Modeling Tool) shell [0] and has been applied to a variety of systems and domains, e.g., Web filtering [2], filtering of enterprise documents [30], and filtering of scholarly publications [17]. IFT matches the profile associated to each category with the internal representation of each message and returns a result made up of three values:

1. *Coverage*: the percentage of the most relevant concepts of the profile which are also present in the documents, computed taking into account also their weights.
2. *Match*: a measure of how much the concepts of the profile are present in the document (i.e., they are more or less numerous in the document).
3. *Rank*: a synthetic value (ranging from 0 to 5), which is obtained as a combination of the previous two values.

Categorization is performed on the basis of all three values; filtering is based on Rank score only.

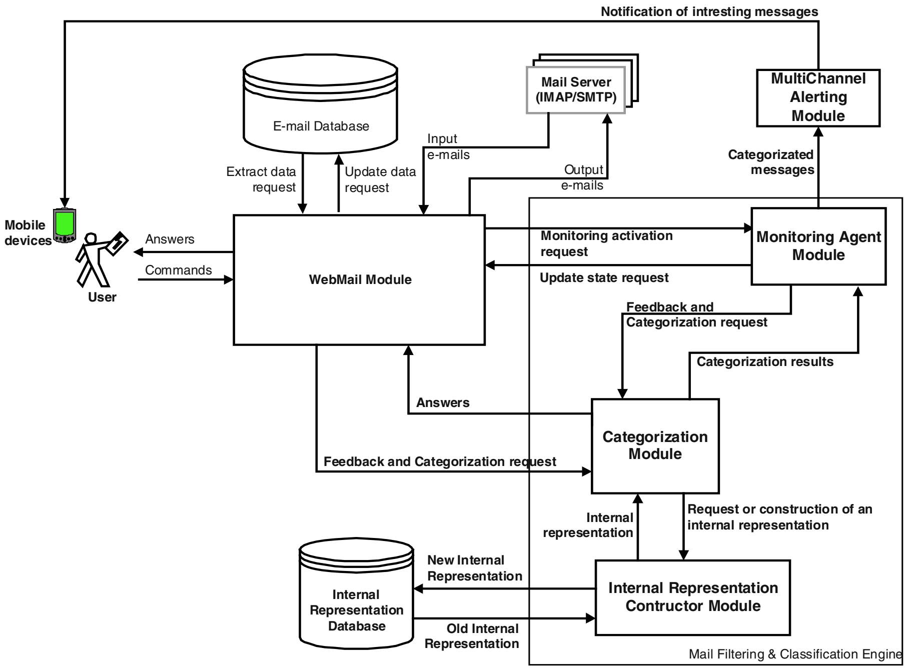


**Fig. 2.** ifMail overall architecture.

Fig. 2 shows the overall architecture of the ifMail system. The main modules are:

- WebMail, that allows the user to access email functionalities via a Web browser. It has been developed specifically for this project in order to connect and integrate categorizing, filtering, and alerting. More specifically, the WebMail module implements the only user interface of the system and it allows the configuration of the innovative services.
- Mail Filtering and Classification Engine, made up by the following three submodules:
  a) Monitoring Agent, that monitors the arrival of new messages and calls the categorization and filtering operations. ifMail supports POP and IMAP servers, and any number of email accounts.
  b) Internal Representation Builder, that parses the text of message subject and body, removes stop words, extracts the stem of the terms, and builds the internal representation of the message, stored in the Internal Representation Database.
  c) Categorization, that executes categorization and handles feedback data. This module contains, and relies on, the IFT submodule: IFT compares the internal representation of the incoming message with each category profile, and modifies the category profile according to user's relevance feedback.

- Multi Channel Alerting, that, on the basis of the categorization results and of user's personalized settings, notifies immediately to the user the most relevant messages via a mobile device.

Fig. 3 shows a snapshot of ifMail Web user interface: a quite standard email interface that allows standard mail management and that provides the commands and visualization items relevant to the new categorization and filtering features. The number of stars associated to each message is given by the Rank score associated to the message.

