

Probabilistic Learning for Information Filtering

Gianni Amati

Fondazione Ugo Bordoni
Roma, Italy

Fabio Crestani*

Department of Computing Science
University of Glasgow
Glasgow, Scotland

Flavio Ubaldini

Fondazione Ugo Bordoni
Roma, Italy

Stefano De Nardis

Dipartimento di Informatica e Sistemistica
Università di Roma “La Sapienza”
Roma, Italy

Abstract

In this paper we describe and evaluate a learning model for information filtering which is an adaptation of the generalised 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 non relevant documents. The proposed learning model is the core of a prototype information filtering system called *ProFile*.

*Previously at Dipartimento di Elettronica e Informatica, Università di Padova, Padova, Italy.

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 information. Users may receive large amounts of information electronic, like for example electronic mail or news, and systems for *Information Filtering* (IF) are required to select from a large amount of incoming documents only those relevant to some user information need.

Information Filtering is concerned with determining the information relevant to the user. The representation of the user's information need may consist of a set of possibly weighted keywords given by the user or induced by the system, the so called *user profile*. Another way of considering the user profile is to consider it as a description of the user's interests. When a user has more than one interest and would like to have documents classified into different classes representing this different interests, then it is preferable to talk about *class profiles*.

Information Filtering and Information Retrieval (IR) have been described as two faces of the same coin [6], because many of the underline issues are the same. 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 recently by Callan [8] and by the proposer of the TREC-5 Filtering track [14]. The idea is different techniques are required in order to design effective IF and IR systems. In particular, IF requires more sophisticated techniques of learning through relevance feedback than IR, since it is important to be able to model the user information need with the most efficient use of the information the user provides. An IF system that would require a long and painful training cannot be considered effective despite its filtering performance. The most effective IF system is the one that requires little training to perform reasonably well and that can be easily tuned by the user in an interactive way.

In this paper we describe a learning model for IF which is an adaptation of the generalised probabilistic model of Information Retrieval (IR) [4]. 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 [13]. The second class contains those models which allows for relevance feedback also on the uncertain documents which were not considered [21]. Our model belongs to this last 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 [21, 20]. Our results indeed show that we need very few documents in the training set to have good performance.

In the rest of the paper we describe and evaluate the learning algorithm of our IF system: ProFile. Section 2 describes the current implementation of ProFile. Section 5 relates Pro-

File with other IF systems and other research on the use of learning algorithm in IR. Section 3 describes in detail the probabilistic learning model at the heart of ProFile. Finally, in Section 6, we report the results of an experimental investigation into the effectiveness of ProFile.

2 ProFile

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 [3]. 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.

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 for 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 satisfy the relation, then it is classified into that class, otherwise it is discarded. If a document satisfy relations with more than one class, then it is either classify 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 [5, 16] and decision theory [17].

ProFile operates according to the following steps:

1. *Definition of the conceptual classes.* The user defines a set of 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 service (Fub Information Filtering Tool) [3], a customised version of SIFT, a filtering system developed at Stanford (see Section 5). 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’s relevance feedback process uses the probabilistic

learning model that will be describe in detail in Section 3. The pre-filtering phase can go on as long as the user requires, with as many retrieval runs (performed by FIFT) and user relevance feedback as the user chooses.

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 sub-phases:
 - (a) *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.
 - (b) *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.

It should be noticed that the initial training phase is very important for the effectiveness of ProFile. Indeed, in the limit case of no relevant document 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 non-relevant ones. *Recall* (R) is 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 observed that the best training set is obtained when the relevance values are equally distributed. Our way of training the system can be assimilated to the uncertainty sampling [21, 20]. In [21], Lewis and Gale observed better performance in IF using uncertainty sampling instead of relevance sampling [12], in particular when the sample size is small in comparison with the number of positive examples in the set of non-evaluated 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 the context of this paper we intend to evaluate the performance of our learning model, in particular when little training data is provided. Moreover, 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 non relevant. In IR the use of negative data in relevance feedback has been received with contrasting views. Salton considered it positively [25], while other researchers considered it dangerous [1] or even harmful [11]. We believe that it all depends on the particular retrieval model one is using. We intend to prove that our

model make 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.

