Amati, G. and Crestani, F. (1999) *Probabilistic learning for selective dissemination of information.* Information Processing and Management, 35 (5). pp. 633-654. ISSN 0306-4573

PERGAMON

Information Processing and Management 35 (1999) 633–654

INFORMATION PROCESSING & MANAGEMENT

www.elsevier.com/locate/infoproman

Probabilistic learning for selective dissemination of information[1]

Gianni Amati[a],*, Fabio Crestani[b]

[a]Fondazione Ugo Bordoni, via B. Castiglione 59, I-00142 Roma, Italy
[b]Computing Science Department, University of Glasgow, Glasgow, Scotland, UK

Received 29 May 1998; accepted 21 December 1998

Abstract

New methods and new systems are needed to filter or to selectively distribute the increasing volume of electronic information being produced nowadays. An effective information filtering system is one that provides the exact information that fulfills user's interests with the minimum effort by the user to describe it. Such a system will have to be adaptive to the user changing interest. In this paper we describe and evaluate a learning model for information filtering which is an adaptation of the generalized probabilistic model of Information Retrieval. The model is based on the concept of 'uncertainty sampling', a technique that allows for relevance feedback both on relevant and nonrelevant documents. The proposed learning model is the core of a prototype information filtering system called *ProFile*. © 1999 Elsevier Science Ltd. All rights reserved.

1. Introduction

New information services deal with a variety of processes concerning the acquisition and the delivery of information. With the increasing availability of information in electronic form, it becomes more important and feasible to have automatic methods to filter and selectively disseminate information. Users may receive large amounts of information, like for example

* Corresponding author. Tel.: +39-06-5480-3424; fax: +39-06-5480-4405.
E-mail address: gba@fub.it (G. Amati)

[1] A shorter version of this paper was presented at the RIAO 97 conference in Quebec, Canada and is included in the proceedings (Amati, Crestani, Ubaldini, & De Nardis, 1997).

PII: S0306-4573(99)00012-6

electronic mail or news, and systems for *information filtering* (IF) are required to select only those documents which are relevant to some user information need.

Information or document filtering, also known as *selective dissemination of information*, is concerned with determining the information relevant to the user. The representation of the user's interest and information need (the *user profile*) may consist of a set of weighted keywords given by the user or induced by the system. When a user would like to have documents classified into different classes representing her/his different interests, then we prefer to talk of *class profiles*.

IF and information retrieval (IR) have been described as two faces of the same coin (Belkin & Croft, 1992). Much of the past research in IF has been based on the assumption that effective IR techniques were also effective IF techniques. Many of the IF approaches proposed at the TREC conferences, for example, were based on past successful IR approaches. This view has been challenged by Callan (1996) and by the proposer of the TREC-5 Filtering track (Harman, 1996). The idea is that different techniques and evaluation methods are required in order to design and evaluate effective IF and IR systems. In particular, IF requires more complex techniques of learning through relevance feedback than IR, since it is important to predict user's needs with a minimal amount of information provided by the user. An IF system that would require a long and painful training cannot be considered effective, despite its filtering performance. An IF system is effective when it performs reasonably well, while requiring a short training and a minimal interaction with the user.

In this paper we describe a learning model for IF which is an adaptation of the generalized probabilistic model of IR (Amati & van Rijsbergen, 1995). Two classes of learning models can be employed in IF: the relevance sampling and the uncertainty sampling. The first class contains the conventional learning techniques of IR, which basically process relevant documents using relevance feedback (Harman, 1992). The second class of models allows for relevance feedback also on the documents which were not considered relevant (Lewis & Gale, 1994). Our model belongs to the second class. In IR it has been observed that the uncertainty sampling is superior over the relevance sampling especially when the training set is very small (Lewis & Gale, 1994; Lewis, 1995). Our results indeed show that one needs very few documents in the training set to have good performance. The main contribution of this paper is in showing that our adaptation of the generalized probabilistic model of IR requires very little amount of training before achieving a stable level of effective performance compared to other training algorithms. Our study also aims at selecting the learning strategies which best combine positive and negative relevance feedback at different recall needs of the users and lengths of the user profiles. Our experiments show that starting from scratch and with small incremental training sessions a user can expect a reasonable effective performance of the system.

In the rest of the paper we describe and evaluate the learning algorithm of our IF system: ProFile. In Section 2 we give our view of the filtering task and compare it with the routing task introduced by TREC. In Section 3 we describe the current implementation of ProFile. In Sections 4 and 5 we describe in detail the probabilistic learning model at the heart of ProFile. In Section 6 we relate ProFile with other IF systems and research on the use of learning algorithms in IR. Finally, in Section 7, we report the results of an experimental investigation about the effectiveness of ProFile.

2. Filtering or routing?

In order to avoid confusion with terminology, we would like to clarify our use of the term 'filtering', as opposed to the term 'routing' that TREC introduced since its first event (Harman, 1993).

In the context of TREC, the routing task investigates the performance of systems that use standing queries to search a new stream of documents. A standing query is provided by an initial query (a natural language text describing the user information need) and a set of documents known to be relevant to that query (the training set). The documents in the new stream are ranked in relation to the standing query, in the same way as an IR system would do. In the routing subtask of filtering at last TREC 7, the required system output is a ranked set of 1000 documents and the evaluation measure is the average uninterpolated precision, namely the sum of all the precision values obtained by the position of the relevant documents in the ranked list, normalized with respect to the total number of relevant documents in the collection (Hull, 1998b).

In TREC, the filtering task is a routing task in which the system must decide whether or not to retrieve each individual document. Instead of producing a ranked list of documents, filtering systems retrieve an unordered set of documents for each query. The decision of retrieving a document or not is obtained through the application of a utility function in which correct decisions (retrieving a relevant document or not retrieving a nonrelevant document) and wrong decisions (retrieving a nonrelevant document or not retrieving a relevant document) have different benefits and costs. However, it is the introduction of the concept of time (Hull, 1998a) that really makes routing and filtering different in the context of TREC. The explicit use of time is to prevent the system from being trained with too many documents before filtering (as imposed by the last TREC 7 adaptive filtering subtask (Hull, 1998b)) or from filtering by ranking instead of exploiting a binary decision function. Indeed, the filtering task assumes that the user wants to be notified about each potentially relevant document immediately after it appears in the stream. A fixed threshold is then compared with the document weight to assess documents as relevant or not. In many filtering systems the choice of this threshold is learned. It is the matching value whose set of retrieved documents in a test collection maximizes the chosen utility function, e.g. F1, F2, F3, ranked and unranked average set precision (Hull, 1998a, b). This utility function can be also used to evaluate the system.

Although we believe that TREC filtering is closer to what goes on in operational IF systems than routing, our experience in running our IF system, ProFile, taught that users tend to behave in a different way from news clipping services and library profiling systems. The users of our system prefer to see a list of documents ordered by their estimated relevance, instead of an unordered set. Users with little time would look only at the documents at the top of the list, while users interested in a more extensive search would look quite further down in the list. The list does not need to comprise the entire set of the selected documents of the incoming stream.

The peculiar feature of ProFile is the construction of the threshold. It is not determined by a utility measure which is function of relevance and retrieval parameters, as it is obtained in the current practice of filtering systems. On the contrary, the initial threshold in ProFile is 0 and is determined by a utility measure which only depends on relevance feedback (see Section 5). The documents with positive and small weights are considered with low relevance in the model. The user may always set the threshold higher or lower than the current value, if he thinks the system is biased towards retrieving too many or too few documents respectively. The choice of the actual value of the threshold is thus left to the user and it is a function of the recall need of the user. In general, it is chosen to provide a list long enough to satisfy the most extensive search. In our operational implementation of ProFile, the list of retrieved documents is updated every day. New documents can appear at any point in the list depending on their relevance weight. Once a document has been seen by the user and assessed, it is removed from the list and stored according to user instructions.

