Adaptive Filtering of Electronic Mail

Joachim Soderberg, Bernard Merialdo
Institut EURECOM
BP 193
06904 Sophia-Antipolis
merialdo@eurecom.fr

Abstract

This paper describes a system for the adaptive filtering of electronic mail. The goal is to detect important messages which are flagged for the user to read in priority. The system uses the Latent Semantic Indexing (LSI) technique to extract "semantic concepts" from a set of sample messages. Adaptation takes place by adding new messages with their evaluation by the user, and eventually modifying the "concept space". Experiments show that the system is able to detect 10% more important messages than a random classifier. The fixed concept space version has been integrated into a working prototype.

This work is part of the MISTRAL project, in collaboration with Telis and Ingenia, with a sponsorship from the French Ministry of Industry.

I. Introduction

The development of communication technologies is causing a tremendous increase in the amount of informations that people receive regularly. Electronic mail is one major tool for transmitting these informations. With features such as aliases, mailing lists, carbon copy, it is very easy to distribute messages to many people, perhaps even too easy. As users receive more and more email messages, reading their mail becomes a difficult task, and the benefits of the time spent to this processing are not always clear. Certain users have arrived to a situation where they will decide not to use their email facility because it takes too much of their time when compared with the amount of interesting information they get.

Therefore, tools are required to assist users in processing efficiently their incoming messages, so that they can maintain their productivity at a sufficient level. Email filtering is an example of such a tool. A filter is a process that will be instructed about the user's preferences (the user profile) and that will evaluate incoming messages according to this profile. This evaluation will be an indication to the user about the content of the messages, and he will take this into account when deciding which messages to look at. Several kinds of evaluations can be performed on a message, for example it can be quantitative such as a relevance rating, or qualitative such as a classification among a set of predefined categories. There are also several ways to use this evaluation, for example by ranking messages that are presented to the user, by clustering them according to their categories, or eventually by presenting only messages satisfying certain criteria (and hiding the others).

In this paper, we propose, experiment and implement a method for the adaptive filtering of incoming messages. Messages are classified as either important or unimportant, based on the following definition: *"the user would like to be notified immediately when this message arrives in the system"*.

Messages are automatically classified using a technique called Latent Semantic Indexing that allows to compare documents. Adaptation takes place as the user provides his evaluation of new messages, that are kept as examples by the system. We evaluate the performance of this mechanism on a database of messages evaluated by three different users. Results show that the LSI filters perform about 10% better than a random filter.

In the rest of the paper, we briefly present the MISTRAL project where this work takes place, we provide references to some related works, we recall the basics of the Latent Semantic Indexing technique and we explain how we apply it in our situation, then we present two experiments of adaptive filtering. Finally, we give some information on the integration of the system into the MISTRAL prototype..