The PDA screenshots in Fig. 4 show the multi-channel alerting of ifMail: in the screenshot on the top, the notification of the arrival of a new relevant message for the "myWork" category is shown. The user can detect (by the number of stars) the message relevance computed by the system, he can archive the message, read message data like sender and subject, or read the whole message body (screenshot below).

The system has been developed with an XML-based technology to allow a higher flexibility for the presentation layer: multiple interfaces are generated by means of XSLT transformations, that produce the output in the markup language suitable for the requesting device by applying the corresponding style sheet to a common set of XML data. In such a way, the service is accessible by a wide range of devices such as PDAs, smartphones, and cell phones, provided that they comply to the WAP 1.2.1 or 2.0 standards [19, 21, 22, 25].

The interface design has been developed according to some guidelines for information access with mobile devices [5, 7, 11, 20]. The navigation through the pages of the service has been designed considering the physical interface used for the interaction with the device. Moreover, the complexity and extension of every page of the service are adapted to the dimension and capabilities of the display of the mobile devices. From a functional point of view, the interfaces are down-scaled (going from the PC version to the WAP version) to reduce the complexity of the service, considering the limited resources of the devices and the mobile context of use of the service.

## 4 Experimental Evaluation

We have discussed in Section 2 the intrinsic limitations in the evaluation of advanced email tools, and some of the issues that make the evaluation of these tools a difficult task. In order to overcome these limitations, we have designed and carried out an extensive evaluation of the ifMail prototype, taking also into account previous experimental work carried out in recent years in our laboratory. The goal of the experimental activity has been the evaluation of categorization, filtering, and alerting capabilities of ifMail. We have run various simulations on 6 collections of email and newsgroups messages (Table 1). We have used the term "simulation" since the experiments have been performed in a simulated environment in which the typical actions that a user could perform on ifMail can be repeated at will, without engaging (and overloading) real users.