Since it is the user in ProFile that actually chooses the threshold value and thus the level of recall to use, we have to evaluate the system by computing the precision at different recall points and not by using aggregate values as in TREC.

Our study was indeed aimed at finding the best learning strategies with *small and incremental training sets* at different recall needs of the users and lengths of the user profiles.

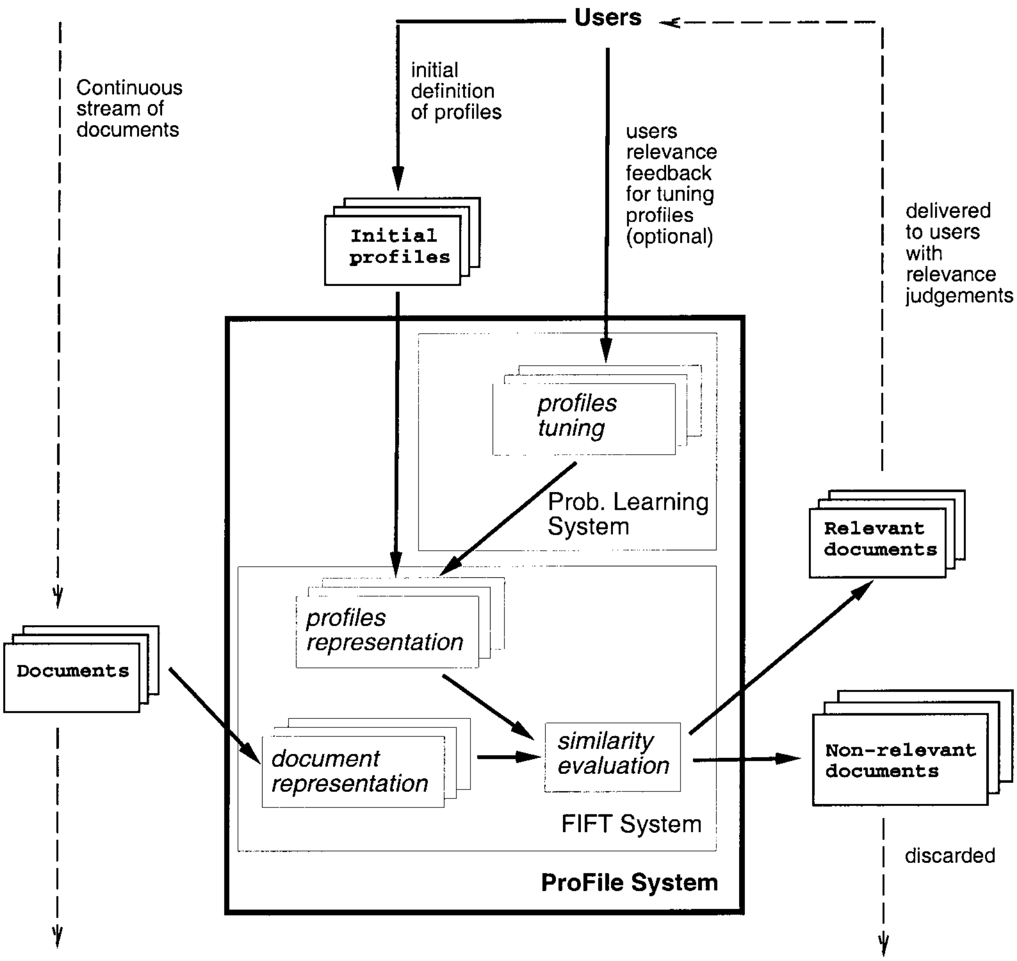

Fig. 1. ProFile architecture.

Given the way our IF system works, it looks like a combination of a routing and a filtering system, although more similar to a TREC routing system than a TREC filtering system. Our evaluation of its effectiveness therefore attempts to follow the guidelines of the TREC routing track and not of the filtering track.

Nevertheless, we decided to adopt the term ‘filtering’ for describing our system, because this is what our users believe the system actually does. This view is in partial agreement with the finding of Fidel and Crandall (1997).

3. The profile system

The *ProFile* (probabilistic filtering) system has been developed at Fondazione Ugo Bordoni in Rome (Italy) in 1996 and has been in used since then by many researchers of that institution for filtering the Usenet News (Amati, D’Aloisi, & Giannini, 1995; Amati, Crestani, Ubaldini, & De Nardis, 1997). Despite being born with the purpose of filtering netnews, ProFile can be adapted to filter any incoming stream of information, like email, newswires or newspaper articles.

3.1. Profile architecture

The ProFile system is made up of two components:

- *the FIFT system* (fub information filtering tool) (Amati et al., 1995), a customized version of SIFT, a filtering system developed at Stanford (see Section 6);
- the ProFile *probabilistic learning system*, that is the heart of the adaptation and tuning that ProFile performs on the users’ queries.

Fig. 1 gives an outline of the architecture of ProFile. In Section 3.2 we describe how ProFile works, while most part of the remaining of the paper will be devoted to explaining the probabilistic learning model employed by ProFile.

3.2. Profile at work

In ProFile each user may define a number of conceptual classes to classify the filtered documents: each class has its own profile. IF systems have two ways of assigning a document to a conceptual class. The first one consists of ranking documents according to a similarity values with the profiles of conceptual classes. A document is then assigned to the conceptual class with the highest level of similarity. This technique is appropriate when conceptual classes cover the set of all possible documents. Differently, another technique consists in defining a relation to be satisfied by each couple class-document. If the document satisfies the relation, then it is classified into that class, otherwise it is discarded. If a document satisfies relations with more than one class, then it is either classified into all classes or one is chosen (an arbitrary one or the one with the strongest relation, if that can be quantified). The model used by ProFile follows this second approach by exploiting semantic information theory (Bar-Hillel

& Carnap, 1953; Hintikka, 1970) and decision theory (Jeffrey, 1965). Details of the probabilistic model used by ProFile will be given in Section 4.

ProFile operates according to the following steps:

1. *Initial definition of the profiles*. The user defines a set of profiles corresponding to conceptual classes in which he wants to filter and classify the incoming stream of documents. ProFile requires from the user a set of keywords for an approximate initial description of each conceptual class.
2. *Training phase*. The initial description of the user interests is used as a query by the FIFT system. FIFT filters out of the document collection a set of documents that will be used as the 'training set'. The user go through the documents of the training set and assigns them relevance values with respect to each conceptual class. The relevance values are chosen from a scale of eleven values of interests (from 0 to 10). The user does not need to go through all the documents retrieved. The number of documents used in the training phase constitutes the *training data*. ProFile learning system performs an adaptation of the original description of the user interests according a probabilistic learning model to take into consideration the training data. The training phase can go on as long as the user requires, with as many retrieval runs, user relevance feedback and learning mode.
3. *Filtering phase*. The user decides to activate the filtering phase when he believes that the definition of the conceptual classes built by FIFT using relevance feedback are accurate enough. The filtering phase is made up of two subphases:
   3.1. *Filtering*. ProFile filters the documents and delivers to the appropriate user's conceptual class. The user can see the filtered documents classified into his personal conceptual classes.
   3.2. *Tuning*. The user can modify the profiles providing additional information. This can be achieved by giving relevance values to the filtered documents in the same way it is done in the training phase. The additional information enables ProFile to tune to the user perception of relevance and adapt the profiles of the conceptual classes. This phase can be repeated as many times as the user wants.

The initial training phase is very important for the effectiveness of ProFile. Indeed, in the limit case of no relevant document is in the training set (i.e. no document has been marked as relevant by the user before starting the filtering phase) the system will not retrieve any document and the user will not have any chance for correcting his profile with the tuning phase. On the other hand, in a preliminary experimental investigation we observed increasing recall, but decreasing precision for training sets which have more relevant documents than nonrelevant ones.

