Automatic Classification and Filtering of Electronic Information: Knowledge-Based Filtering Approach

Omar Nouali[1&2], Philippe Blache[2]
[1] Basic Software Laboratory, CE.R.I.S.T, Algérie
[2] Laboratoire Parole et Langage, Université de Provence, France

Abstract: *In this paper we propose an artificial intelligent approach focusing information filtering problem. First, we give an overview of the information filtering process and a survey of different models of textual information filtering. Second, we present our E-mail filtering tool. It consists of an expert system in charge of driving the filtering process in cooperation with a knowledge-based model. Neural networks are used to model all system knowledge. The system is based on machine learning techniques to continuously learn and improve its knowledge all along its life cycle. This email filtering tool assists the user in managing, selecting, classify and discarding non-desirable messages in a professional or non-professional context. The modular structure makes it portable and easy to adapt to other filtering applications such as web browsing. The performance of the system is discussed.*

Keywords: *Information filtering, expert systesms, machine learning, neural networks, relevance feedback, genetic algorithms.*

Received March 29, 2003; accepted July 20, 2003

1. Introduction

The current exponential growth of the Internet precipitates a need for new automatic tools to help users cope with the amount of electronic information. Filtering is recognized as one way of helping a user in selecting of what to read and managing large information flows. It saves users the time they would spend in tiresome exploration and useless reading. Information filtering is neither a new concept, nor exclusively limited to electronic information. In addition, it does not focus on textual information only, i.e., in our daily live, we filter different types of information, be it textual, vocal, graphical, etc. For example, when we read a newspaper, the order in which we read the different articles depends on the level of interest we give them respectively. The representation of information in electronic form facilitates that such a process be carried out automatically by the system, which is given the responsibility to present information to users.

Several information filtering systems have been proposed in several domains, such as Mailing List, Usenet News (articles), Electronic Mail, World Wide Web, Electronic Conferences, Electronic bulletin boards, Clearing House Service, etc. [10, 22]. Such systems are limited because they do not use various strategies or approaches in information filtering process. They are based on the occurrence of a given set of keywords identifying possibly relevant information. They involve in human beings writing a set of logical rules which can filter and classify documents [5]. Given the amount of work required to design such rules by hand (time-consuming and often tedious process) and the success of machine learning techniques in text classification [2, 19], led us to use learning-based approaches (adaptive methods). These approaches consist of building automatic classifiers using machine learning methods trained on a collection of texts.

A growing number of machine learning techniques have been applied to text categorization in recent years, including multivariate regression models [11], nearest neighbour classification [23], Bayes probabilistic approaches [3, 15], decision trees [13], neural network [17], inductive learning algorithms [9], and support vector machines [12].

We propose an approach that combines expert systems, neural networks and machine learning techniques (relevance feedback and genetic programming). This paper is organized as follows. Section 2 gives an overview of the textual information filtering. Section 3 presents our e-mail filtering tool. Section 4 presents the experiments and the results. Section 5 summarizes the conclusions.

2. Textual Information Filtering Methods

A filtering system acts as an intermediate between the sources of information and users as shown in Figure 1. Filtering process aims to select and/or eliminate information from a dynamic data-stream. It is considered as the dual problem of the information retrieval: Information retrieval is concerned with the indexing of documents, while filtering is concerned with the indexing of profiles. Namely they are both concerned with getting information to people who need it. The used methods are similar [4]. In this section, we present the main models used in the filtering field.

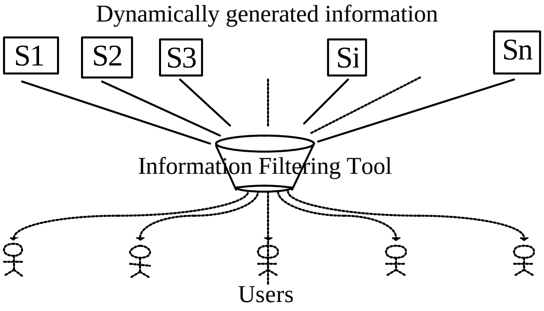

Figure 1. Information filtering process.