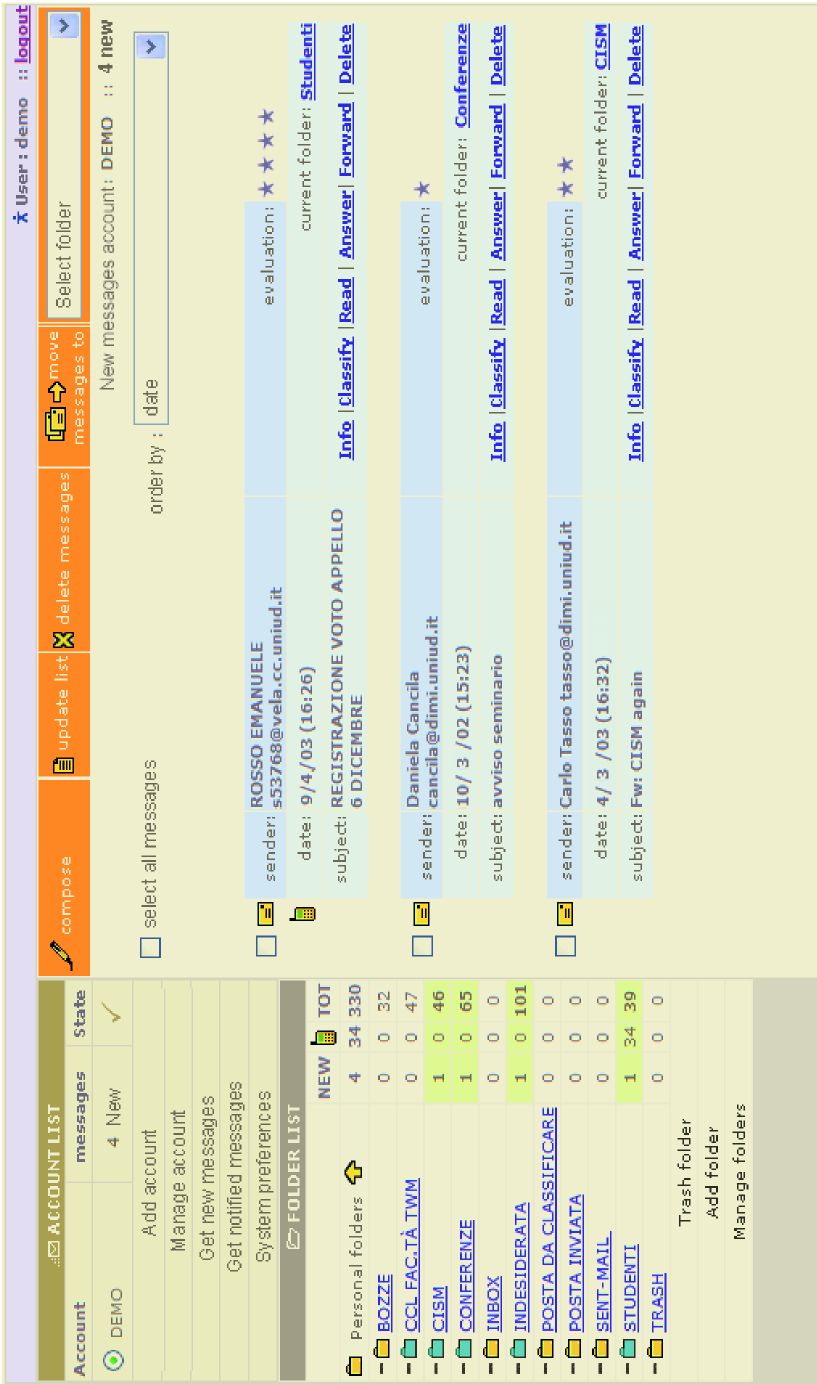


**Fig. 3.** ifMail user interface for Web mail.


**Fig. 4.** ifMail user interface: email reading on a PDA (left) and folders of new categorized messages on an Openwave WAP phone simulator (right).

Obviously, with this approach, we have intentionally not evaluated the usability of the user interface, nor we wanted to claim the effectiveness of our system in absolute terms. On the other hand, given the early development stage of the ifMail prototype, we were interested in evaluating some design decisions and in harvesting an experimental set of real data with a quick, light, and formative evaluation, capable of giving us hints on how to proceed with the development of the system.

Table 1 provides basic data on the six collections of email messages we have exploited: two of them come from real users, and include all the messages received over a period of about 30-40 days. All the messages received over that period were included, and none was eliminated. Both users (one of them is the third author of this paper) defined a set of categories (folders), to be used for evaluating the classification capabilities.

The collections extracted from newsgroups concern a similar number of messages and categories, with the exception of collection F, which is significantly larger and was considered for evaluating whether the results obtained with similar collections (A through E) were maintained in a much heavier situation.

**Table 1.** Email message collections used in the experiments.

| Message kind | Collection | Number of categories | Total number of messages |
|---|---|---|---|
| Personal messages | A | 9 | 540 |
| | B | 7 | 645 |
| Newsgroups messages | C | 7 | 525 |
| | D | 6 | 450 |
| | E | 7 | 540 |
| | F | 16 | 1309 |

We have defined two different modes of operation of ifMail usage:

- Mode *One-by-one*, in which ifMail provides only an advice: the user reading a message is shown a hint on which category(ies) are likely to be the correct destination of that message. By confirming or not confirming on each single message the (automatically) proposed categorization, the user provides relevance feedback, exploited by the system to update the relevant category profiles.
- Mode *Session*, in which ifMail automatically categorizes all the messages received during the current day (we have assumed daily batches of fixed size including 15 messages per day). The user provides relevance feedback only after all these categorizations have been done.

A first set of experiments concerned the comparison of these two modes of operation. The profiles associated to each folder were initially empty, and were incrementally built only through relevance feedback. Table 2 illustrates the average (over all the available collections) of precision, recall, and F1 measure [31, 34], where the results obtained for each category are combined using the micro-average indicator [28].