We should remind that in IR and IF, the performance of a system are measured either by precision at different values of recall or by utility measures which depend on the four sets of the relevance/retrieval contingency table (Hull, 1998b). *Recall* (R) is defined as the proportion of all documents in the collections that are relevant to a query and that are actually retrieved. *Precision* (P) is the proportion of the retrieved set of documents that is also relevant to the query. We will make extensive use of these measures in the remainder of the paper.

We observed that the best training set is obtained when the relevance values are equally distributed. Our way of training the system is similar to uncertainty sampling (Lewis & Gale, 1994; Lewis, 1995). Lewis and Gale (1994) observed better performance in IF using uncertainty sampling instead of relevance sampling (Ghosh, 1991), in particular when the sample size is small in comparison with the number of positive examples in the set of nonevaluated data. This is an important feature of ProFile, because the first set of evaluated document in the training set is very small. Typically, a user wants to activate the filtering phase after only 20 or 30 documents have been examined.

In Section 7 we describe the evaluation framework used and results achieved by ProFile. A special attention is given to the performance of our learning model, in particular when little training data is provided. We intend to evaluate the effect of using negative data in the relevance feedback, that is using the information provided by documents the user indicated as nonrelevant. In IR the use of negative data in relevance feedback has been received with contrasting views. Salton considered it positively (Salton & McGill, 1983a), while other researchers considered it dangerous (Aalbersberg, 1992) or even harmful (Dunlop, 1997). We believe that it depends on the particular retrieval model. We intend to prove that our model makes an effective use of negative data in relevance feedback and that the presence of negative data speeds up the learning of the parameters of a IF system.

4. A probabilistic learning model for IR

In this section we describe in detail the probabilistic learning model of ProFile. The model is derived from the generalized probabilistic model of IR presented in (Amati & van Rijsbergen, 1995).

4.1. Learning theory

At the abstract level IF can be seen as a process dealing with a repetitive event: a document is delivered to the user or not according to his current profile. A profile is a description of what the user is interested at. We assume that the document is represented by a set of terms (phrases, manually or automatically assigned index terms). The semantic relations between terms in the set $\mathscr{T}$ are implicitly explained by means of the set $\Omega(\tau)$ of documents which have been examined by the filter up to the current instant of time τ. In statistics this set can be considered as a *sample* of the *population*. Relations between terms are often expressed using frequency values. The user relevance assessments also provide a way of expressing semantic relations between terms.

A learning theory (Renyi, 1969) for IF is a triple $\langle \Omega, \mathscr{A}, \mathscr{P} \rangle$. Ω depends on a temporal parameter τ, $\Omega(\tau)$ being the set of all documents processed before the time τ. Here we assume that Ω is the set of documents which have constituted the data stream up to the current moment, so that τ can be omitted. $\mathscr{A}$ is the power set of Ω, namely the set of all subsets of Ω. $\mathscr{P}$ is defined by the user starting from the mutually exclusive elementary events, that is the elements d of Ω. This function is lifted from the elementary events to all the events e_i of the space $\mathscr{A}$ by using the additivity axiom.

In a finite space, a probability can be then obtained by conditioning. When

$$\mathscr{P}(e_2) > 0$$

the *conditioning* of $\mathscr{P}$ is defined as:

$$\mathscr{P}(e_1|e_2) = \frac{\mathscr{P}(e_1 \wedge e_2)}{\mathscr{P}(e_2)}.$$

Functions defined from Ω to the set of real numbers are called random variables. In our model a random variable is associated to each term $t \in \mathscr{T}$. With a little abuse of language we denote this random variable with t itself. Given a document $d \in \Omega$, the value $t(d)$ of the random variable t is the statistics on the term t in the document d. For example it can be either the tf weighing (the relative frequency of t in d) or the normalized idf weighing (defined as $\mathrm{idf}(t) = (-\log(n/N))/\log N$, where n is the number of documents in which t occurs and N is the number of documents in the collection (Salton & McGill, 1983a)) or the normalized combination of tf and idf(t) (defined as $c + (1-c)(-\log(n/N) \times \mathrm{tf}\ )/\log N \times \max) < 1$, where max is the maximum number of occurrences of the term in a document in the collection and c an arbitrary constant $0 \leq c < 1$ (Turtle, 1990)). In this paper we analyze the simplest case of the relative frequency of the terms, that is when little amount of information is provided to the system.

In other words if we denote by $\langle a_t^d \rangle_{d\in\Omega, t\in\mathscr{T}}$ the matrix $\langle t(d) \rangle_{d\in\Omega, t\in\mathscr{T}}$, then the column associated to d is the vector $\langle t(d) \rangle_{t\in\mathscr{T}}$ made out of the statistics of the set of terms in the document d, while the random variables $t \in \mathscr{T}$ are obtained by the rows of the matrix. In IR the matrix $\langle t(d) \rangle_{d\in\Omega, t\in\mathscr{T}}$ is called the *inverted file* of the collection Ω.

We can define the *conditioning expectation* of a discrete random variable t with respect to the measure $\mathscr{P}$ as:

$$E_{\mathscr{P}}(t) = \frac{\sum\limits_{d\in\Omega} t(d)\mathscr{P}(d)}{\mathscr{P}(\Omega)} \tag{1}$$

Note that if $0 \leq t(d) \leq 1$ then $0 \leq E_{\mathscr{P}}(t) \leq 1$.

In Amati and van Rijsbergen (1995), an IR model is introduced as follows. $\mathscr{P}$ corresponds to a subjective measure R of relevance on the event space Ω, its form is a scale of relevance weights $R(d)$, with $0 \leq R(d) \leq 1$, arbitrarily generated by the user. In ProFile, for example, we used a scale of 11 degree of relevance that are naturally mapped to the [0, 1] interval, but the whole continuous interval could be used. $\langle R(d) \rangle_{d\in\Omega}$ may be defined as a subjectively held vector and can be seen as a person's belief at the current instant of time. The dual measure of nonrelevance, $\neg R(d) = 1 - R(d)$, can be also defined. $\langle \neg R(d) \rangle_{d\in\Omega}$ can be seen as a person's disbelief on Ω.

As already pointed out, a random variable t takes the values $t(d)$ by means of statistics. Since $t(d)$ is related to frequencies we may suppose that $0 \leq t \leq 1$. $E_R(t)$ can be considered as a relevance/frequency weight of the term t, while $E_{\neg R}(t)$ as a nonrelevance/frequency weight of the term t.

When the system must decide whether a term is relevant or not on the basis of the expected measures of relevance and nonrelevance of documents, an error can occur and then a loss is produced. To make this decision the system computes the *expected monetary value* of decision theory (Amati & van Rijsbergen, 1995), that is:

$$\mathrm{EMV}(t) = \lambda_1 E_R(t) - \lambda_2 E_{\neg R}(t), \tag{2}$$

where λ_1 is the 'gain' when t is relevant to the user, while λ_2 is the 'loss' when t is not relevant to the user. The event 't is relevant' produces a benefit whenever $\mathrm{EMV}(t) > 0$. EMV can be equivalently given by the formula:

$$\mathrm{EMV1}(t) = \log\frac{\lambda_1 \times E_R(t)}{\lambda_2 \times E_{\neg R}(t)}. \tag{3}$$

4.2. Decision theory and semantic information

Since the fifties, the concept of information has been central in communication theory. Hintikka (1970) rightly argues that what is now known as *information theory* was first known as *theory of transmission of information*. He then suggested to call it *statistical information theory* in contrast to *semantic information theory* (Carnap, 1950; Bar-Hillel & Carnap, 1953). The basic connection between these two areas was the assumption of the *entropy* expression as a measure of information content either of a binary vector conveying information or of a logical sentence, respectively. The interpretations of this mathematical function however are deeply different: frequency is presupposed to be the basis in one case, while a purely logical characterization is sought in the second one. This difference has split the research into independent studies on the nature of information. The development of the semantic interpretation of information has been ignored, but we believe that it can be useful in the context of IR. Indeed, we show how to generalize the semantic information theory of Hintikka (1970) and how the probabilistic model can be easily derived in our framework as a particular case. We do not resort to the Bayesian inference as in van Rijsbergen (1979) but instead use utility theory.