3 A Probabilistic learning model for IR

In this section we describe in detail our probabilistic learning model. The model is derived from the generalised probabilistic model of IR presented in [4].

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, indexes, words or lexical units). The semantic relations between terms in the set \mathcal{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 [23] for IF is a triple $\langle \Omega, \mathcal{A}, \mathcal{P} \rangle$. 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. \mathcal{A} is the power set of Ω , namely the set of all subsets of Ω . \mathcal{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 \mathcal{A} by using the additivity axiom.

In a finite space, a probability can be then obtained by conditioning. The *conditioning* of \mathcal{P} is defined as:

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

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 $t \in \mathcal{T}$. 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 the *tf* weighting (the relative frequency of t in d) or the *idf* weighting (defined as $idf(t) = -\log(n/N)$, where n is the number of documents in which t occurs and N is the number of documents in the collection) [25].

In other words if we denote by $\langle a_i^d \rangle_{d \in \Omega, t \in \mathcal{T}}$ the matrix $\langle t(d) \rangle_{d \in \Omega, t \in \mathcal{T}}$, then a row associated to d is the vector $\langle t \rangle_{t(d) \in \mathcal{T}}$ made out of the statistics of the set of terms in the document d , while the random variables $t \in \mathcal{T}$ are obtained by the columns of the matrix. In IR the matrix $\langle t(d) \rangle_{d \in \Omega, t \in \mathcal{T}}$ 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 \mathcal{P} as:

$$E_{\mathcal{P}}(t) = \frac{\sum_{d \in \Omega} t(d) \mathcal{P}(d)}{\mathcal{P}(\Omega)} \quad (1)$$

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

In [4], an IR model is introduced as follows. \mathcal{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 $[0, 10]$ 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 non-relevance, $\neg R(d) = 1 - R(d)$, 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 non-relevance\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 non-relevance 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 [4], that is:

$$EMV(t) = \lambda_1 E_R(t) + \lambda_2 E_{\neg R}(t) \quad (2)$$

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 $EMV(t) > 0$.

EMV can be equivalently given by the formula:

$$EMV1(t) = \log \frac{\lambda_1 * E_R(t)}{\lambda_2 * E_{\neg R}(t)} \quad (3)$$

Decision theory and semantic information

Since the fifties the concept of information has been central in communication theory. Hintikka [16] 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* [9, 5]. 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 characterisation 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 generalise Hintikka’s semantic information theory [16] 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 [28] 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 $E_{-R}(t)$). In the above formula what we actually gain or lose in taking t is unclear. But if “ t is relevant”, then the user will gain the amount of information of non-relevance of t : let us denote it by $Inf_{-R}(t)$. On the other hand, the loss λ_2 can be quantified by the amount of information of relevance of t , that is $Inf_R(t)$. In both information theories (semantic and frequency-based) the amount of information is taken to be inversely proportional to probability, that is $Inf_{\mathcal{P}}(e) = -\log \mathcal{P}(e)$ 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 [16], following Carnap’s semantic information theory, 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 formalised as $inf(e) = 1 - \mathcal{P}(e)$. In our case we have to assign the amount of information to random variables instead to sentences. By analogy, following Jeffrey’s suggestion [17] and observing that the conditioning expectations do not go beyond the value 1, we may define the amount of information as:

$$Inf_{\mathcal{P}}(t) =_{def} 1 - E_{\mathcal{P}}(t)$$

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

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

and

$$Inf_R(t) = 1 - E_R(t) = E_R(\neg t)$$

Substituting the values of the λ 's into (3), we have

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

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

$$w(t_i) = \log \frac{E_R(t_i) * E_{\neg R}(\neg t_i)}{E_R(\neg t_i) * E_{\neg R}(t_i)} > 0 \quad (4)$$

The probabilistic model of IR

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

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 non relevant 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 $\sum_{d \in \Omega} a_i^d R(d)$ is the cardinality r^i of the set of relevant document in which the term t_i occurs. Substituting r^i into (1) we get $E_R(t_i) = \frac{r^i}{n_R}$.