**Table 2.** Comparison between session mode and one-by-one mode.

| | Session mode | One-by-one mode |
|---|---|---|
| Average Precision | 75% | 79% |
| Average Recall | 72% | 76% |
| Average F1 | 74% | 78% |

First of all, we notice that the values obtained are in the range from 70% to 80%. Other experiments reported in the literature [18, 28] concern the categorization of the Reuters-22713 collection (constituted by 21.450 articles, subdivided into 135 categories) or the Reuters-21578 collection (constituted by 12.902 articles, subdivided into 90 categories): the values obtained for the F1 measure are in the same range between 70% and 80%. We have considered this result as a confirmation of the adequacy of the baseline performance of ifMail. Furthermore, it should be highlighted that the values reported in Table 2 are average values, which include also the initial phases, where errors are most likely to happen: this implies that saturation ('steady state') values can be significantly higher.

Secondly, it can be noticed that precision reaches higher levels than recall. We can interpret this phenomenon in the following way: the number of messages considered (i) is capable of reducing the number of categorization errors, but, on the other hand,

(ii) is not sufficient for building profiles that cover all the concepts included in a category (and some message are not categorized, i.e. not assigned to any category). Finally, one-by-one mode outperforms session mode, reaching almost 80% in all the three considered indicators.

With reference to the same experiment, Fig. 5 shows the evolution (over the sequence of daily sessions and only for collection E) of the F1 measure. Both modes of operation reach values above 80%. The 70% level (conventionally taken as the value indicating the termination of the initial learning phase), is reached earlier in the one-by-one mode. In the long run the two mode of operation reach the same level of performance.

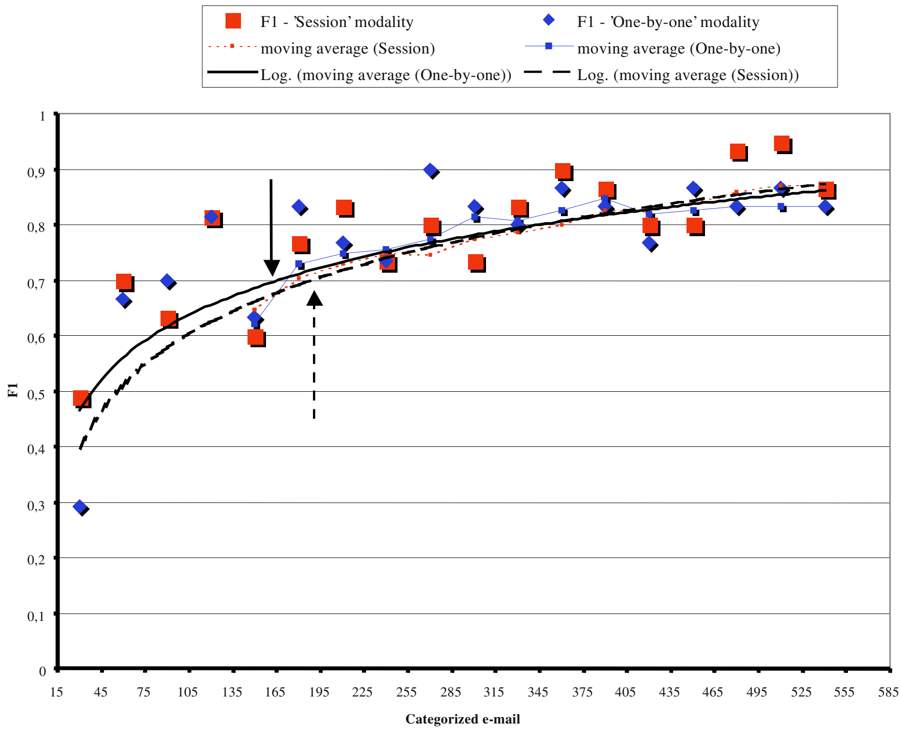


**Fig. 5.** Microaverage F1 in both operation modes for collection E.

Collections A and B, provided by real users, contained a Spam category, defined by the two users in order to collect all the 'not desired' messages (typically unsolicited advertising). In Fig. 6, we report the evolution over time of both precision and recall for the Spam folder of collection B. Precision reaches more than 95% and recall the range 70%-80%: this can be explained by the fact that when a Spam message is received, all the subsequent massages concerning the same topic will be detected, while new Spam topics are not known since never seen before, so they are left in the inbox, i.e., not categorized. This highlights a significant advantage of our content-based approach to Spam detection, in comparisons with standard anti-Spam systems based on an archive of spam messages: our system can detect any new Spam message which concerns topics that previously have been already classified as Spam, independently from other facts (sender or subject already encountered or not).