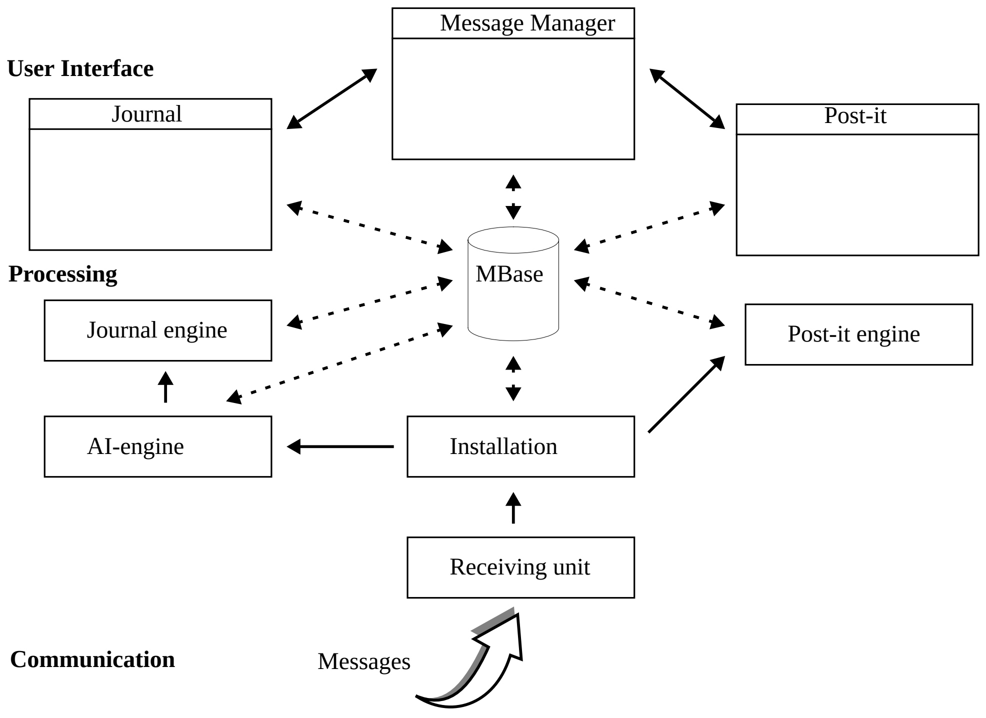


FIGURE 1. Architecture of the MISTRAL prototype

The MISTRAL project

The MISTRAL project (Messagerie Intelligente à Système de Traitement Reposant sur l'Analyse Linguistique) is a joint effort of three french parties: Télis, Ingénia and the Institut Eurécom, under the sponsorship of the French Ministry of Industry. The goal of the project is to augment an electronic mail software (developed by Télis) with advanced processing capabilities using Artificial Intelligence and statistical techniques. More precisely, two advanced functions have been implemented:

- linguistic analysis of the message body to perform a thematic classification,
- statistical analysis to evaluate the importance of the message.

The linguistic analysis of the message was realized by Ingenia. It uses a Natural Language parser and a semantic network to assign one or more categories to a message. These categories are used to organize the storage of messages and to facilitate message retrieval

The subject of this paper is the statistical analysis that was developed by the Institut Eurécom. Its purpose is to realize a "post-it" function that automatically detects important messages. On the user interface, messages that have been evaluated as important will be displayed with a special indicator, as if someone had put a post-it on the message to remind that it had to be processed in priority. The user who opens its mailbox will use this information to determine in which order he should read his messages.

II. Related Works

Information Filtering is a domain that is closely related to Information Retrieval, because similar methods can be used [Belk92]. Both deal with the detection of relevant information from a large amount of data. Filtering has the view that this data is flowing, while in the case of Retrieval, this data has a much static appearance. In Filtering, the criteria to define relevant information are fixed, or slowly changing over time, while in Retrieval, each new request can be completely different from the previous one.

Filtering has been applied to a variety of domains:

- electronic mail [Poll88] [Mack89] [Lutz90] [Palm93],
- newsgroups [Folt90] [Shet94] [Jaco93],
- selective diffusion of documents [Folt92],
- etc...

Even in the same domain, filtering can be used for different tasks. As an example, email filtering can be used by prioritizers [Mack89], users who do not read all their mail and use filtering to select the messages they look at, and archivers, users who read all incoming messages, and use filtering to categorize messages in one of several categories, so that they can store them in the proper location.

Many different approaches have been used to construct filtering systems, among which:

- rule based systems [Poll88] [Lutz90] [Mack89],
- natural language analysis [Ram92],
- statistical methods [Jaco93],
- neural networks [Scho92],
- collaborative approaches [Gold92]...

Filtering is also part in the development of intelligent agents, which assist users in performing certain tasks [Maes94].

Among the systems that have been developed, one can cite projects such as Information Lens [Malo87], MAFIA [Lutz90], INFOSCOPE [Fisc91], Tapestry [Gold92].

III. Latent Semantic Indexing

In Information Retrieval, a classical method for comparing documents is the vector space method, where a document is represented by a vector of number of occurrences of terms. Documents are close if the scalar product of the corresponding vectors is high. This method and its variants has proven to be valuable, however it is also known to have certain limitations, in particular it does not account for two phenomenons:

- polysemy: a given word may have several meanings, so that an occurrence of this word is not a firm indication of the meaning of the document,
- the fact that a given concept may be described in different terms by different users, so that documents which look different may have very related meanings.

