E-mail Categorization, Filtering, and Alerting on Mobile Devices: The ifMail Prototype and its Experimental Evaluation

Marco Cignini, Stefano Mizzaro, and Carlo Tasso

Department of Mathematics and Computer Science, University of Udine
Via delle Scienze, 206, Udine, Italy
{cignini,mizzaro,tasso}@dimi.uniud.it
http://www.dimi.uniud.it/~{cignini,mizzaro,tasso}

Abstract. We propose an integrated approach to email categorization, filtering, and alerting. 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

Email overload is an important facet of information overload. 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 method, both at work and at home: email has, at least partially, supplanted paper mail, messages, and telephone conversations. The main problem is that the average user receives dozens of messages per day, and the trend is not slowing down at all [23].

Moreover, 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, 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. Because of this creative use of email, another meaning for the "email overload" expression (the overloading of uses of this tool) has arosen [23], and email has been named a serial-killer application [8].

Also, usage of email is a highly personalized activity, and people use email in amazingly different ways. People read emails with different strategies: *archivers*

A. Cappelli and F. Turini (Eds.): AI*IA 2003, LNAI 2829, pp. 523–535, 2003.

try to read everything and not miss anything important, and *prioritizers* want to limit the time spent on email reading to switch to "real work" [9]. Accordingly to Whittaker and Sidner [24], 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). 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.

Finally, 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. Notifications could and should be delivered on the nowadays widely available smaller and portable devices (cell phones, PDAs, pagers, and so on), that are enabled to various network connection modes (GSM, GPRS, UMTS, Wi-Fi, Bluetooth, etc.), with the most appropriate modality (WAP-push, SMS, etc.). Notification should be done depending on features of the received messages like their number, their importance, the category they pertain to, and so on.

In this scenario, end users desperately need advanced tools for email processing, i.e., threading, categorizing, archiving, filtering, alerting, and so on. Today's email clients provide these functions in a rather limited way. All email tools notify the user sitting in front of his/her desktop that new mail has arrived by visual and/or acoustic messages. Mail tools allow manual categorization of 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 filter out spam, to automatically move the received messages in the appropriate folder, and so on. All 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 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).

The coming of portable devices is a new and important variable to add. Alerting will have an increased importance, and 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. 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. Complex engineering solutions are needed because the limited computational power available on the mobile devices require a server side based solution, in which most of the computation takes place on the server. And, finally, the increased email access by mobile devices will change the way people use email: nobody can predict all the range of new "perverted" or "creative" usages 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).

One of the research challenges today is to improve and make (at least partially) automatic the tasks of email categorization, filtering, and alerting. In this paper we deal with these research issues. The paper is structured as follows. In Section 2 we highlight the main issues related to email categorization and filtering and briefly survey the literature. 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 [19]. 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 [7]: Cohen [6] uses the RIPPER algorithm; Payne and Edwards [16] compare CN2 (a rule induction algorithm) with IBPL1 (a modified version of K-nearest Neighbor algorithm using memory based reasoning); Rennie [17] exploits naïve Bayes classifiers; Segal and Kephart [20] 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. 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 [10].

Spam (or junk) email filtering has seen an increasing interest in last years, due to the increasing amount of unsolicited emails: Pantel and Lin [15] and Sahami et al. [18] 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 good classification rule by combining 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., messages can be divided on the basis of the sender, of the topic, of one's own a-priori categorization of his 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.

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 (e.g., 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 erase some of the more confidential messages, thus making available only a portion of their message archive, that would be a biased sample; anyway, people willing to make public their email archives are not a good sample for sure, since more reserved people are left out; and relying on message 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 as follows. 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 [12]. 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.

ifMail categorization and filtering are based on the IFT (Information Filtering Tool) system [11,12], capable of profile building, storing, and matching. IFT has been developed on the basis of the UMT (User Modeling Tool) shell [3] and has been applied to a variety of systems and domains, e.g., Web filtering [2], filtering of enterprise documents [21], and filtering of scholarly publications [13]. IFT matches the profile associated to each category with the internal representation of each message and returns a result made up of three values: (i) *Coverage*, i.e., the percentage of the most relevant concepts of the profile which are also present in the documents, computed considering also the weights; (ii) *Match*, i.e., a measure of how much the concepts of the profile are present in the document (if they are more or less numerous in the document); and (iii) *Rank*, i.e., 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. 1 shows the overall architecture of ifMail. 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 three sub-modules:
  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, extracts lexical tokens, 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 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.