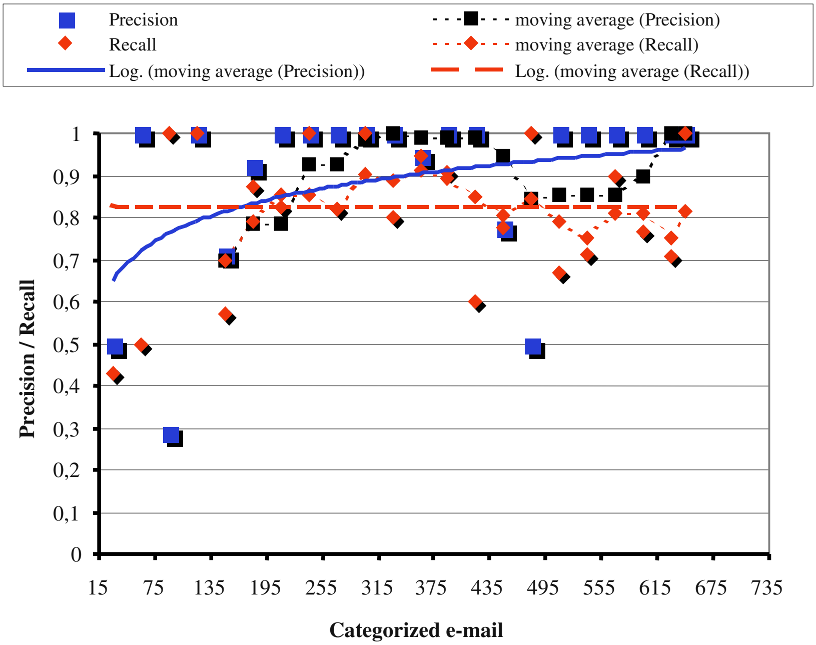


**Fig. 6.** Precision and Recall for the Spam category of collection B.

Another (expected) phenomenon observed in the experimentation concerns the relationship between performance and level of specificity of a category: whenever a category includes a well defined and limited topic, performance in terms of precision and recall is higher, reaching for both indicators the level of 85%. Analogously, for such categories, the learning phase is shorter.

Table 3 illustrates such a situation for some categories with this characteristics.

Other experiments have been focused on the identification of the best threshold to be employed for alerting. We have seen that using only the Rank value (an integer ranging from 1 to 5), precision was maximized (over 80%) and that, by increasing the specific value considered for the threshold, precision was further improved. Fig. 7 shows that the higher the threshold (4 or 5), the steeper is the learning curve, and higher are the precision values obtained (several values saturate at 100%).

Finally, we have computed a measure of the effort required to the user of ifMail, in terms of the number of 'move operations' of a mail message towards its correct folder (category). More specifically, we have considered successive groups of 60 messages (i.e., four days), and we have counted:

- the number of correct system categorization operations (green line on the top part of Fig. 8);
- the number of user moves, i.e., the explicit indication done by the user on a single message, since the system was not able to categorize the message correctly (red line in Fig. 8).

It is interesting to see that, as the user ‘teaches’ to the system how to categorize, the system ‘learns’. After about 70 messages received, the user needs to move about 50% of the messages to their correct folder. After about 300 messages, the system ‘has learned’, and it is able to categorize correctly more than 50 messages out of the incoming 60, with a missed-categorization rate of less than 16%.