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 (4) satisfies the following relation:

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

This is the the well known weighting formula of the probabilistic model of IR [24, 28].

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 the user's profile we would not take into account neither the non-relevant 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 (4) generalises the probabilistic model of IR.

4 ProFile's learning model

Let us now define 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, \dots, 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 Equation (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 [25]. For the inner product, for example, we would get the equation:

$$SIM(x, E_{R_C}(t)_{t \in \mathcal{T}}) = \frac{\frac{\sum_t x_t \sum_{d \in \Omega} a_d^t r_d^C}{R_C(\Omega)}}{\frac{\sum_t \sum_{d \in \Omega} a_d^t r_d^C}{R_C(\Omega)}} = \frac{\sum_t \sum_{d \in \Omega} x_t a_d^t r_d^C}{\sum_t \sum_{d \in \Omega} a_d^t r_d^C} \quad (6)$$

where $R_C(d)$ is denoted by r_d^C . Note that in the above formula r_d^C 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_d^t) and the vector (r_d^C) to be able to compute the similarity function, that is $(|\mathcal{T}| + n) \cdot |\Omega|$ 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 equation (1) and incrementally updating this measure as soon as a new document is processed. In this way we need to store $(1 + |\mathcal{T}|) \cdot n$ global parameters, that is the values $R_C(\Omega_{old})$ and $E_{R_C}^{old}(t)$. Suppose now that a new document $y = \langle y(t) \rangle_{t \in \mathcal{T}}$ is incoming, so that $\Omega_{new} = \Omega_{old} \cup \{y\}$. Then the relation among the new values, $E_{R_C}^{new}(t)$ and $R_C(\Omega_{new})$, and the old values, $E_{R_C}^{old}(t)$ and $R_C(\Omega_{old})$, is ruled by the following transition equations, derived from the equation (1) and by the definition of Ω_{new} :

$$E_{R_C}^{new}(t) = \frac{E_{R_C}^{old}(t)R_C(\Omega_{old}) + y_t r_y^C}{R_C(\Omega_{old}) + r_y^C} \quad (7)$$

$$R_C(\Omega_{new}) = R_C(\Omega_{old}) + r_y^C \quad (8)$$

Applying some algebra to equation (1) we easily get the non-relevance parameters for t :

$$\begin{aligned} E_{\neg R_C}(t) &= \frac{\sum_{d \in \Omega} a_d^t \neg R_C(d)}{\sum_{d \in \Omega} \neg R_C(d)} = \frac{\sum_{d \in \Omega} a_d^t (1 - r_d^C)}{\sum_{d \in \Omega} (1 - r_d^C)} = \\ &= \frac{\sum_{d \in \Omega} a_d^t - \sum_{d \in \Omega} a_d^t r_d^C}{|\Omega| - R_C(\Omega)} = \frac{\sum_{d \in \Omega} a_d^t - E_{R_C}(t) R_C(\Omega)}{|\Omega| - R_C(\Omega)} \end{aligned}$$

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

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

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

$$|\Omega_{new}| = |\Omega_{old}| + 1 \quad (10)$$

$$a_{new}^t = a_{old}^t + y_t \quad (11)$$

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

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

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

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

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

To summarise, 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. $(|Terms| + 1)(n + 1)$ 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 equation (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 equation (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 the equations (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: $SIM(x, w_{C_j}) > s_C$ 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.

5 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 [6].