2.1. Boolean Information Filtering

Most commercial systems are based on the Boolean model [17]. It is the standard model based on an exact match of the profiles (user's interests) with the documents. A profile is described by a Boolean expression. The logical operators allowed are AND, OR and NOT. If a document satisfies the Boolean expression, that document is deemed to be relevant; otherwise it is deemed irrelevant.

The main advantage of this model is its simplicity, but it suffers particularly from strong limitations: It is difficult (even impossible) to determine the difference between the most significant terms and those which are not, because all the words have the same weight and the same level of importance. Interesting documents may not be retained if they do not contain all the words describing a user's profile. In addition, a classification of retrieved documents by order of relevance is not possible.

2.2. Vector Space Model

The vector space model is based on the statistical occurrence of terms in the profile (representing the user's information need) and the documents. Both documents and profiles are identified by terms and represented as vectors in a multidimensional space [21]. Each term is assigned a weight which represents its degree of importance. The degree of similarity between documents and profiles is measured by comparing their related vectors. The most commonly used similarity function is the cosine of the angle between the profile vector and the document vector [1]

$$COS\,\overline{(P,D)} = \frac{\text{å}^{\,T}(v_i * w_i)}{\sqrt{(\text{å}^{\,T} v_i * \text{å}^{\,T} w_i^2)}}$$

where v_i is the weight assigned to the ith term describing the profile P and w_i is the weight assigned to the ith term describing the document D. The advantages of this model are adaptability and robustness. It is more interesting, because it includes an evaluation of the relevance of the responses, but it suffers from the semantic problems (Synonymy, homonymy, word ordering, etc.).

2.3. Query Expansion

In general, the terms contained in a document are different from those the user would use to specify his interests. This raises a fundamental problem of term mismatch in information filtering. Query expansion techniques have long been suggested as an effective way to overcome this problem, such as term clustering, similarity thesauri, relevance feedback, latent semantic indexing (LSI), etc. [6]. The central idea is to extract expansion terms from a subset of documents and to use them in query expansion.

The LSI approach is one of the most used techniques [8]. It represents documents with concepts. It requires studying the entire text to extract the useful relationship between the terms and the documents. An automatic statistical method is used to calculate and simulate these relationships. First, a matrix *A* (terms-document) is built where A_{ij} is the weight of the ith term in the jth document. Second, *A* is decomposed into the product of three other matrices using a statistical technique, called Singular value decomposition (SVD): *A* = *U W V*, such that *U* and *V* are orthogonal and *W* is diagonal. *W* is reduced by ignoring some axes that correspond to the minimal singular values as shown in Figure 2.

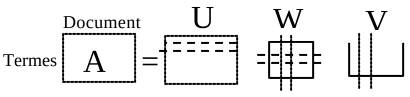

Figure 2. Singular value decomposition.

The result of the SVD is interpreted as establishing the relationships between terms and documents through a space of concepts. A document *X* is represented in the concepts space by: *X U W*, and the estimated similarity between two documents, called *conceptual similarity*, is processed by the cosine or scalar product of their representation in the space of the concepts. Contrary to the traditional methods, LSI often takes into account some undesirable phenomena. Indeed, it can filter and select documents, which don't match any words with the user's interests. However, it can be used to filter new information for more stable user's interest. But the update operation of the concept space is expensive in time: It requires (1) the availability of a corpus to build the matrix (terms-profiles) and (2) a long time to execute the SVD method. A solution to this problem is to run this operation regularly during hollow periods.

2.4. Neural Networks

The use of this model in information filtering field, consists in representing user’s interests as a network where the nodes represent concepts carrying weights and arrows represent relationships between concepts. This model is dynamic: it can learn and modify its behaviour progressively. After the training phase, the network can be used as a black box to process new data. There are multitudes of different types of neural networks (NNs) [7]. Some NNs are classified as feedforward while others are recurrent (i.e., implement feedback) depending on how data is processed through the network.

Another way of classifying NN types is by their method of learning (or training), as some NNs employ supervised training while others are referred to as unsupervised or self-organizing. Supervised training is analogous to a student guided by an instructor. Unsupervised algorithms essentially perform clustering of the data into similar groups based on the measured attributes or features serving as inputs to the algorithms. This is analogous to a student who derives the lesson totally on his or her own [7]. However, one of the main disadvantages of the NNs is their incapacity to explain the result which they provide.