**Table 3.** Results for categories with well defined topic.

| Collection | Folder | Precision | Recall | F1 |
|---|---|---|---|---|
| **A** | *News* | 0,91 | 0,83 | 0,87 |
| **B** | *Students and courses* | 0,94 | 0,93 | 0,93 |
| | *Department news* | 0,85 | 0,91 | 0,88 |
| | *Seminars* | 0,86 | 0,91 | 0,88 |
| **C** | *ADSL* | 0,92 | 0,92 | 0,92 |

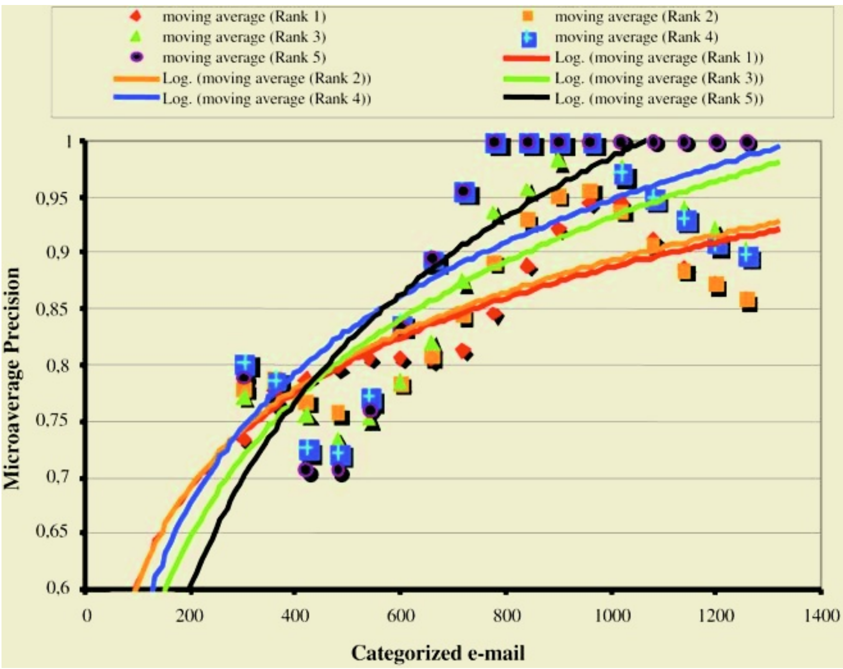


**Fig. 7.** Precision with different values as alerting threshold for collection F.

## 5 Conclusions and Future Work

We have discussed the issues of email categorization, filtering, and alerting. After a general introduction to the problem and a brief literature survey, we have presented the ifMail prototype, capable of: categorize incoming email messages into pre-defined categories; filter and rank the categorized messages according to their importance; and alert the user on mobile devices when important messages are waiting to be read.

We have also performed an extended evaluation of the ifMail prototype. The results show the high effectiveness levels reached by the system.