Let us assume that the user has to decide whether to use the term t or not. t has the 'a priori' relevance value $E_R(t)$. Suppose also that t is relevant to the information need of the user. λ_1 would be then the 'award' if he takes t while λ_2 would be the 'cost' if he discards t (with a priori probability $E_{\neg R}(t)$). In the above formula what we actually gain or lose in taking t is unclear. However, if 't is relevant', then the user will gain the amount of information of nonrelevance of t: let us denote it by $\inf_{\neg R}(t)$. On the other hand, the loss λ_2 can be quantified by the amount of information of relevance of t, that is $\inf_{\mathrm{R}}(t)$. In both information theories (semantic and frequency-based) the amount of information is taken to be inversely proportional to probability, that is $\inf_{\mathscr{P}}(e) = -\log \mathscr{P}(e)$ or by the similar entropy expression. They share the principle that a sentence is more informative if it excludes more alternatives, that is, if it has a low probability (in particular tautologies are not informative at all because no alternatives can be excluded). Hintikka (1970), following Popper's notion of falsifier, suggests to use as a measure of information of a sentence the relative number of alternatives that the sentence excluded, more generally this can be formalized as

$\inf(e) = 1 - \mathscr{P}(e)$. In our case we have to assign the amount of information to random variables instead to sentences. By analogy, following the suggestion from Jeffrey (1965) and observing that the conditioning expectations do not go beyond the value 1, we may define the amount of information as:

$$\inf_{\mathscr{P}}(t) =_{\text{def}} 1 - E_{\mathscr{P}}(t).$$

Let us define $\neg t = 1 - t$, then:

$$\text{Inf}_{\neg R}(t) = 1 - E_{\neg R}(t) = \neg R(1) - \int_{\Omega} t d\neg R = E_{\neg R}(\neg t)$$

$$\text{Inf}_{R}(t) = 1 - E_{R}(t) = E_{R}(\neg t).$$

Substituting the values of the λs into Eq. (3), we have

$$\log \frac{E_{\neg R}(\neg t) \times E_R(t)}{E_R(\neg t) \times E_{\neg R}(t)} > 0.$$

The *absolute relevance of the term* must satisfy the constraint:

$$w(t_i) = \log \frac{E_R(t_i) \times E_{\neg R}(\neg t_i)}{E_R(\neg t_i) \times E_{\neg R}(t_i)} > 0. \tag{4}$$

4.3. The IR probabilistic model

Let us apply the model $\langle \Omega, P(\Omega), R \rangle$ with a particular relevance measure R. We assume that

1. R is the counting measure for the relevance of documents i.e. R takes a value $R(d) = 0$ or $R(d) = 1$ for every document according to the user relevance feedback;
2. a_i^d is the counting document-term matrix, that is:

$$a_i^d = \begin{cases} 1, & \text{if the } i\text{th term occurs in } d; \\ 0, & \text{otherwise.} \end{cases}$$

In the following n_R denotes the cardinality of the relevant set of documents, N the cardinality of Ω, r^i the cardinality of the set of relevant documents in which the term t_i occurs, $n\neg_R^i$ the cardinality of the set of nonrelevant documents in which the term t_i occurs, and finally n^i the cardinality of the set of documents in which the term t_i occurs.

By definition of a_i^d, the value $\Sigma_{d \in \Omega}\, a_i^d R(d)$ is the cardinality r^i of the set of relevant document in which the term t_i occurs. Substituting r^i into Eq. (1) we get $E_R(t_i) = r^i / n_R$.

Analogously, since:

$$\sum_{d \in \Omega} a_i^d \neg R(d) = \sum_{d \in \Omega} a_i^d (1 - R(d)) = \sum_{d \in \Omega} a_i^d - \sum_{d \in \Omega} a_i^d R(d) = n^i - r^i,$$

we have

$$E_{\neg R}(t_i) = \frac{n^i - r^i}{N - n_R}.$$

Finally

$$E_R(\neg t_i) = 1 - E_R(t_i) = \frac{n_R - r^i}{n_R}$$

and

$$E_{\neg R}(\neg t_i) = 1 - E_{\neg R}(t_i) = \frac{N - n_R - n^i + r^i}{N - n_R}.$$

The weight $w(t_i)$ defined as in Eq. (4) satisfies the following relation:

$$w(t_i) = \log\frac{E_R(t_i) \times E_{\neg R}(t_i)}{E_R(t_i) \times E_{\neg R}(t_i)} = \log\frac{(r^i/n_R - r^i)}{(n^i - r^i/N - n_R - n^i + r^i)} > 0. \quad (5)$$

This is the well known weighing formula of the probabilistic model of IR (Robertson & Sparck Jones, 1976; van Rijsbergen, 1979; Crestani, Lalmas, van Rijsbergen, & Campbell, 1998).

More generally, w_t can be used as a weight of relevance of the term t for the user and it must be greater than 0: greater is the value of w_t, higher is the degree of relevance of t. The vector $\langle w_t \rangle_{t \in \mathcal{T}}$ in ProFile can be thus considered as a weighted description of the user's profile. Note that if we used the vector $\langle E_R(t) \rangle_{t \in \mathcal{T}}$ as a description of user's profile we would not take into account neither the nonrelevant documents nor the documents where t does not occur. Hence the vector $\langle w_t \rangle_{t \in \mathcal{T}}$ is a more informative description of the user profile.

This result shows that relation of Eq. (4) generalizes the probabilistic model of IR.

5. Profile's learning model

The expected probability of relevance for IR can be easily adapted to define a filtering function. Let us assume that *n conceptual classes* $C_1, C_2, \ldots, C_n$ are associated to a single user. These conceptual classes can possibly be reduced to two: the user's class of relevant documents and the set of uncertain documents. Let us examine one document $x = \langle x_t \rangle_{t \in \mathcal{T}}$, on the set $\mathcal{T}$ of terms, at a time from a stream of documents. Then the probabilistic model $\langle \Omega, \mathcal{A}, R_C \rangle$, as described above, can be applied to each class.

Let $R_C(\Omega)$ be the sum of all assessment values $R_C(d)$ given to the processed documents up to the current instant of time. The vector of all weights $\langle w_t^C \rangle_{t \in \mathcal{T}}$, as defined by Eq. (3), will be matched with the new document x by a similarity function *SIM* (e.g. the vector space similarity function). In ProFile we use a variant of the vector space similarity function (Salton & McGill, 1983a). For the inner product, for example, we would get the equation:

$$\mathrm{SIM}(x,\langle E_{R_C}(t) >_{t\in\mathscr{T}}) = \frac{\sum_t x_t \sum_d a^t_d r^C_d}{R_C(\Omega)} \Bigg/ \frac{\sum_t \sum_d a^t_d r^C_d}{R_C(\Omega)} = \frac{\sum_t \sum_d x_t a^t_d r^C_d}{\sum_t \sum_d a^t_d r^C_d}, \tag{6}$$

where $R_C(d)$ is denoted by r^C_d. Note that in the above formula r^C_d can assume any real value since we are not restricting to considering a two-valued relevance probability R_C. This formula is not effectively usable since we need to store all the matrix (a^t_d) and the vector (r^C_d) to be able to compute the similarity function, that is $(|\mathscr{T}|+n)\times|\Omega|$ values where n is the number of conceptual classes.