The Latent Semantic Indexing (LSI) method has been proposed by Dumais et al [Furn88] [Duma88] [Deer90] to introduce an intermediate level (the latent semantic structure), so that documents are described according to “concepts” deducted from the terms used in the document. The relationship between terms and concepts is not one to one, in fact there is a reduction in dimensionality so that there are less concepts than terms. Several different terms can therefore contribute to the detection of a given concept, and also a given term may contribute to the detection of several concepts, so that this mechanism has the potential power of attacking the phenomenons previously described.

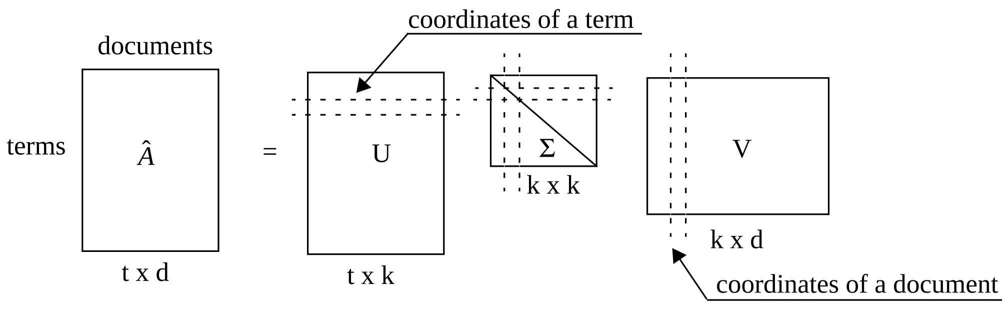
The latent semantic structure is found by using a mathematical method called “Singular Value Decomposition” (SVD) by which any rectangular *txd* matrix A can be decomposed as the product:

$$A = U\Sigma V^T$$

where U and V are orthogonal matrices of sizes *txn* and *dxn*, and Σ is a diagonal matrix of size *nxn* with non-negative coefficients (n is the rank of the matrix A):

$$U^T U = I_n \qquad V^T V = I_n \qquad \Sigma = diag(\sigma_1, \ldots, \sigma_n)$$

We can apply the SVD technique to the document-term matrix A, which contains the number of occurrences for each term in each document. This can be interpreted as introducing an intermediate space (the “concept space”) so that the relation between documents and terms is decomposed into a description of documents in the concept space (the V matrix), and a description of concepts into words (the U matrix). The coordinates of a document (resp. term) in the concept space are the rows of the V (resp. U) matrix.



The importance of each concept is evaluated by the quantity σ_i. The LSI method introduces a reduction in dimensionality by which only the k highest quantities are kept (and the others discarded). The rationale for this reduction is to keep only the most important information and to discard the noise.

For any document we can construct its representation in the term space as the vector X containing the number of occurrences of terms (this can be considered as a *tx1* matrix). We can then compute its representation in the concept space as the product (which is a matrix of size *1xk*):

$$X^T U \Sigma^{-1}$$

This representation allows to compute the “conceptual similarity” between two documents as the scalar product of their representation in the concept space.

It can be noticed that if there is no reduction in dimension (we choose $k=n$), the conceptual similarity is identical to the standard vector space measure. Therefore the gain in the LSI method is obtained by discarding unimportant information.

It can also be noticed how the occurrence of a term will affect a document in the concept space:

- a term may contribute to the occurrences of several concepts, as indicated by the coefficients in each row of the U matrix.
- several terms may contribute to the occurrences of the same concepts, as indicated by the coefficients in each column of the U matrix.

In practise, the computation of the SVD decomposition was realized using Michael Berry's SVD-PACK package [Berr93]. Details of the algorithm that is used can be found in the SVDPACKC User's Guide.