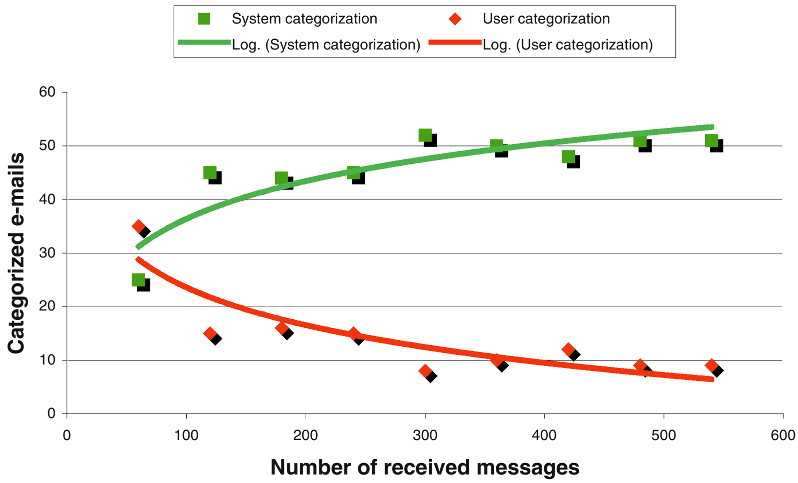


**Fig. 8.** Comparison of the number of user and system categorization actions (session mode).

We will continue this research in various ways. We are currently working at improving the ifMail prototype and we plan a more complete evaluation after these improvements. We intend to deal with privacy issues with a novel approach, by implementing a software capable of analyzing the email archives of users by running on their computers and simulating the behavior of a categorization algorithm. The categorization algorithm results should then be compared with the hand-made categorization and only the comparison results are made public. This software should be open source (to guarantee the privacy) and could be designed as a framework capable of hosting any categorization algorithm conforming to some well defined specifications. To take into account the time characteristics of messages (how long a message has been staying in the inbox, how long it has been in the unread status, for how long the user has not been checking his/her email, how much time the user spent in reading it, or in answering it, and so on) the software should also be capable of monitoring user's activity for a period of time.

## Acknowledgements

We would like to thank Luca Chittaro and Paolo Dal Cin for their help in the design of the user interface on PDAs.

## References

1. I. Androutsopoulos, J. Koutsias, K.V. Chandrinos, G. Paliouras, C.D. Spyropoulos, An Evaluation of Naive Bayesian Anti-Spam Filtering**.** *Proceedings of the Workshop on Machine Learning in the New Information Age, 11th European Conference on Machine Learning (ECML),* pp. 9-17, Barcelona, Spain, 2000.

2. F. A. Asnicar., M. Di Fant, C. Tasso User Model-Based Information Filtering. In M. Lenzerini (Ed.) AI*IA 97: Advances in Artificial Intelligence - *Proceeding of the 5th Congress of the Italian Association for Artificial Intelligence*, Rome, I, September 17-19, 1997, Springer Verlag, Berlin, LNAI 1321, 1997, pp. 242-253.
3. G. Brajnik, C. Tasso, A shell for Developing Non-Monotonic User Modeling System*, International Journal of Human-Computer Studies*, vol.40, pp.31-62, 1994.
4. C. Brutlag, J. Meek, Challenges of the email domain for text classification, In *Proceedings of the Seventeenth International Conference on Machine Learning*, 2000, pp. 103-110.
5. G. Buchanan, M. Jones, H. Thimbleby, S. Farrant, M. Pazzani, *Improving mobile internet usability,* Proceedings 10th WWW Conf., ACM Press, New York, 2001, pp. 673-680.
6. X. Carreras, L. Marquez, Boosting Trees for Anti-Spam Email Filtering *Proceedings of RANLP-01, 4th International Conference on Recent Advances in Natural Language Processing*, Tzigov hark, BG, 2001.
7. L. Chittaro and P. Dal Cin , Evaluating interface Design Choices on WAP Phones: Navigation and Selection, *Personal and Ubiquitous Computing*, 6(4), 237-244, 2002.
8. W. Cohen, Learning Rules that Classify E-Mail, Papers from the AAAI Spring Symposium on Machine Learning in Information Access, 1996, pp. 18-25.
9. E. Crawford, J. Kay, E. McCreath, Automatic Induction of Rules for e-mail classification, *Proceedings of the Sixth Australian Document Computing Symposium*, Coffs Harbour, Australia, Dec. 7, 2001
10. N. Ducheneaut and V. Bellotti, Email as Habitat. *Interactions*, September/October 2001.
11. M. Jones, G. Buchanan, H. Thimbleby *Sorting Out Searching on Small Screen Devices*, In F. Paterno, (Ed.), Proceedings of the 4th International Symposium on Mobile HCI, Pisa, Italy, 2002, LNCS 2411, pp 81-94, Springer.
12. W. Mackay, Diversity in the Use of Electronic Mail: A Preliminary Inquiry. *ACM Transactions on Office Information Systems*, 6(4), 380-397, 1988
13. G. Manco, E. Masciari, M. Ruffolo, A. Tagarelli, Towards An Adaptive Mail Classifier, *Atti dell'Ottavo Convegno AI*IA 2002*, Siena, Italy, 2002, pp. 63.
14. E. McCreath, J. Kay, Iems: Helping Users Manage Email, In P. Brusilowski, A. Corbett, F. de Rosis, *User Modeling 2003 9th Intl. Conf. on User Modeling, UM2003*, Springer Verlag, LNAI 2702, 2003, pp. 263-272
15. M. Minio, C. Tasso, User Modelling for Information Filtering on Internet Services: Exploiting an Extended Version of the UMT Shell, *UM96 Workshop on "User Modeling for Information Filtering on the WWW"*, Kaiula-Kona, Hawaii, USA, January, 2-5, 1996.
16. M. Minio, C. Tasso, IFT: un'Interfaccia Intelligente per il Filtraggio di Informazioni Basato su Modellizzazione d'Utente, *AI*IA Notizie IX(3)*, 21-25, 1996.
17. S. Mizzaro and C. Tasso, Ephemeral and Persistent Personalization in Adaptive Information Access to Scholarly Publications on the Web. In P. De Bra, P. Brusilovsky, R. Conejo (Eds.), *Adaptive Hypermedia and Adaptive Web-Based Systems, Second International Conference AH 2002, LNCS 2347*, pages 306-316, Malaga, 29-31 May 2002. ISBN 3-540-43737-1
18. I. Moulinier, G. Raskinis, J. G. Ganascia, Text Categorization: a Symbolic Approach, *In Proceedings of the Fifth Annual Symposium on Document Analysis and Information Retrieval,* Las Vegas, April 1996, pp. 87-99.
19. Nokia Corporation, *WML to XHTML migration. Version 2.0*, http://www.nokia.com/, 2002.
20. Openwave Systems Inc., *Openwave Usability Guidelines for WAP Applications*, http://developer.openwave.com/support/techlib.html/, 2001.
21. Openwave Systems Inc., *Migrating to WML with GUI Extensions and XHTML Mobile Profile,* http://developer.openwave.com/support/techlib.html/, 2001.
22. Open Mobile Alliance. http://www.wapforum.org/.
23. P. Pantel, D. Lin, Spamcop: A spam classification & organization program, in *Proceedings of AAAI-98 Workshop on Learning for Text Categorization*, pages 95-98, 1998.