2.5. Collaborative Filtering

Collaborative filtering is a form of social filtering [10]. It is based on the relationships between people and on their judgments. Users can use content filters to select, retrieve or filter documents based not only on the content of the documents themselves, but also on the evaluations and recommendations of other users. For example, users can indicate to the system the wish to accept all articles read by certain persons or authors. The system allows its users to annotate the documents they read. Collaborative filtering does not suffer from the problems which automatic techniques have with semantic aspect (synonymy, polysemy, homonymy, etc.).

2.6. User's Information Interests

The effectiveness of filtering is closely dependent on the quality of the profiles representation. The description of user's interests (profiles) is the most crucial and difficult operation in the building of an information filtering system. It is not easy for users to specify what those interests are because they differ from a user to another, and they are constantly changing [4, 20]. Generally, people describe their interests by providing either simply a set of keywords (*Keywords profile*) or a set of messages (*Documents profile*). Patterns of keywords are not enough to model interests. Semantic and contextual information must also be used. The profile of documents provides a simple and a very effective representation of user's interests.

3. Email Filtering

In this section, we present our approach to deal with information filtering problem, particularly in electronic mail domain. The design of an efficient filtering tool involves several tasks as shown in Figure 3, such as:

- Analysis and representation of messages.
- Representation of user’s interests (user’s model).
- Similarity measure (example: cosine).
- Filtering process (actions: delete, forward, save, etc.).
- Learning process (example: relevance feedback).

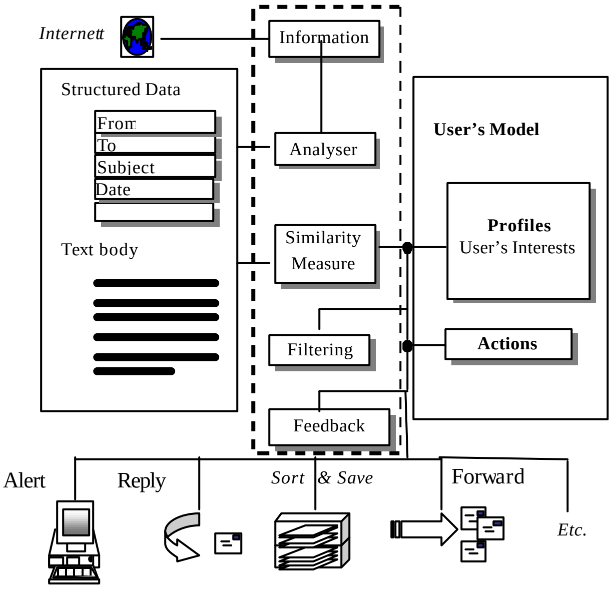

Figure 3. The basis model of filtering

Our system combines several techniques: an expert system to perform the filtering actions, a neural network based model of the user's interests, a relevance feedback and a set of genetic algorithms for the learning process as shown in Figure 4.

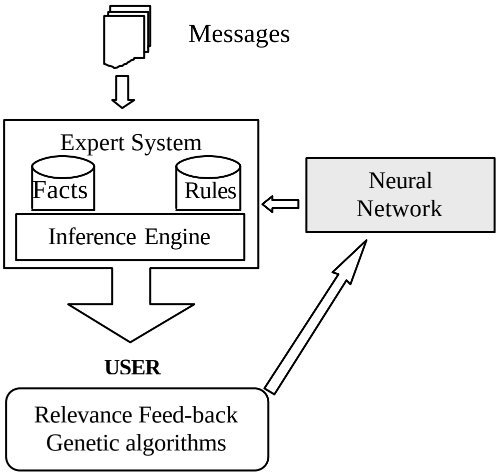

Figure 4. Architecture of the system.

3.1. Expert System

The expert system is composed of a set of rules *(IF <condition> THEN <action>)* and an inference engine. Upon arriving of a new message, a fact base is calculated and then presented to the inference engine. This base includes facts related to the different fields of the message and facts related to the user's status (*absent* or *present*). It might be extended during rules execution by facts such as *urgent message*, *personal message, professional message*, *interesting message, spam,* etc. The inference engine uses the forward chaining mode [17].