IV. Experimentations

Database of evaluated messages

To evaluate the performance of the LSI method on our filtering task, we needed a database of evaluated messages. Let us recall that we want to classify messages as either important or unimportant. By definition, a message is considered important for the user if the user desires to be notified immediately when this message is received by the mailing software.

We first gathered a set of 347 electronic mails that were sent to an internal mailing list during a definite period of time. This mailing list was devoted to general information among students, so that there is a great variety in the content of the messages: class schedules, social activities, announcements, advertising, etc... This database was annotated by three different users (we will call them A, B and C). Each user was presented the messages from the database one by one, in the order where they were sent to the mailing list, and he was asked to give his opinion on the importance of the message (in accordance with the previous definition of importance).

Each user annotated the messages according to his/her own preferences, so that the evaluations from various users are quite different. The following table indicates the percentage of important messages, as evaluated by each user.

User	A	B	C
important messages	19%	38%	19%

We also compute the coherence between the evaluation of different users as the percentage of messages that receive the same evaluation. The following table provides the numbers for all user-pairs. It can been seen that this correlation can be as low as 53%, which means that users B and C have quite different topics of interest.

Correlation	B	C
A	60%	74%
B		53%

Adaptive Filtering using LSI

The filtering mechanism will try to make an estimation of the importance of a message that is identical to the evaluation provided by the user. The LSI method provides a similarity measure between two documents. The filtering mechanism uses the LSI method to estimate a new message by computing the similarities of this message with examples of previously evaluated messages. If the new message is sufficiently close to important messages, it will be estimated as important, if not, it will be estimated as unimportant.

In this scheme, adaptation can take place in two different places:

- first, the concept space as defined by the LSI method is the result of the decomposition of the document-term matrix describing a set of messages. As new messages get through the system, it is possible to recalculate this concept space so that it provides a better description of the current terms. The effect is then on the similarity measure that is used to compare new messages.
- second, as the user browses new messages, he can provide an evaluation for these, so that the set of evaluated messages can be augmented.

It should be noted that the only information that is required from the user is an evaluation for sample messages. This is much less work than for adaptive systems which require the user to write rules or provide keywords.

When we compare a new message to examples of previously evaluated messages, there are several decision rules that can be used. For example, we can estimate a new message as important using:

- the same evaluation as the closest message,
- a majority vote over the N closest messages,
- the sum of the similarities of important messages minus the sum of similarities for

unimportant messages (among the N closest messages).

We tested several of these rules in our experiments, but did not find any clear advantage of a particular one (except for the fact rules tend to perform badly when N is too big, because irrelevant messages get introduced in the decision). We will report results using a majority vote over the 3 closest messages, which seems to provide good results in our case.

V. Filtering with a growing concept space

In our first experiment, we consider an adaptation with a growing concept space. The experiment is conducted as follows:

1. the filter starts with an empty set of evaluated messages,
2. a new message is presented to the filter,
3. using the LSI distance with the current concept space and the current set, the filter decides whether the new message is important or not (the decision for the first message is done at random),
4. the evaluation provided by the user is compared with the estimation of the filter, and the performance of the filter is updated,
5. the message and the corresponding user evaluation is added to the set of evaluated messages, which is used to construct a new concept space through a SVD decomposition of the new message-terms matrix,
6. the process goes on at step 2, until all messages in the database have been processed.

Thus, in this experiment, adaptation takes place at the two different places we already mentioned:

- the concept space is reconstructed each time a new message is added,
- each message is added in the set of evaluated message after its estimation.

In Figure 2, we plot the performance of the filtering mechanism as the database is processed. The performance is measure as the percentage of messages that have been correctly classified. The graphs are plotted for each of the three users A, B and C.

At the beginning of the experimentation, the performance of the filter looks erratic because the filter does not have much examples of the user interests and the percentages are computed on a small number of samples. After about 200 messages, the filter shows a more regular performance.