24. T. Payne, P. Edwards, Interface agents that learn: An investigation of learning issues in a mail agent interface, *Applied Artificial Intelligence,* Vol. 11, pp. 1-32, 1997.
25. Phone.com Inc., *WML Application Style Guide,* http://www.openwave.com/, 2000.
26. J. D. M. Rennie, ifile: An application of Machine Learning to E-Mail Filtering, *In Proceedings KDD00 Workshop on Text Mining,* Boston, 2000.
27. M. Sahami, S. Dumais, D. Heckerman, E. Horvitz, A bayesian approach to filtering junk e-mail, in *AAAI-98 Workshop on Learning for Text Categorization,* 1998.
28. F. Sebastiani, Machine Learning in Automated Text Categorization, *ACM Computing Surveys*, 34(1), 1-47, 2002.
29. R.B. Segal, J.O. Kephart, Incremental Learning in SwiftFile, *Proceeding of the International Conference on Machine Learning,* IBM, San Francisco, 2000, pp. 863-870.
30. C. Tasso and M. Armellini, Exploiting User Modeling Techniques in Integrated Information Services: The TECHFINDER System. In E. Lamma and P. Mello (Eds.) *Proceedings of the 6th Congress of the Italian Association for Artificial Intelligence*, Bologna, I, September 14-17, 1999, Pitagora Editrice, Bologna, 2000, pp. 519-522.
31. K. Van Rijsbergen, *Information Retrieval*, 2nd ed. Butterworths, London, UK, 1979. http://www.dcs.gla.ac.uk/Keith/pdf.
32. G. Venolia, L. Dabbish, J.J. Cadiz, A. Gupta, Supporting Email Workflow. Microsoft Research Tech Report MSR-TR-2001-88
33. S. Whittaker and C. Sidner, Email Overload: Exploring Personal Information Management of Email. *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI 1996)*, pp. 276-283.
34. Y. Yang and X. Liu, A re-examination of text categorization methods, *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Berkley, CA, USA, August 15-19, pp. 42-49, 1999.