The probabilistic model of IR combine frequency values with relevance assessments using the Bayes' theorem. In [28] relevance as well as the set of terms are taken as elementary events. On the contrary, in [22] 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 [19] 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. This method is known as the complement method [15]. 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 $tf \cdot idf$ (term-frequency multiplied by inverse-document-frequency, see [25]) 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 [29] 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 [4] a learning model proposes a natural interpretation of relevance as well as a way to amalgamate it with rank-frequencies theory. This is the model used by ProFile and described in Section 4.

In SMART [26] 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 the null or the certain values). In ProFile the relevance feedback consists in 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 the filtering process is required. This makes

a non-trivial 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 the user's interests each night by taking into account the overall history of the user's relevance assignment on the training documents which must be saved and kept for each user as a profile. In [19] 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 comparison of ProFile with the many IF systems proposed in the last few years is outside the scope of this section. In recent years a large number of IF systems have been proposed. One application area that has been heavily targeted is news filtering [18]. 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 [14], for example). The area of IF brings together many different experiences from other areas, like machine learning, data mining, knowledge representation, and so on. 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 evaluating ProFile in a almost pure TREC-style, as reported in the next section.

6 Evaluation

In the context of the work reported in this paper we intended to evaluate the performance of our IF learning model, in particular when little training data is provided.

The collection we used is the *TREC-5 B* [14] 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 3 years (1990-92) of selected full text articles of the Wall Street Journal. The total number of documents (articles) in the collection is about 75.000. Each document is about 550 words in length. The size of the collection is about 260Mbyte. This is quite a large collection for IF and IR standards. We also used a set of 50 already prepared 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.

Recall	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Precision	0.54	0.4	0.31	0.25	0.20	0.15	0.10	0.06	0.03	0.01

Table 1: Performance of ProFile for the base run.

The evaluation was performed in true IR style, since this is the current practice for IF systems (see the evaluation methodology used in the various TREC conferences). The main retrieval effectiveness measures used in IR are Recall and Precision, already defined in Section 2. We want just to remind the reader that, experimentally, these measures have proved to be related in such a way that high precision brings low recall and viceversa. In other words, if one desires high precision, he has to accept low recall, and viceversa. 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. The training data of each run was a subset of up to 32 documents among 32 relevant and 32 non relevant documents, randomly chosen. The filtering runs shown in Tables 1, 2, 3 and 4 are thus incremental. Table 1 reports the base run, that has been performed using only the information provided by the text of the topics without any additional information. It is important to keep in mind that for all the 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 quite low performance compared with those and IR system could have produced. Moreover, we only used a simple stop list (list of term not to be used in the indexing) and we did not employ any stemming function (a function that reduces words to stems), since we wanted the system to be language independent. Although with these settings we considerably reduced the retrieval effectiveness compared with IR techniques, we believed we should mimic, as far as possible, the normal situation in which many IF systems work, e.g. IF systems for the net news. The hypothesis we made was that the system could know and process in advance the incoming data. A different approach was followed by Allan in [2]. Allan determined statistical information about the full collection by generalising the statistical information extracted from a sample of relevant and non relevant documents, that is documents processed up to a give time. Of course this technique works the best the largest the sample.

In order to evaluate ProFile in the closest possible way to the real use of a IF system, we considered three fictitious users:

User A demands a high precision performance from the system and is happy with low

User	8R-TT	16R-TT	32R-TT	32R-HF
A	+ 6.3%	+13.0%	+13.8%	+ 9.6%
B	+ 5.4%	+13.1%	+13.5%	-9.0%
C	+25.6%	+28.1%	+32.5%	+2.2%

Table 2: Precision increment w.r.t. the base run by using only relevant documents (R) as training. AT = all terms of the training data and of the topic in the profile, TT = only terms in the topic, HF = only the high frequency terms and terms in the topic.

recall performance (a recall value 0.3, that is 30% of the total number of relevant documents in the collection);

User B is the average user that requires average levels of recall and precision (a recall of 0.5);

User C wants to retrieve most of the relevant information stored in the collection (a recall value of at least 0.8) and accept that the system will also retrieve a lot of non relevant documents.