The performance of the filter is different depending on the user. The lowest performance is obtained for the user who has the highest rate of messages he considers important. In fact, we can observe that

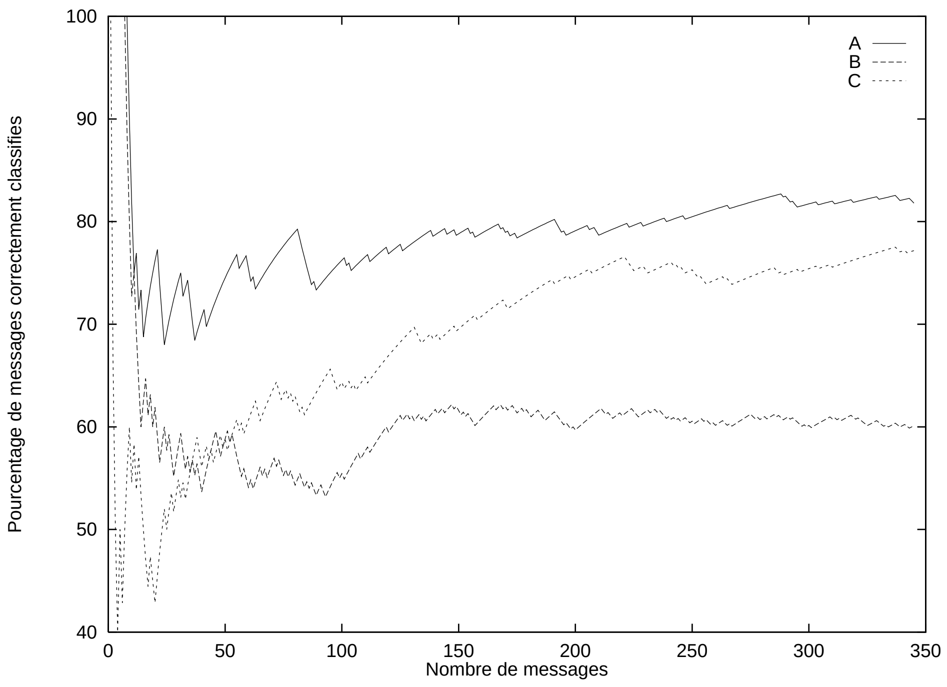


FIGURE 2. Performance of adaptive filtering

this rate has a great influence on the difficulty of the filtering task. If the user evaluates p% messages as important on the average (we assume that p<50), a random filter which would randomly consider q% of the messages as important would exhibit a performance that can be computed according to the formula:

$$1 - p - q + 2pq$$

In our case, we can compare the performance of the LSI filter with the performance of a random filter. Figure 3 presents the results for user A:

The results for all users are reported in the following table (we assume p = q for the random filter):

User	A	B	C
LSI Filter	82%	60%	77%
Random Filter	68%	52%	68%

This shows that the LSI filter provides an increase of about 10% in performance when compared with a random filter.