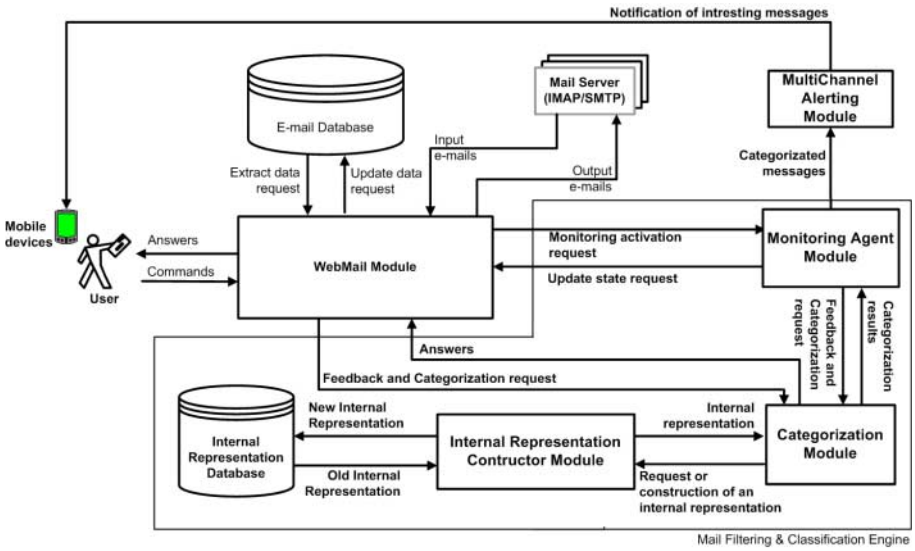

Fig. 1. ifMail overall architecture

– 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. 2 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. 2 show the multi-channel alerting of ifMail: in the screenshot on the left, 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 on the right).

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. We use the term "simulation"