Two classes of rules are considered:

- *Class 1:* contains rules which conditions are applied only on structured attributes of the message (*From, Subject, To, etc.*). A single pass is sufficient to determine if a rule is selected or not. The filtering rule might, for example select only messages which contain the word *money* in the subject field.
- *Class 2:* the rules pertaining to this class cannot be selected in one pass. Their conditions are facts resulting from applying rules of the *class 1.* So they can only be selected in a second pass.

For example, one filtering rule sets a variable *interesting message* and another filtering rule might use this setting.

The system causes different actions to be performed on incoming messages, such as:

- To delete the message without reading it.
- To forward the message automatically to some other e-mail address.
- To cause the message to be automatically processed by a particular parser that might analyze the message body and estimate the similarity between the message and the user's profiles.
- To sort and save the messages into a separate folder for each user's domain.
- To present the messages in a certain order.
- Etc.

Furthermore, the system can explain and justify all message filtering decisions, by displaying all selected rules.

3.2. Neural Network-Based Model

The system knowledge model is represented by a neural network. The nodes represent the features and the arrows represent the relationships between features (the frequency of features occurrence which appears together in the message). Each feature is assigned a weight, which represents its degree of importance. We have created our initial knowledge model by simply selecting words with the highest value according to mutual information criteria [23]. The mutual information *MI (w, C)* between each word *w* and the class *C* is given by

$$MI(w,C) = \log_2 \frac{A*N}{(A+D)*(A+B)}$$

where *A* is the number of times *w* and *C* co-occur, *B* is the number of time *w* occurs without *C, D* is the number of times *C* occurs without *w*, and *N* is the total number of messages. Table 1 shows a summary of some important keywords for three considered profiles: *personal, professional* and *spam.*

Table 1. Message keywords.

Personal	Professional	Spam
à plus, a+, *amitiés,* *à bientôt,* *Bisous,* *bonjour,* *je, me, mes,* *mon, ma,* *moi,* *tu, te, tes,* *ton,* *ok, salut,* *etc.*	*Actes,* *cher,* *communication,* *cordialement,* *critères,* *langue,* *madame,* *monsieur,* *salutations,* *etc.*	*business,* *desires,* *free,* *fast,* *investment,* *miracle,* *money,* *quick,* *sex,* *etc.*

In addition to keywords, we have included specific features into initial model. Some important specific features taken into account are: *domain type of the message sender (.com, .edu, etc.), header length, type of message (html, plain text), message length, abbreviation, non-alphanumeric characters, numeric characters, language, attached documents, sentence length, punctuation (!, ?), date, etc.*

The adapted neural network model of our system is composed of three layers as shown in Figure 5 [16]:

- *Layer 1:* It is the *input layer*, represents the input message and is created dynamically. A message is conceptually represented by a vector $M= \{(f_1, q_1), (f_2, q2)\dots (f_k, q_k)\}$, where f_i is the ith feature describing the message and q_i is the weight assigned to f_i.
- *Layer 2:* It is the *hidden layer*. The nodes of this layer represent the system knowledge. The coming signal (layer 1) is propagated through this layer. Each node receiving a sufficient signal becomes active and sends it to its neighbors.
- *Layer 3:* It is the *output layer,* represents messages classes (or types). It provides the outputs of the network. Each node receiving a sufficient signal (layer 2) becomes active and constitutes the class corresponding to an incoming message.

The same sigmoid function [7] is used to activate nodes in different layers of the system model.

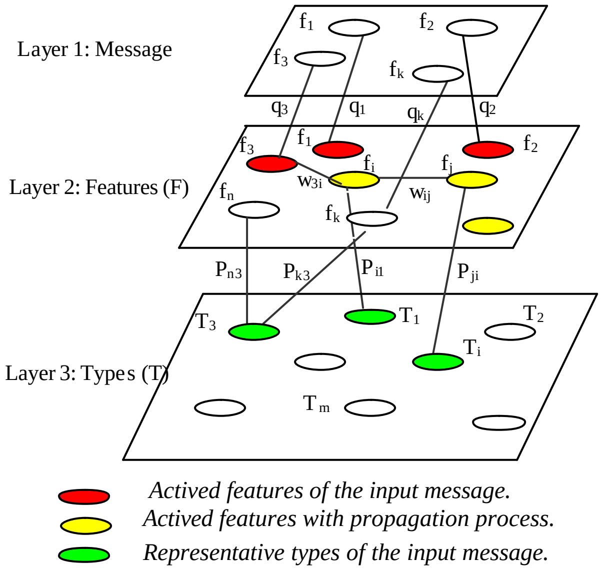
Layer 1: Message
f1
f2
f3
fk
q3
q1
qk
q2
Layer 2: Features (F)
f3
f1
f2
fi
fj
w3i
wij
fn
fk
Pn3
Pk3
Pi1
Pji
T3
T1
T2
Ti
Layer 3: Types (T)
Tm
🟥 Actived features of the input message.
🟨 Actived features with propagation process.
🟩 Representative types of the input message.