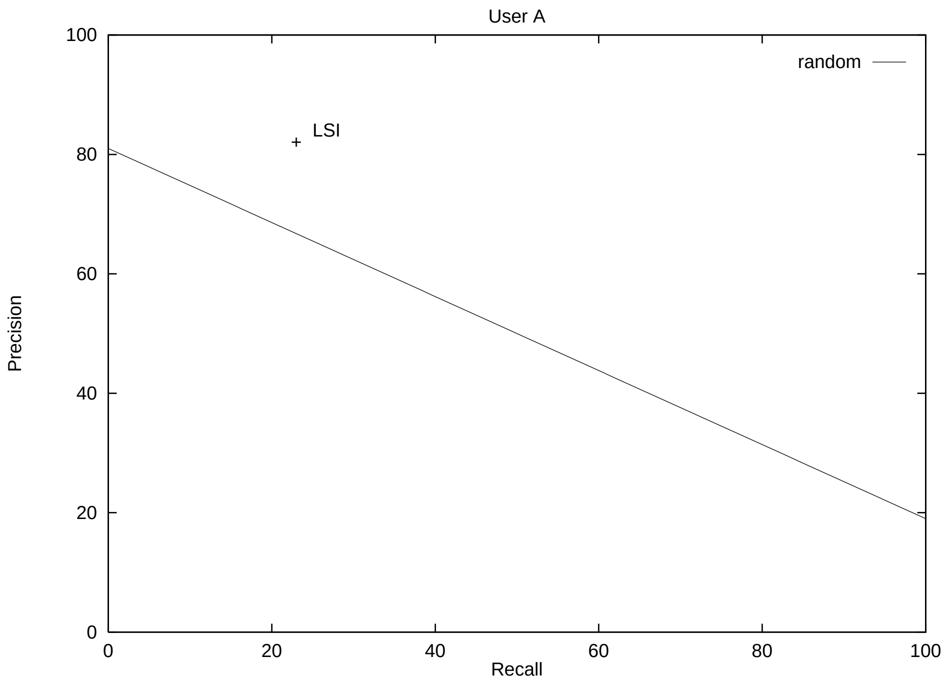


FIGURE 3. Precision/recall of LSI versus random

VI. Adaptive filtering with a fixed concept space

In the previous experiment, a new concept space is reconstructed each time a new evaluated message is known the system. This allows to make the best usage of the information available, since the concept space is always adequately chosen for the set of known messages. However, the computation that is required to perform the SVD decomposition can be expensive, so that it is difficult to implement this strategy in a real system. If a SVD decomposition were to take place each time the user provides a new evaluation, this would have a negative impact on the response time of the system. So we experimented with a reduced version of the adaptation mechanism that requires much less computation.

If we have a sufficient number of messages in the set of evaluated messages, it is expected that the concept space will not change much when we add a new message. Therefore it is reasonable to think that the performance of the filter will not be greatly affected if the reconstruction of the concept space does not take place at every new message, but only after a number of new messages have been seen.

When we augment the set of evaluated messages, this results in a linear increase of the time required to evaluate a new message, based on the number of comparisons that have to be made. This can even be reduced by an improved structuring of the data, and eventually by removing the oldest messages from this set.

We experimented with a second version of the LSI filter where the concept space does not change, but simply new messages are added in the set of evaluated messages after they have been seen by the user. To perform this experiment, we constructed a second database of 100 new messages (obtained from the same source as our first database), and asked our users for an evaluation of these messages. The table below presents the percentage of important messages as evaluated by each user in this new database.

User	A	B	C
Important messages	16%	43%	21%

We use our first database of 347 messages as the starting point of the second experiment. The filter starts with this set of 347 evaluated messages, and the concept space built from these messages. This concept space will remain the same through out the experiment.

Then we perform the following steps:

1. we take the next message from the new database of 100 messages,
2. we compare this new message to the current set of evaluated messages using the LSI similarity (which uses the fixed concept space), and the filter estimates if this new message is important or not,
3. by comparing with the evaluation from the user, we update the performance of the filter,
4. the message is added in the set of evaluated messages, but the concept space is not modified.
5. we iterate until all messages in the new database have been processed.

Figure 4 shows a plot of the performance of the filter for each of our three users.

We also compare these results with those that would be obtained by a random filter. Results are reported in the table below.