Similar considerations apply when adopting other similarity functions instead of the Salton's similarity coefficient. This problem can be avoided by computing the conditioning expectation $E_{R_C}(t)$ of the relevance of each term t by means of Eq. (1) and incrementally updating this measure as soon as a new document is processed. In this way we need to store $(1+|T|)\times n$ global parameters, that is the values $R_C(\Omega_{\mathrm{old}})$ and $E^{\mathrm{old}}_{R_C}(t)$. Suppose now that a new document $y=\langle y(t)\rangle_{t\in\mathscr{T}}$ is incoming, so that $\Omega_{\mathrm{new}}=\Omega_{\mathrm{old}}\cup\{y\}$. Then the relation among the new values, $E^{\mathrm{new}}_{R_C}(t)$ and $R_C(\Omega_{\mathrm{new}})$ and the old values, $E^{\mathrm{old}}_{R_C}(t)$ and $R_C(\Omega_{\mathrm{old}})$, is ruled by the following transition equations, derived from the Eq. (1) and by the definition of Ω_{new}:

$$E^{\mathrm{new}}_{R_C}(t) = \frac{E^{\mathrm{old}}_{R_C}(t)R_C(\Omega_{\mathrm{old}}) + y_t r^C_y}{R_C(\Omega_{\mathrm{old}}) + r^C_y} \tag{7}$$

$$R_C(\Omega_{\mathrm{new}}) = R_C(\Omega_{\mathrm{old}}) + r^C_y. \tag{8}$$

Applying some algebra to Eq. (1) we easily get the nonrelevance parameters for t:

$$E_{\neg R_C}(t) = \frac{\sum_{d\in\Omega} a^t_d \neg R_C(d)}{\sum_{d\in\Omega} \neg R_C(d)} = \frac{\sum_{d\in\Omega} a^t_d(1-r^C_d)}{\sum_{d\in\Omega}(1-r^C_d)} = \frac{\sum_{d\in\Omega} a^t_d - \sum_{d\in\Omega} a^t_d r^C_d}{|\Omega| - R_C(\Omega)} = \frac{\sum_{d\in\Omega} a^t_d - E_{R_C}(t)R_C(\Omega)}{|\Omega| - R_C(\Omega)}.$$

By defining $a^t=\Sigma_{d\in\Omega}\, a^t_d$, we finally get:

$$E_{\neg R_C}(t) = \frac{a^t - E_{R_C}(t)R_C(\Omega)}{|\Omega| - R_C(\Omega)}. \tag{9}$$

This formula shows that we need to store other $1+|T|$ global parameters that is a^t and $|\Omega|$. When a new document $y=\langle y_t\rangle_{t\in\mathscr{T}}$ is incoming we can set up the equations for the transition from the old to the new parameters as follows:

$$|\Omega_{\mathrm{new}}| = |\Omega_{\mathrm{old}}| + 1 \tag{10}$$

$$a^t_{\mathrm{new}} = a^t_{\mathrm{old}} + y_t. \tag{11}$$

Once $E_{R_C}(t)$ and $E_{\neg R_C}(t)$ are computed and observing that

$$E_{RC}(\neg t) = 1 - E_{R_C}(t)$$

$$E_{\neg R_C}(\neg t) = 1 - E_{\neg R_C}(t),$$

we can now substitute them into the weights w_t of Eq. (4) and obtain the new value

$$w_C(t) = \log\frac{E_{R_C}(t)E_{\neg R_C}(\neg t)}{E_{R_C}(\neg t)E_{\neg R_C}(t)}. \tag{12}$$

To summarize, ProFile works in the following way:

1. For each incoming document and for each conceptual class C the user provides a relevance measure R_C, $0 \leq R_C \leq 1$.
2. $(|\text{Terms}| + 1)(n + 1)$ global parameters are needed to define a probabilistic model of filtering, where n is the number of the conceptual classes. These are the conditioning expectations $E_{R_C}(t)$, a^t, $|\Omega|$ and $R_C(\Omega)$.
3. By applying the decision theory we are able to provide a term t with a weighting formula $w_C(t)$ (see Eq. (12)). The weight $w_C(t)$ depends on the values $E_{R_C}(t)$, $E_{\neg R_C}(t)$, $E_{R_C}(\neg t)$ and $E_{\neg R_C}(\neg t)$. $E_{\neg R_C}(t)$ is obtained by the Eq. (9); $E_{R_C}(\neg t)$ and $E_{\neg R_C}(\neg t)$ are equal to $1-E_{R_C}(t)$ and $1-E_{\neg R_C}(t)$ respectively.
4. When a new document $y = \langle y_t \rangle_{t \in \mathcal{T}}$ is examined, the global parameters are updated according to Eqs. (7), (8), (10) and (11).
5. Finally, any similarity function SIM can be applied to the vectors $x = \langle x_t \rangle_{t \in \mathcal{T}}$ and $w_C = \langle w_C(t) \rangle_{t \in \mathcal{T}}$ to compute a real number value for the membership of x to C. The conceptual classes containing the document x are such that: $\text{SIM}(x, w_{C_j}) > s_C$, where s_C is a threshold value. From a theoretical point of view s_C must be equal to 0. However, this threshold is experimentally greater than 0. Note also that if the user always gives the maximum uncertain value $\frac{1}{2}$ to each document in the stream of documents then w^C is the null vector.

6. Related work

Most current models of IF have their origins in the studies of the use of relevance feedback in IR. The learning process required by filtering is, in fact, very similar to the learning process used by relevance feedback. In both cases an initial description of the user information need (the query or topic) is augmented/modified through the provision of additional relevance information. The additional relevance information is often provided in the form of documents that are relevant to the same user information need expressed in the query. It is the task of the learning process to extract statistical relevance information from these documents to adapt a user relevance profile. However, despite these apparent similarities, IF and IR differ greatly in other respects, as was pointed out in Belkin and Croft (1992).

The probabilistic model of IR combine frequency values with relevance assessments using the Bayes' theorem. In van Rijsbergen (1979) relevance as well as the set of terms are taken as elementary events. On the contrary, in Maron (1961) the absolute probability of a document is given by the number of its uses divided by the number of total uses, while relevance is a subjective weight attached to each couple term-document and interpreted as the conditioning probability of a term given a document.

In relevance feedback models of IR it has been argued that the estimation of the prototype vector of a class of relevance should be made also from the remainder of the collection. In NewsWeeder (Lang, 1995) this is partially recovered by computing linear regression from the rating categories. The probabilistic model of IR solves this problem just for two classes of relevance (Crestani et al., 1998). This method is known as the complement method (Harper & van Rijsbergen, 1978). NewsWeeder uses a finite number of user's rating categories (the first for the class of most relevant documents up to the last for the class of completely irrelevant documents) partitioning the training set, it then uses the tf × idf (term-frequency multiplied by inverse-document-frequency, see Salton & McGill, 1983a) to assign a new document to exactly one category. This approach is a breakthrough from the classical two-valued interpretation of relevance proposed in IR. On the other hand, this approach considers these categories unrelated and only in the predictive phase a comparison is made by using a similarity function between the prototype vector of a category (centroid according to Salton's terminology) and the new document.

In SIFT (Yan & Garcia-Molina, 1995) the user describes the topics of his interest. However, this initial representation is not effective or complete and relevance feedback is needed to correct the definition of the profile. Typically, the system must learn a profile containing thousands of weighted terms, starting from a vector of a few initial terms, in order to be effective.