Figure 5. Adapted neural network model.

3.2.1. Training Process

The network is trained using backpropagation of error algorithm [7, 17], whereby the term weights of a training message are loaded into the input units, and if a misclassification occurs the error is backpropagated so as to change the parameters of the network and minimize the error. The learning process of the neural network is influenced by several parameters, such as connection weights, learning rate and momentum. The initial connection weights are chosen from an interval [-0.5, +0.5]. The default is 0.5. A higher learning rate can speed up training. However, if the learning rate is too large, training may fail completely. Often a value is chosen from an interval [0.05, 0.2]. The default is 0.1. The value of the momentum should be less than 1. The default is 0 (standard backpropagation without momentum).

3.2.2. Filtering Strategies

Once a message arrives, the vector of features is calculated and then presented to the network in a signal form (layer 1). This signal is propagated through the network. In fact, each node receiving a sufficient signal becomes active and sends it to its neighbours. At layer 3, each node receiving a sufficient signal from layer 2 becomes active and constitutes the response corresponding to incoming message. To compute the vector representation of a message, we first collect the individual words occurring in the message. Words that belong to the *stop list*, which is a list of high frequency words with low content discriminating power, are deleted. Then a stemming process is used to reduce each remaining word to word-stem form (term). For each term, a weight is assigned to represent its degree of importance.

The system performs two strategies of filtering: filtering without propagation and filtering with propagation. In filtering without propagation, each activated node of layer *F* will transmit its out-coming signal throughout links p_{ij} towards the layer 3. In filtering with propagation, the nodes of layer *F* with a sufficient signal become active and forward the signal to their neighbours throughout links w_{ij}. Actived message types whose outcome is the higher value will be considered as representative type of the input message.

3.3. Learning Process

In the last years, the machine learning approach has gained popularity and has become the dominant one in the research community, due to the amount of electronic information available. It has been used for a number of different applications such as automatic indexing, document organization, word sense disambiguation, text classification, hierarchical categorization of web pages, etc. [19]. Our system includes two kinds of learning: a*ssisted learning* named *relevance feedback* and *automatic learning* insured by a genetic algorithm.

3.3.1. Relevance Feedback

On presentation of the results from the filtering system, the user is invited to give indications on system decision quality. The user may provide relevance judgments for the filtered messages. These relevance judgments may subsequently be used to adjust weights of features that had contributed in the system decision.

3.3.2. Genetic Algorithms

Genetic algorithms intend to simulate the natural evolution process. They have been successfully applied to optimization and machine learning problems [17]. Coming from an initial population of individuals or chromosomes, representing tentative solutions of a problem, a new generation is created by combining or modifying the best individuals of the previous generation. The process ends when the best solution is achieved or after a number of fixed generations. Unlike relevance feedback, the system tries, upon explicit user request, to generate other profiles using existing ones. This allows the elimination of bad profiles and exploration of new domains which can interest user.

The application of genetic algorithms to generate profiles implies to establish the representation of chromosomes, the definition of crossover and mutation operators fitted to chromosome representation, and the definition of the fitness function used to determine the best chromosomes of a population. Each profile is a chromosome. Each word of the profile is a gene. Population size matches the number of different profiles. The fitness function of a chromosome is a performance measurement of the neural network defined by*: f(p)=Histoty_true(p)/History_total(p),* where *History_true* is a number of true decisions of the system (neural network) using profile *p* and *History_total* is a participation number of profile *p* in system decision. Crossover operator is the informatics transposition of natural reproduction phenomenon which allows inheritance of some characteristics from parents [17]. It selects the best profiles and generate from each pair of them two children. Each child inherits some of its characteristics from the first parent and the rest from the second one. Mutation operator consists of random change of one or several profile characteristics [17]. Crossover operator becomes inefficient with time, because children generated tend to be similar to their parents; at this moment mutation takes all its importance, it allows to propose to the user new domains which can interest him. The proportion of chromosomes involved in crossover and mutation operations is determined by crossover and mutation probabilities, which are set empirically. Crossover probability is initialized to 0.9 and mutation probability is initialized to 0.1.