One should notice that the ideal case of having high precision together with high recall is not realistic with the current state of the IF technology.

Table 2 and Table 3 show that the learning, considered as expansion of the current topic and restricted to only highly frequent terms (HF) in the training data, should be done with a balanced set of training data. By considering 32 relevant document, the relative run gave worse performance than that with 16 relevant and 16 non relevant documents. Even if we restrict the learning to tune the weights of the topic terms, the Tables also shows 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 non relevant document (notice the better behavior respectively of the runs 8R-8N and 16R-16N-AT with respect to the runs 8R and 16R, which have the same number of relevant documents). In this particular case, negative examples (non relevant documents) are neither harmful nor useless when combined with positive information for not high values of recall. However, a training set made up of only negative examples do not contribute much in the tuning phase since many terms will not be present in the topic.

Even though the topics were long and complex, the results show that few training documents improve substantially the performance of the system for high recall values, hence a short tuning phase is indeed useful especially when there are diverse document sources and they are not known in advance.

Tables 2 and 3 show that with little training it is possible to increase considerably the

User	4R-4N-TT	8R-8N-TT	16R-16N-TT	16R-16N-HF
A	+6.7%	+10.9%	+8.9%	+23.1%
B	+7.5%	+10.8%	+13.2%	+19.5%
C	+25.2%	+28.5%	+35.3%	+34%

Table 3: Precision increment w.r.t. the base run with a balanced set of relevant (R) and non relevant (N) documents.

K	Precision	Recall
10	70%	5.4%
20	59%	8.9%
40	51.4%	14.2%
80	31.2%	27.3%

Table 4: Average precision and recall values after retrieving K documents for the run 16R-16N-HF.

performance for user B and C. With very little training compared with the size of the collection (8 or 16 documents out of about 75.000) there is a high increase in precision at high levels of recall. This means that the users are getting more and more relevant documents.

As for low values of recall and high values of precision, the requirement of user A, results show that the system needs a longer phase of learning (at least 20 – 30 relevant documents). Nevertheless, it has been shown by Allan in [2] the use of a subset of 10% of the relevance judgments (about 8.000 documents over 90.000) for learning works quite well with respect to the *full* set. However, Allan uses the training set, which is made up of several thousand of documents, to evaluate the *idf* function. The *idf* function is indeed decisive for improving precision for low values of recall, but conversely a large amount of information is required.

Table 4 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 chosen it closed 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 ProFile after having been trained with as little as 32 documents, can achieved quite good performance. Table 4 shows, for example, that among the first 10 documents retrieved by ProFile on average 7 are relevant, and that among the first 20 at least 11 are relevant. The user can then select anyone of the

relevant or non relevant documents retrieved, mark them accordingly, and use them for the tuning phase, further improving the performance of the filtering. A learning strategy employing a balanced combination of relevant and non relevant has proved to be the best strategy.

7 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 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 information rich relevance feedback.* 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 8 classes of relevance [27], the Cranfield test collection with 5 classes [10], and the STAIRS collection with 6 classes [7]. With more precise relevance information we expect higher performance levels.

Acknowledgments

We would like to thank Keith van Rijsbergen for the many and interesting discussions and suggestions on the probabilistic models of Information Retrieval. Thanks also to Mark Sanderson for his help in the evaluation.