These proposals do not offer a general way to directly combine relevance with the frequentist analysis of a data stream. In Amati and van Rijsbergen (1995) a learning model proposes a natural interpretation of relevance as well as a way to amalgamate it with rank-frequencies theory. This is the probabilistic model used by ProFile and described in Sections 5 and 4.

Okapi (Robertson, Walker, Hancock-Bealieu, Gull, & Lau, 1993) and InRoute (Callan, 1996) are other two examples of filtering systems based on probabilistic models. The classical Robertson–Sparck Jones weighting function, which is at the basis of the Okapi system, was shown to be poorly effective (Robertson et al., 1993). The reason is that the Robertson–Sparck Jones weighting function does not take into account the statistics about the observed document, like the within-document term frequency and the document length (Robertson, Walker, Hancock-Bealieu, Jones, & Gatford, 1995). Many new weighting functions have been then tried as variations of the Robertson-Sparck Jones weighting function w_{RSJ}. The w_{RSJ} has been corrected by a multiplicative factor which takes into account any extra relevant document parameter. ProFile has a different weighting philosophy: it is the term weighting w itself that includes the document statistics. In particular w is the weighting w_{RSJ} when the within-document term frequency reduces to the simple binary value of occurrence nonoccurrence of the term in the document. In addition, ProFile allows for nonbinary relevance judgment values which cannot be considered in the Robertson-Sparck Jones weighting function.

InRoute assumes the same weighting philosophy of Okapi: the initial term weighting given by the idf function is corrected by a linear combination $p_1 + p_2 \times \mathrm{ntf} \times \mathrm{idf}$, where the beliefs p_1,

p_2 satisfy the condition $p_1+p_2=1$ and ntf is a normalized within-document term frequency formula. InRoute stores and updates the idf weights for only those terms that appear in at least one profile. This term selection allows for a fast and effective use of the inverted file, provided that the set of terms of the profiles do not change in time and the number and the size of profiles are small.

In SMART (Salton & McGill, 1983b) the relevance feedback interaction is similar to that used in IR, where the system takes into account also the number of relevant and irrelevant documents among the selected ones. Similarly to what happens in IR, the user is asked to make a sharp decision on relevance. This is not an easy task because of the presence of documents with uncertain relevance (i.e. different from completely nonrelevant or completely relevant). In ProFile the relevance feedback consists of choosing arbitrary degrees of relevance values, which are interpreted in the model as a subjective probability distribution on the incremental set of filtered documents. The user is thus able to express his rate of uncertainty. In general, graded relevance feedback and on-line adaptability seem necessary for the development of effective and personalized filtering systems in which long-term requests are subscribed and a selection of only few documents for training is required. This makes a nontrivial difference from IR, which is usually concerned with retrieving documents from a relatively static database by means of only few sessions of interaction and retrieval.

In NewsWeeder, relevance feedback consists in rating values of interest. In contrast to ProFile which has a single profile for each topic of user's interest, NewsWeeder considers the associated class of documents with the same degree of interest (a rating category) as a profile, and the filter classifies documents into these categories. The learning phase of NewsWeeder is off-line: indeed the system learns a new model of user's interests each night by taking into account the overall history of user's relevance assignment on the training documents which must be saved and kept for each user as a profile. In Lang (1995) filtering results are reported, comparing precision against the number of training examples. These results were built only with two users. For the user A the system has a precision of 59%, and for the user B the system has a precision of 44% with respect to very large training sets (some thousands of documents). We consider this evaluation very poor.

A further comparison of ProFile with other many IF systems proposed in the literature would take far too much space. In fact, in recent years a large number of IF systems have been proposed. One application area that has been heavily targeted is news filtering (Kilander, 1995). Moreover, much effort has been devoted to IF in the context of the TREC initiative, as the increasing number of participants to the two sessions of 'routing' and 'filtering' proves (see TREC-5; Harman (1996) for example). The area of IF brings together many different experiences from other areas, like machine learning, data mining, knowledge representation, human–computer interaction. The main contribution of IR, and in particular of TREC, to the IF community is in providing sound evaluation techniques. We believe that a sound set of evaluating techniques was really needed in IF, where researchers have been evaluating their work in many different and sometimes arguable ways. We intend to take advantage of the TREC contribution by adapting the TREC evaluation guidelines to the evaluation of ProFile, as reported in Section 7.

7. Evaluation framework and results

In this section we report on the evaluation of the performance of the proposed IF learning model, in particular when little training data is provided.

The collection we used is the *TREC-5 B* (Harman, 1996) a subset of the collection used in the experiments done in 1996 in the context of the TREC 5 initiative. The collection is made of selected full text articles of the Wall Street Journal (years 1990–1992). Some of the characteristics of this test collection are reported in Table 1. We used a set of 50 queries (or topics, as they are called in TREC) with the corresponding set of relevant documents that were used for the training and for the evaluation.

The evaluation was performed according to a routing system (see Section 2). The retrieval effectiveness measures we used are recall and precision, already defined in Section 3. The motivation of using the standard precision function at different recall values was explained in Section 2. We remind that these measures are related in such a way that high precision brings low recall, and vice versa. In other words, if one desires high precision, he has to accept low recall, and vice versa. In order to give a measure of the learning performance of the filtering algorithm, recall and precision have been evaluated with different dimensions and compositions of the set of training examples. The results reported in the following tables are averaged over the entire set of 50 topics.

At each run we trained the system with only very few documents. In ProFile this training phase corresponds to the initial phase in which users assess a small number of documents retrieved by FIFT. We remind that ProFile does not exploit the idf function, hence the whole information which we use in the experiment is contained in this small set of documents. The training data of each run was a subset of up to 64 documents randomly chosen among the known relevant and nonrelevant documents (i.e. those marked as relevant and nonrelevant by TREC assessors). The filtering runs shown in Tables 3 and 4 are thus incremental. The figures reported in those tables should be read as percentage variation of the precision of a base run. In Table 2 we report the base run, performed using only the information provided by the text of the topics, without any additional information.

Before commenting the results reported in the tables, it is important to notice that for all the

Table 1
The Wall Street Journal 1990–1992 document collection

Data sets:	WSJ 1990–92
Number of documents	74.520
Size in MB	247
Number of queries	50
Unique terms in documents	123.852
Unique terms in queries	3.504
Avarage document length	550
Avarage document length (unique terms)	180
Avarage query length	40
Avarage number of rel. doc. per query	35

Table 2
Performance of ProFile for the base run

Rec.	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Prec.	0.68	0.49	0.40	0.33	0.28	0.22	0.17	0.11	0.08	0.04	0.01

runs reported in this evaluation we did not exploit any statistical information concerning the entire collection, like for example the idf weighting function used by many IR systems. The knowledge of such information would have required the processing of the whole collection in advance, something that can be done for IR applications, but not for IF applications. This explains why our base run produced rather low performance compared with those an IR system could have produced. Moreover, we only used a simple stop list (i.e. a list of term not to be used in the indexing of documents and queries (van Rijsbergen, 1979) and we did not employ any stemming function (i.e. a function that reduces words to stems (Frakes, 1992)), since we wanted the system to be language-independent. Although with these settings we considerably reduced the retrieval effectiveness compared with IR techniques, we wanted our experimentation to be as general as possible and therefore language-independent (no use of stemming) and collection-independent (no use of date stamp on document or information concerning their domain). The hypothesis is that the system cannot absolutely know and process in advance the incoming data and, similarly, making some ‘educated guesses’ on the content and term distributions in the stream cannot be possible. A different approach was followed by Allan (1996). Allan assumes that the statistical information about the full collection can be determined by the statistical information extracted from a sample of relevant and nonrelevant documents, namely from the set of documents which have been processed up to a given time. Of course in statistics this technique works when the sample data are not biased. In IR this amounts to say that sampling works the better the larger the sample and the more homogeneous in content the documents are. It would not work for a stream of