4. Evaluation

4.1. Quantitative Evaluation

We illustrate and discuss the results obtained by two experimental evaluations. The system performances are measured using recall and precision rate. In the first evaluation, we seek to determine the efficiency of the features based model. The evaluation consists of measuring the system performances using two different strategies: filtering without propagation and filtering with propagation. The results are given in Table 2. In the first strategy, precision is greater than recall. The filtered messages which are not represented by profiles are ignored. In the second strategy, links between features contribute in filtering and increase recall.

Table 2. Filtering performances.

Filtering Strategy	Performances	
	Precision	*Recall*
Model without propagation	92,2%	83,1%
Model with propagation	93,4%	91,3%

In the second evaluation (Table 3), we analyze how the learning process influences the precision and recall rates. An automatic learning is performed after a series of assisted learning operations. During each session, the precision and recall are measured and the system is given the user's position to accomplish feedback operation in order to see its influence on these two rates. After feedback of many sessions weights are adjusted and bad features tend to disappear from the model, consequently the improvement of precision and recall is reached.

Table 3. Automatic learning efficiency.

Sessions	Performances	
	Precision	*Recall*
Session 1	92,2%	83,1%
Session 2	91,4%	88,3%
Session 3	93,3%	90,2%
Session 4	94,1%	91,4%
Session 5	93,7%	91,9%

4.2. Qualitative Evaluation

The features based model achieves high precision and recall. We notice some remarks:

- Certain features correlate with message content at a level that approaches but does not reach significance. Thus, any measure of a message text must take into account features that are always present and those that occur only occasionally.
- Automatic learning has a higher impact when the number of assisted learning operations varies from a session to another. Automatic learning efficiency depends linearly on assisted learning session's number.
- The messages which are not represented by keywords of profiles are ignored. Some semantic treatments are lacked.

5. Conclusion

Research in information filtering is still an open field. The current research in artificial intelligence, particularly in the field of natural language understanding, has resulted in technologies which in our opinion may help designing intelligent information filtering systems. However, a pure natural language approach may involve several capabilities in building and reasoning from explicit representation of user's goals. This may result in implementation and performance complexities that will not be acceptable and too costly [18, 22]. One of the most promising approaches is to combine natural language techniques (lexical semantic, terminology, shallow parsing, etc) and statistical methods. This combination may lead to a balance between the complexity of natural language approach and the less precise results of the traditional filtering approaches.

The E-mail filtering tool we presented here is an intelligent and dynamic, however we are convinced that by introducing some semantic treatments, we will significantly improve its performance. In our ongoing research, we are introducing the natural language techniques into the filtering process in order to take into account the semantic aspect of the application (information to filter). Some partial results have been obtained, but are still under validation.

References

[1] Abdelaziz Y. R., “Système de filtrage du courrier électronique: E-FILTER,” *Engineer Thesis*, INI, Algiers, Algeria, 2000.

[2] Amini M. R., “Apprentissage Automatique et recherche de l’information: Application à l’extraction d’information de Surface et au résumé de Texte,” *PhD Thesis*, Université de Paris 6, France, 2001.

[3] Androutsopoulos I., Koutsias J., Chandrinos K V., Paliouras G., and Spyropoulos C. D., “An Evaluation of Naïve Bayesian Anti-Spam Filtering,” *in Proceedings of 11th European Conference on Machine Learning in the New Information Age,* Barcelona, Spain, pp. 9-17, 2000.

[4] Belkin N. J. and Croft W. B., “Information Filtering and Information Retrieval: Two Sides of the Same Coin?,” *Communication of the ACM*, vol. 35, no. 12, pp. 29-38, 1992.