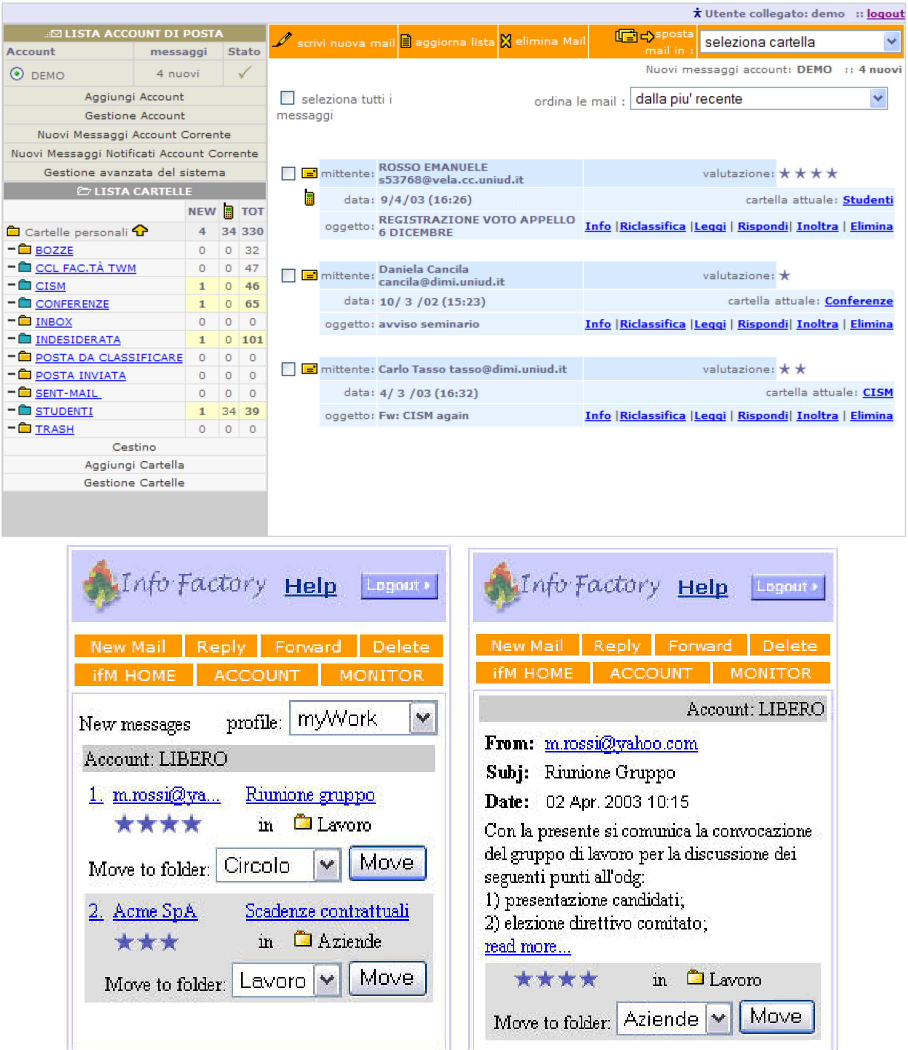

Fig. 2. ifMail user interface for Web mail (above) and PDA (below)

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. 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

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

data with a quick, light, and formative evaluation, capable of giving us hints on how to proceed with the development of the system.

Tab. 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. Both users 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.

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 its correct destination. 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. Tab. 2 illustrates the average (over all the available collections) of precision, recall, and F1 measure [22, 25], where the results obtained for each category are combined using the micro-average indicator [19].

First of all, we notice that the values obtained are in the range from 70% to 80%. Other experiments reported in the literature [14, 19] concern the categorization of the Reuters-22713 collection (21.450 articles divided into 135 categories) or the Reuters-21578 collection (12.902 articles divided into 90 categories): the values obtained for the F1 measure are in the same 70%–80% range. We have considered this result as a confirmation of the adequacy of the baseline performance of ifMail. Furthermore, the values reported in Tab. 2 are average values,

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%	74%

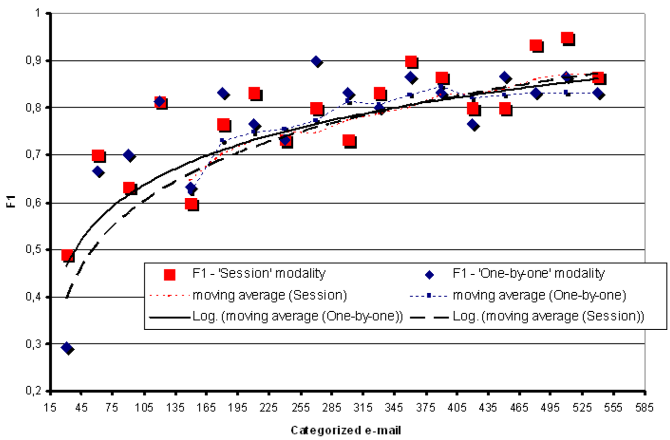

Fig. 3. Microaverage F1 in both operation modes for collection E

which include also the initial phases, where errors are most likely to happen: saturation ("steady state") values can be significantly higher.

Secondly, precision reaches higher levels than recall. We can interpret this phenomenon as follows: the number of messages considered is capable of reducing the number of categorization errors, but, on the other hand, 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. 3 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 indicating the termination of the initial learning phase) is reached earlier in the one-by-one mode. In the long run the two modes of operation reach the same level of performance.

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). Considering the Spam folder of collection B, precision reaches more than 95% and recall the range 70%-80%: this can be explained by

Table 3. Results for categories with well defined topics

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

the fact that when a spam message is received, all the subsequent messages 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).

Another (expected) phenomenon observed in the experimentation concerns the relationship between performance and specificity level 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. Tab. 3 illustrates such a situation for some categories with this characteristics.

Other experiments have been focused on identifying 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.

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 message towards its correct folder (category). More specifically, we have considered successive groups of 60 messages (i.e., four days), and we have counted (Fig. 4) the number of correct system categorization operations and 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. 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%.

5 Conclusions and Future Work

We have discussed 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-

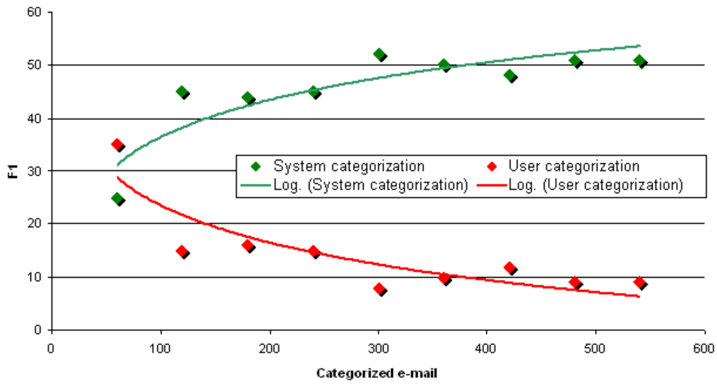

Fig. 4. Comparison of the number of user and system categorization actions

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.