Table 3
Precision increment in percentage w.r.t. the base run by using only relevant documents (R) as training. AT means all terms of the training data and of the topic in the profile, TT only terms in the topic and HF only high frequency terms and terms in the topic

Recall	4R-TT	8R-TT	16R-TT	32R-TT	32R-HF	32R-AT
0.0	+0.25	+4.00	+5.95	+8.18	+13.15	−8.50
0.1	+0.20	+5.96	+6.71	+10.78	+16.28	−8.70
0.2	+1.72	+2.76	+9.51	+8.84	+3.63	−7.39
0.3	+0.45	+3.78	+12.82	+8.69	+3.17	−13.34
0.4	+0.77	+1.98	+7.33	+4.52	−6.94	−23.99
0.5	+1.28	+2.82	+9.60	+4.96	−16.45	−30.7
0.6	+0.68	+2.09	+6.05	+6.54	−22.25	−37.76
0.7	−0.28	−1.22	+1.61	+6.05	−18.71	−36.70
0.8	+1.16	−3.14	−10.22	+5.12	−20.14	−39.06
0.9	+2.40	−13.24	−10.65	+3.40	−16.42	−43.27
1.0	−1.18	−11.82	−5.89	+10.31	−18.32	−53.09

documents coming from many and very heterogeneous sources like net news articles, for which small samples can be easily inadequate to represent all possible information needs.

Tables 3 and 4 report the comparative increments in filtering performance, with regards to the base run, of three different training strategies. The three strategies are:

TT strategy: it consists of using and modifying the weights of only the topic terms (TT), that is only those terms present in the original topic and not learning new terms from the training data;

HF strategy: it consists of using and modifying the weights of the topic terms and of only high frequency terms (HF) in the training data; by HF terms we mean terms which occur in at least $\log_2 N$ documents in the training collection.

AT strategy: it consists of using and modifying the weights of all terms of the training data including the topic terms.

As we will show, there is no strategy that score consistently better than the others for different levels of training, but it will be necessary to move from one strategy to another with varying sizes of the training and with different user requirements in terms of his preferred combination of recall and precision levels.

Table 3 reports the results of the three training strategies using only relevant information. It can be noticed that the best results are achieved using only the topic terms, that is the TT strategy. The worst performance is obtained by the AT strategy, which seems to introduce a lot of noise by using all terms in the training data. The performance of the HF strategy lies in between, since less noise is introduced. However, we can notice that the HF strategy does improve the performance for low levels of recall, therefore helping us inferring that the terms introducing noise are the least frequent ones.

Table 4 shows the comparative performance of the three strategies using both relevant and nonrelevant information. Notice that now the performance of the AT strategy and HF strategy are better than before. In actual fact, they are better than those obtained by the TT strategy, given the same amount of information used. This can be explained by the contribution to the

Table 4
Precision increment in percentage w.r.t. the base run with a balanced set of relevant (R) and nonrelevant (N) documents

Recall	4R-4N-TT	8R-8N-TT	16R-16N-TT	16R-16N-HF	16R-16N-AT
0.0	+0.12	+2.50	+5.81	+9.11	+8.8
0.1	+0.2	+3.34	+14.43	+18.32	+19.07
0.2	+0.99	+0.96	+8.88	+17.21	+19.83
0.3	−0.14	+4.41	+10.03	+15.89	+18.60
0.4	+2.56	+2.96	+9.25	+9.72	+4.83
0.5	−0.39	+1.74	+9.12	+9.71	+0.78
0.6	−3.77	+2.22	+14.76	+11.25	+2.25
0.7	+2.71	+2.16	+21.70	+14.85	−1.57
0.8	+1.49	+0.48	+6.22	+4.76	−4.58
0.9	+2.36	+1.46	+16.14	+13.00	−1.44
1.0	+1.80	+2.61	−22.57	−12.49	−17.55

Table 5
Average precision and recall values after retrieving *K* documents for the run 16R-16N-HF

K	Precision (%)	Recall (%)
10	71.1	5.5
20	59.1	8.9
40	51.4	14.3
80	31.3	27.3

learning given by negative information (nonrelevant documents). This information helps in weighting in a better way the terms, and even terms not present in the topic come to play an active role in the training. However, we can still notice that not all terms can be useful in the training, since the AT strategy is still worse that the HF strategy. This helps us conclude that some terms can still be harmful in the training for high levels of recall.

By comparing the two tables we can notice that if we use the same amount of absolute information, we have better results by using all relevant information, especially for low amount of training. For example, if we compare the performance of using 16 relevant documents (column 16R-TT of Table 3) with the performance of using eight relevant and eight nonrelevant documents (column 8R-8N-TT), we have better performance when using 16 relevant documents, although this effects seems to reverse with a large number of documents in the training set. However, if we consider that the nonrelevant information can be obtained easily by randomly picking documents from the entire collection, given the very low probability of randomly selecting a relevant document, we could say that the nonrelevant information 'comes for free'. In this case it is fair to compare runs using the same amount of relevant information, like for example the two runs 16R-TT and 16R-16N-TT. From this comparison we can see that adding 'free' (or perhaps 'cheap') nonrelevant information increases the performance, in particular for low levels of recall. Similar conclusions can be obtained by comparing 4R-TT with 4R-4N-TT and 8R-TT with 8R-8N-TT. The conclusion that can be obtained for these data is that if the information need of an end user is stable in the long-term, learning is in general no faster using only relevant documents compared with using a balanced training set, that is a set containing both relevant and nonrelevant documents[2].

The results reported in Tables 3 and 4 show that the system is robust in learning new terms when the amount of relevant information is balanced by a similar amount of nonrelevant information. Indeed, the performance of the system is stable by using either HT terms or all terms (AT) in the training. In both cases the precision is significantly better than that obtained by tuning the terms TT in the topic. This allows us to conjecture that adopting ProFile learning algorithm can be successfully applied to filter a language-independent document source. This conjecture will have be checked carefully and this is one of the future directions of our work.

[2] The use of only nonrelevant documents in the training was experimented too, but as expected provided so poor results that we decided not to report them.

Table 5 reports the precision and recall figures at particular ranking points, that is after the user has inspected a number K of documents. The results reported refer to our best learning strategy, the 16R-16N-HF. It shows how many documents our users have to inspect to satisfy their precision and recall requirements. We chosen the value of K in realistic terms, that is we chose it close enough to the number of documents a user is really willing to inspect in real applications. Values higher than these (and 80 is already quite high a value) will be unrealistic. The results show that, after having been trained with as little as 32 documents, ProFile can already achieve quite good performance. Table 5 shows, for example, that among the first 10 documents retrieved by ProFile on average seven are relevant and that among the first 20 at least 11 are relevant. The user can then select anyone of the relevant and nonrelevant documents, mark them accordingly, and use them for the tuning phase, further improving the performance of the filtering. ProFile will balance the number of relevant documents marked by the user with nonrelevant document chosen at random from the collection, since the learning strategy employing a balanced combination of relevant and nonrelevant has proved to be the best strategy.