[5] Cohen W. W., “Learning Rules That Classify E-Mail,” *in Proceedings of AAAI Spring Symposium on Machine Learning in Information Access*, 1996.

[6] Cui H., Wen J. R, Nie J. Y., and Ma W. Y., “Probabilistic Query Expansion Using Query Logs,” *in Proceedings of 11th International World Wide Web Conference (WWW2002),* Honolulu, Hawaii, USA, 2002.

[7] Dreyfus G., Martinez J. M., Samuelides M., Gordon M. B., Badran F., Thiria S., and Hérault L., “Réseaux de Neurones, Méthodologie et applications,” *Edition Eyrolles*, 2002.

[8] Dumais S. T., “Using LSI for information retrieval, information filtering and other things,” *Bellcore Cognitive Technology Conference*, 1997.

[9] Dumais S. T., Plat J., Heckerman D., and Sahami M., “Inductive Learning Algorithms and Representation for Text Categorization,” *in Proceedings of Seventh International Conference on Information and Knowledge Management,* pp. 148-155, 1998.

[10] Goldberg D., Nichols D., Oki B. M., and Douglas T., “Using Collaborative Filtering to Weave an Information Tapestry,” *Communication of the ACM*, vol. 35, no. 12, pp. 61-70, 1992.

[11] Joachims T., “A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization,” *in Proceedings of 14th International Conference on Machine Learning*, pp. 143-151, 1997.

[12] Joachims T., “Text Categorization with Support Vector Machines: Learning With Many Relevant Features,” *in Proceedings of 16th European Conference on Machine Learning*, pp. 137-142, 1999.

[13] Lewis D. D. and Ringuette M., “Comparison of Two Learning Algorithms for Text Categorization,” *in Proceedings of Third Annual Symposium on Document Analysis and Information Retrieval (SDAIR),* 1994.

[14] Manning C. D. and Schütze H., *Foundations of Statistical Natural Language Processing*, MIT Press, Cambridge, Massachusetts, 1999.

[15] Mc Callum A. and Nigam K., “A Comparison of Event Models for Naïve Bayes Text Classification,” *Learning for text categorization*, 1998.

[16] Nouali O., “Classification Automatique De Messages: Une Approche Hybride,” *RECITAL*, Nancy, 2002.

[17] Oubbad L., Fouial O., and Nouali O., “Système Intelligent De Filtrage Du Courrier Électronique,” *Engineer Thesis*, INI, Algiers, Algeria, 2000.

[18] Ram A., “Natural Language Understanding for Information Filtering Systems,” *Communications of the ACM*, vol. 35, no. 12, pp. 80-81, 1992.

[19] Sebastiani F., “Machine Learning in Automated Text Categorisation,” *ACM Computing Surveys*, vol. 34, no. 1, pp. 1-47, 2002.

[20] Stadnyk I. and Kass R., “Modeling User’s Interests in Information Filters,” *Communications of the ACM*, vol. 35, no. 12, pp. 49-50, 1992.

[21] Yan T. W. and Garcia-Molina H., “Index Structures For Information Filtering Under The Vector Space Model,” *Department of Computer Science*, Stanford University, Stanford, CA, 1994.

[22] Yan T. W. and Garcia-Molina H., “SIFT: A Tool for Wide-Area Information Dissemination,” *in Proceedings of USENIX Technical Conference*, pp. 177-186, 1995.

[23] Yang Y. and Pedersen J. O., “A Comparative Study on Feature Selection in Text Categorization,” *International Conference on Machine Learning (ICML),* Nashville, TN, USA, 1997.

Omar Nouali had his Engineer degree in computer science in 1988 from Houari Boumediene University of Science and Technology (USTHB), and the Master degree (Magister) in computer science in 1991 from Advanced Technology Center, Algiers, Algeria. Currently, a "Responsible of research" in basic software laboratory. Research interests include artificial intelligence, expert systems, neural networks, natural language processing, information filtering, and human computer interface.

Philippe Blache is a "Research Director" at the CNRS (Laboratoire Parole et Langage, Université de Provence). His work concerns the implementation of linguistic theories and the development of NLP applications (especially concerning parsing, dialogue, alternative communication). He also has several international responsibilities in different associations and foundations in the field of computational linguistics (board member of the EACL, ESSLLI, ATALA, etc.).