User	A	B	C
LSI Filter	79%	63%	78%
Random Filter	73%	51%	69%

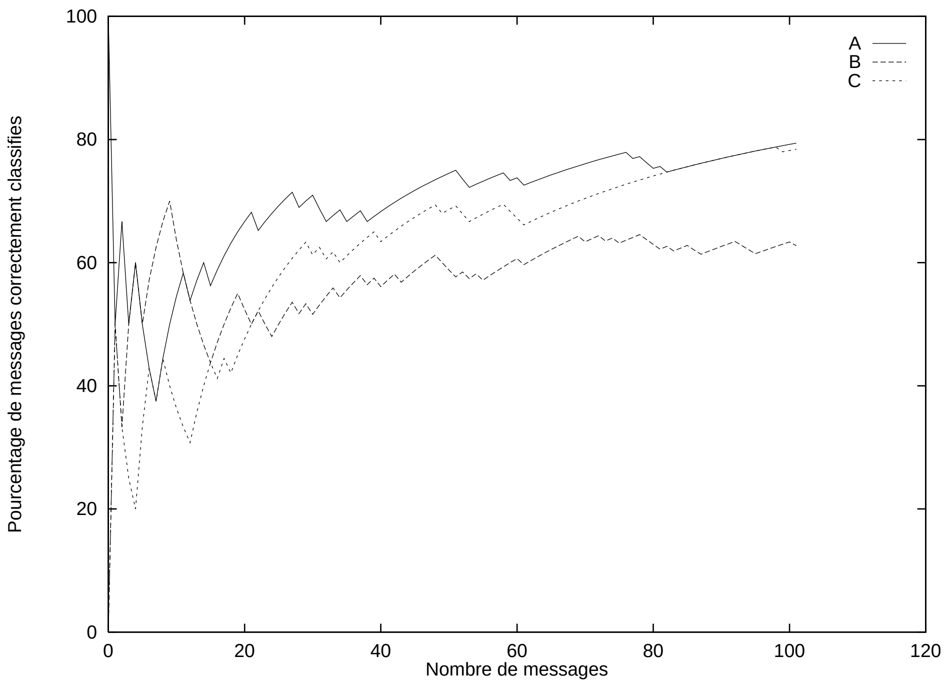


FIGURE 4. Performance of adaptive filtering using fixed concept space

This results are very comparable to those obtained in the first experiments. This indicates that the concept space that was constructed from the 347 messages provides reasonable performance when used on the new database. In our case, it indicates that the reconstruction of the concept space is not necessary at each new message, but that a fixed concept space can be used instead. Of course, on the long term, it is reasonable to expect that the concept space will become inadequate for the messages (for examples if new terms have been introduced), so that a reasonable alternative seems to reconstruct the concept space at regular intervals (either every N documents for large N, or at regular time intervals such as every week).

VII. Prototype Integration

One goal of the MISTRAL project was to integrate the advanced functionalities that had been independently studied into an existing mailing software to build a working prototype. This required to take into account implementation constraints that are generally not considered when the focus is on the evaluation of certain methods. In our case, this motivated the study of the filtering using fixed vector space, so as to avoid a reconstruction of the concept space at each new message.

Integration of the LSI filter into the prototype raised two kinds of questions:

- how to connect the filtering process to the prototype,

- what modifications of the user interface are required to accommodate the informations required or provided by the filter.

The integration was realized by defining a DLL to contain the routines that were involved in the LSI filter (the prototype was a PC Windows implementation). This includes routines to evaluate a new message, and routines to add an evaluated message to the set of evaluated messages. The standard mailing software was modified so that it could call the LSI filter for new messages. We also added routines so that the LSI filter can recover the header and body of the message from the message base maintained by the software.

On the interface side, we decided that the system should not be intrusive. Rather than prompting the user when new messages arrived and were estimated as important, we chose to flag important messages in the user mailbox with a special marker (the “post-it” marker). When the user opens his mailbox, or refreshes his window to include new incoming messages, important messages will appear with their post-it marker displayed next to the message subject, so that he can readily browse these in priority. He can also provide an evaluation to these messages (important or unimportant) to the system, in which case they will be added in the set of evaluated messages. Such a set is maintained for each user.

VIII. Conclusion

In this paper, we have presented an adaptive filtering system based on the LSI technique. This filtering system evaluates incoming messages as important or unimportant. Messages classified as important are flagged to the user. Adaptation takes place by augmenting the set of messages with known evaluation, and eventually by updating the concept space involved in the LSI computation. Experiments have shown that the LSI filter performs 10% better than a random filter. The reduced version of the filter without concept space update is shown to have similar performance.

This work is part of the MISTRAL joint project, and the filter has been implemented into a working prototype, together with other advanced functions developed in the project.

References

[Belk92] Nicholas J. Belkin and W. Bruce Croft. “Information filtering and information retrieval: Two sides of the same coin.” *Communications of the ACM*, 35(12):29–38, December 1992.