8. Conclusions and future works

In this paper we presented a probabilistic learning algorithm and its current implementation: the ProFile IF system. The first results of the evaluation of ProFile are encouraging and prove our theoretical conclusions. A more extensive evaluation is however needed, in particular with regards to finding the best possible learning strategies. We believe that many aspects of the training phase (i.e. the training data, the form of the initial topic, the combination of positive and negative training examples, etc.) depend on the application and on the document collection being used. To prove that, we intend to test ProFile using different collections of documents and in different application areas. The following two directions will be explored:

- *The use of ProFile for multilingual news filtering*. In this context it will be necessary to set a threshold on the ranked list of news items so that items above that level will be retrieved and presented to the user and those below it will be discarded. Setting such a threshold at an optimal level is not trivial, since it is user- and application-dependent.
- *Testing the learning algorithm with multiple levels of relevance*. In the evaluation presented in this paper ProFile learning only uses ‘binary’ information about the relevance of a document (a document is either relevant or not), because such was the information available for the TREC test collection. However, ProFile is capable of using more detailed information about the relevance of a document. We will test ProFile using test collections with documents classified according to several classes of relevance. Examples of such collections are: the Cystic Fibrosis Database with eight classes of relevance (Shaw, Wood, Wood, & Tibbo, 1991), the Cranfield test collection with five classes (Cleverdon, Mills, & Keen, 1966) and the STAIRS collection with six classes (Blair, 1996). With more precise relevance information we expect higher performance levels.

We believe our initial results, presented in this paper, provide a very good starting point for the above further experimentations.

Acknowledgements

We would like to thank Flavio Ubaldini and Stefano De Nardis for their invaluable help in the implementation and testing of the learning algorithm presented here. Thanks also to Keith van Rijsbergen for the many and interesting discussions and suggestions on the probabilistic models of IR. Finally, thanks to the anonymous reviewer who pointed out some of the unclear points of this paper. Amati's work was carried out in the framework of the agreement between the Fondazione Ugo Bordoni and the Italian PT Administration, while Crestani's was supported by a 'Marie Curie' Research Fellowship from the European Commission.

References

Aalbersberg, I. (1992). Incremental relevance feedback. In *Proceedings of ACM SIGIR, Copenhagen, Danmark* (pp. 11–22).

Allan, J. (1996). Incremental relevance feedback for information filtering. In *Proceedings of ACM SIGIR, Zurich, Switzerland* (pp. 270–278).

Amati, G., Crestani, F., Ubaldini, F., & De Nardis, S. (1997). Probabilistic learning for information filtering. *Proceedings of the RIAO Conference, Montreal, Canada*, *Vol. 1* (pp. 513–530).

Amati, G., D'Aloisi, D., & Giannini, V. (1995). A framework for dealing with email and news messages. In *Proceedings of AICA 95, Cagliari, Italy* (pp. 27–29).

Amati, G., & van Rijsbergen, C. (1995). Probability, information and information retrieval. In *Proceedings of the First International Workshop on Information Retrieval, Uncertainty and Logic, Glasgow, Scotland, UK.*

Bar-Hillel, Y., & Carnap, R. (1953). Semantic information. *British Journal of the Philosophy of Science*, *4*, 147–157.

Belkin, N., & Croft, W. (1992). Information filtering and information retrieval: two sides of the same coin? *Communication of the ACM*, *35*(12), 29–38.

Blair, D. (1996). STAIRS Redux: thoughts on the STAIRS evaluation, 10 years after. *Journal of the American Society for Information Science*, *47*(1), 4–22.

Callan, J. (1996). Document filtering with inference networks. In *Proceedings of ACM SIGIR, Zurich, Switzerland* (pp. 262–269).

Carnap, R. (1950). *Logical foundations of probability*. London, UK: Routledge and Kegan Paul Ltd.

Cleverdon, C., Mills, J., & Keen, M. (1966). *ASLIB Cranfield Research Project: factors determining the performance of indexing systems*. ASLIB.

Crestani, F., Lalmas, M., van Rijsbergen, C., & Campbell, I. (1998). Is this document relevant? ...Probably. A survey of probabilistic models in information retrieval. *ACM Computing Surveys*, *30*(4), in press.

Dunlop, M. (1997). The effect of accessing nonmatching documents on relevance feedback. *ACM Transactions on Information Systems*, *15*(2), 137–153.

Fidel, R., & Crandall, M. (1997). User's perception of the performance of a filtering system, Philadelphia, PA, USA. In *Proceedings of ACM SIGIR* (pp. 198–205).

Frakes, W. (1992). Stemming algorithms. In W. Frakes, & R. Baeza-Yates, *Information retrieval: data structures and algorithms*. Englewood Cliffs, NJ, USA: Prentice Hall (Ch. 8).

Ghosh, G. (1991). *A brief history of sequential analysis*. New York, USA: Marcel Dekker.

Harman, D. (1992). Relevance feedback and other query modification techniques. In W. Frakes, & R. Baeza-Yates, *Information retrieval: data structures and algorithms*. Englewood Cliffs, NJ, USA: Prentice Hall (Ch. 11).

Harman, D. (1993). Overview of the first TREC conference. In *Proceedings of ACM SIGIR, Pittsburgh, PA, USA* (pp. 36–47).

Harman, D. (1996). Overview of the fifth text retrieval conference (TREC-5). In *Proceeding of the TREC Conference, Gaithersburg, MD, USA.*

Harper, D., & van Rijsbergen, C. (1978). An evaluation of feedback in document retrieval using cooccurrence data. *Journal of Documentation*, *34*(3), 189–216.

Hintikka, J. (1970). On semantic information. In *Information and inference*. Dordrecht, The Netherlands: Synthese Library, Reidel.

Hull, D. (1998a). The TREC-6 filtering track: description and analysis. In *Proceeding of the TREC Conference, Washington, DC, USA* (pp. 45–67).

Hull, D. (1998b). Guidelines for the TREC-7 filtering track. In *Notebook of the TREC Conference, Washington, DC, USA*.

Jeffrey, R. (1965). *The logic of decision*. New York, USA: McGraw-Hill.

Kilander, F. (1995). A brief comparison of news filtering software. Unpublished paper.

Lang, K. (1995). NewsWeeder: learning to filter netnews. In *Proceedings of ML 95* (pp. 331–339).

Lewis, D. (1995). A sequential algorithm for training text classifiers: corrigendum and additional data. *Sigir Forum*, *29*(2), 13–19.

Lewis, D., & Gale, W. (1994). A sequential algorithm for training classifiers. In *Proceedings of ACM SIGIR, Dublin, Ireland* (pp. 3–11).

Maron, M. (1961). Automatic indexing: an experimental inquiry. *Journal of the ACM*, *8*, 404–417.

Renyi, A. (1969). *Foundations of probability*. San Francisco, USA: Holden-Day Press.

Robertson, S., & Sparck Jones, K. (1976). Relevance weighting of search terms. *Journal of the American Society for Information Science*, *27*, 129–146.

Robertson, S., Walker, S., Hancock-Bealieu, M., Gull, A., & Lau, M. (1993). Okapi at TREC. In *Proceeding of the TREC Conference (TREC-1), Gaithersburg, MD, USA* (pp. 21–34).

Robertson, S., Walker, S., Hancock-Bealieu, M., Jones, S., & Gatford, M. (1995). Okapi at TREC-3. In *Proceeding of the TREC Conference (TREC-3), Gaithersburg, MD, USA* (pp. 109–126).

Salton, G., & McGill, M. (1983a). *Introduction to modern information retrieval*. New York: McGraw-Hill.

Salton, G., & McGill, M. (1983b). *The SMART retrieval system: experiments in automatic document retrieval*. Englewood Cliffs, USA: Prentice Hall Inc.

Shaw, W., Wood, J., Wood, R., & Tibbo, H. (1991). The Cystic Fibrosis Database: content and research opportunities. *LISR*, *13*, 347–366.

Turtle, H. (1990). *Inference networks for document retrieval*. Ph.D. thesis, Computer and Information Science Department, University of Massachusetts, Amherst (USA).

van Rijsbergen, C. (1979). *Information retrieval* (2nd ed.). London: Butterworths.

Yan, T., & Garcia-Molina, H. (1995). SIFT: a tool for wide-area information dissemination. In *Proceedings of the 1995 USENIX Technical Conference* (pp. 177–186).