We will continue this research in various ways. We are currently 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 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, how much time the user spent in reading it, or in answering it, for how long the user has not been checking his/her email, and so on) the software should also be capable of monitoring user's activity for a period of time.

References

[1] I. Androutsopoulos, J. Koutsias, K. V. Chandrinos, G. Paliouras, and C. D. Spyropoulos, An Evaluation of Naive Bayesian Anti-Spam Filtering. *Proc. 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 *AI*IA 97: Advances in Artificial Intelligence - Proc. of the 5th Congress of the AI*IA*, Springer Verlag, Berlin, LNAI 1321, 1997, pp. 242-253, 1997.

[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 *Proc. of the 17th International Conference on Machine Learning*, pp. 103-110, 2000.
[5] X. Carreras, L. Marquez, Boosting Trees for Anti-Spam Email Filtering, *Proc. of RANLP-01, 4th International Conference on Recent Advances in Natural Language Processing*, Tzigov hark, BG, 2001.
[6] W. Cohen, Learning Rules that Classify E-Mail, *Papers from the AAAI Spring Symposium on Machine Learning in Information Access*, pp. 18-25, 1996.
[7] E. Crawford, J. Kay, and E. McCreath, Automatic Induction of Rules for e-mail classification, *Proc. of the Sixth Australian Document Computing Symposium*, Coffs Harbour, Australia, 2001.
[8] N. Ducheneaut and V. Bellotti, Email as Habitat. *Interactions*, Sept./Oct. 2001.
[9] W. Mackay, Diversity in the Use of Electronic Mail: A Preliminary Inquiry. *ACM Transactions on Office Information Systems*, 6(4), 380-397, 1988
[10] G. Manco, E. Masciari, M. Ruffolo, and A. Tagarelli, Towards An Adaptive Mail Classifier, *Atti dell'Ottavo Convegno AI*IA 2002*, Siena, Italy, p. 63, 2002.
[11] 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, 1996.
[12] 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.
[13] S. Mizzaro and C. Tasso, Ephemeral and Persistent Personalization in Adaptive Information Access to Scholarly Publications on the Web. In *Adaptive Hypermedia and Adaptive Web-Based Systems, 2nd International Conference, LNCS 2347*, Malaga, pp. 306-316, 2002.
[14] I. Moulinier, G. Raskinis, J. G. Ganascia, Text Categorization: a Symbolic Approach, In *Proc. of the 5th Annual Symposium on Document Analysis and Information Retrieval*, Las Vegas, pp. 87-99, 1996.
[15] P. Pantel, D. Lin, Spamcop: A spam classification & organization program, in *Proc. of AAAI-98 Workshop on Learning for Text Categorization*, pp. 95-98, 1998.
[16] 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.
[17] J. D. M. Rennie, ifile: An application of Machine Learning to E-Mail Filtering, In *Proc. KDD00 Workshop on Text Mining*, Boston, 2000.
[18] 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.
[19] F. Sebastiani, Machine Learning in Automated Text Categorization, *ACM Computing Surveys*, 34(1), 1-47, 2002.
[20] R. B. Segal, J. O. Kephart, Incremental Learning in SwiftFile, *Proc. of the International Conference on Machine Learning*, San Francisco, pp. 863-870, 2000.
[21] C. Tasso and M. Armellini, Exploiting User Modeling Techniques in Integrated Information Services: The TECHFINDER System. *Proc. of the 6th AI*IA Congress*, Bologna, I, September 14-17, 1999, Pitagora Editrice, pp. 519-522, 2000.
[22] K. Van Rijsbergen, *Information Retrieval*, 2nd ed. Butterworths, London, UK, 1979. http://www.dcs.gla.ac.uk/Keith/pdf.

[23] G. Venolia, L. Dabbish, J. J. Cadiz, and A. Gupta, Supporting Email Workflow. Microsoft Research Tech Report MSR-TR-2001-88, 2001.
[24] S. Whittaker and C. Sidner, Email Overload: Exploring Personal Information Management of Email. *Proc. of the ACM CHI Conference*, pp. 276-283, 1996.
[25] Y. Yang, X. Liu, A re-examination of text categorization methods, *Proc. of the 22nd ACM SIGIR Conference*, Berkley, CA, August 15-19, pp. 42-49, 1999.