[Berr93] M. Berry. “SVDPACKC: Version 1.0 user’s guide.” Tech. Report CS-93-194, University of Tennessee, Knoxville, TN, 1993.

[Deer90] Scott Deerwester, Susan T. Dumais, and George W. Furnas. “Indexing by latent semantic analysis.” *Journal of the american society for information*, 41(6):391–407, 1990.

[Duma88] Susan T. Dumais, George W. Furnas, Thomas K. Landauer, Scott Deerwester, and Richard Harshman. “Using latent semantic analysis to improve access to textual information.” In *Proceedings of ACM CHI’88 Conference on Human Factors in Computing Systems*, Innovative Information Access, pages 281–285, 1988.

[Fisc91] Gerhard Fischer and Curt Stevens. “Information access in complex, poorly structured information spaces.” In *Proceedings of ACM CHI’91 Conference on Human Factors in Computing Systems*, Information Retrieval, pages 63–70, 1991.

[Folt90] P. W. Foltz. “Using latent semantic indexing for information filtering.” In *OIS90*, Filtering, Querying, and Navigating, page 40. 1990.

[Folt92] Peter W. Foltz and Susan T. Dumais. “Personalized information delivery: An analysis of information-filtering methods.” *Communications of the ACM*, 35(12):51–60, December 1992.

[Furn88] G. W. Furnas, S. Deerwester, S. T. Dumais, T. K. Landauer, R. A. Harshman, L. A. Streeter, and K. E. Lochbaum. “Information retrieval using a singular value decomposition model of latent semantic structure.” In *Proc. Eleventh Int’l. Conf. on Res. and Development in Information Retrieval*, Artificial Intelligence (1), page 465, 1988.

[Gold92] David Goldberg, David Nichols, Brian M Oki, and Douglas Terry. “Using collaborative filtering to weave an information tapestry.” *Communications of the ACM*, 35(12):61–70, December 1992. Special Issue on Information Filtering.

[Jaco93] Paul S. Jacobs. “Using statistical methods to improve knowledge-based news categorization.” *IEEE Expert*, 8(2):13–23, April 1993.

[Lutz90] E. Lutz, H. v. Kleist-Retzow, and K. Hoernig. “MAFIA - an active mail-filter-agent for an intelligent document processing support.” In S Gibbs and A. A. Verrighn-Stuart, editors, *Proceedings of IFIP WG8.4 Conference on Multi-User Interfaces and Applications*, Crete, 1990. North Holland.

[Mack89] Wendy E. Mackay, Thomas W. Malone, Kevin Crowston, Ramana Rao, David Rosenblitt, and Stuart K. Card. “How do experienced information lens users use rules?” In *Proceedings of ACM CHI’89 Conference on Human Factors in Computing Systems*, User Interface System Evaluations, pages 211–216, 1989.

[Maes94] Pattie Maes. “Agents that reduce work and information overload.” *Communications of the ACM*, 37(7):31–40, July 1994.

[Malo87] Thomas W. Malone, Kenneth R. Grant, Franklyn A. Turbak, Stephen A. Brobst, and Michael D. Cohen. “Intelligent information-sharing systems.” *Communications of the ACM*, 30(5):390–402, May 1987.

[Palm93] Jacob Palme, Jussi Karlgreen, and Daniel Pargman. “Issues when designing filters in messaging systems.” Tech. report, Dept of Computer Science, Stockholm University, Sweden, December 1993.

[Poll88] Stephen Pollock. “A rule-based message filtering system.” *ACM Transactions on Office*

Information Systems, 6(3):232–254, 1988.

[Ram92] Ashwin Ram. “Natural language understanding for information-filtering systems.” *Communications of the ACM*, 35(12):80–81, December 1992.

[Scho92] J. C. Scholtes. “Neural nets for free-text information filtering.” In *Proceedings of the 3rd Australian Conference on Neural Nets, Canberra, Australia, February 3-5*, 1992.

[Shet94] Beerud D. Sheth. “A learning approach to personalized information filtering.” Master’s thesis, MIT Media Lab, January 1994.