References

- [1] I.J. Aalbersberg. Incremental relevance feedback. In *Proceedings of ACM SIGIR*, pages 11–22, Copenhagen, Danmark, jun 1992.
- [2] J. Allan. Incremental relevance feedback for information filtering. In *Proceedings of ACM SIGIR*, pages 270–278, Zurich, Switzerland, August 1996.
- [3] G. Amati, D. D’alosi, and V. Giannini. A framework for dealing with email and news messages. In *Proceedings of AICA 95*, pages 27–29, Cagliari, Italy, September 1995.
- [4] G. Amati and C.J. van Rijsbergen. Probability, information and information retrieval. In *Proceedings of the First International Workshop on Information Retrieval, Uncertainty and Logic*, Glasgow, Scotland, UK, September 1995.
- [5] Y. Bar-Hillel and R. Carnap. Semantic information. *British Journal of the Philosophy of Science*, 4:147–157, 1953.
- [6] N.J. Belkin and W.B. Croft. Information Filtering and Information Retrieval: two sides of the same coin? *Communication of the ACM*, 35(12):29–38, 1992.
- [7] D.C. Blair. STAIRS Redux: thoughts on the STAIRS evaluation, ten years after. *Journal of the American Society for Information Science*, 47(1):4–22, 1996.
- [8] J. Callan. Document filtering with inference networks. In *Proceedings of ACM SIGIR*, pages 262–269, Zurich, Switzerland, August 1996.
- [9] R. Carnap. *Logical Foundations of probability*. Routledge and Kegan Paul Ltd, London, UK, 1950.
- [10] C. Cleverdon, J. Mills, and M. Keen. *ASLIB Cranfield Research Project: factors determining the performance of indexing systems*. ASLIB, 1966.
- [11] M.D. Dunlop. The effect of accessing non-matching documents on relevance feedback. *ACM Transactions on Information Systems*, 1997. (Forthcoming).
- [12] G. Ghosh. *A brief history of sequential analysis*. Marcel Dekker, New York, USA, 1991.
- [13] D. Harman. Relevance feedback and other query modification techniques. In W.B. Frakes and R. Baeza-Yates, editors, *Information Retrieval: data structures and algorithms*, chapter 11. Prentice Hall, Englewood Cliffs, New Jersey, USA, 1992.
- [14] D. Harman. Overview of the fifth text retrieval conference (TREC-5). In *Proceeding of the TREC Conference*, Gaithersburg, MD, USA, November 1996.

- [15] D.J. Harper and C.J. van Rijsbergen. An evaluation of feedback in document retrieval using co-occurrence data. *Journal of Documentation*, 34(3):189–216, September 1978.
- [16] J. Hintikka. On semantic information. In *Information and inference*. Synthese Library, Reidel, Dordrecht, The Netherlands, 1970.
- [17] R.C. Jeffrey. *The logic of decision*. McGraw-Hill, New York, USA, 1965.
- [18] F. Kilander. A brief comparison of news filtering software. Unpublished paper, June 1995.
- [19] K. Lang. NewsWeeder: learning to filter netnews. In *Proceedings of ML 95*, pages 331–339, 1995.
- [20] D.D. Lewis. A sequential algorithm for training text classifiers: corrigendum and additional data. *SIGIR FORUM*, 29(2):13–19, 1995.
- [21] D.D. Lewis and W.A. Gale. A sequential algorithm for training classifiers. In *Proceedings of ACM SIGIR*, pages 3–11, Dublin, Ireland, July 1994.
- [22] M.E. Maron. Automatic indexing: an experimental inquiry. *Journal of the ACM*, 8:404–417, 1961.
- [23] A. Renyi. *Foundations of probability*. Holden-Day Press, San Francisco, USA, 1969.
- [24] S.E. Robertson and K. Sparck Jones. Relevance weighting of search terms. *Journal of the American Society for Information Science*, 27:129–146, May 1976.
- [25] G. Salton and M.J. McGill. *Introduction to modern Information Retrieval*. McGraw-Hill, New York, 1983.
- [26] G. Salton and M.J. McGill. *The SMART retrieval system - experiments in automatic document retrieval*. Prentice Hall Inc., Englewood Cliffs, USA, 1983.
- [27] W.M. Shaw, J.B. Wood, R.E. Wood, and H.R. Tibbo. The Cystic Fibrosis Database: content and research opportunities. *LISR*, 13:347–366, 1991.
- [28] C.J. van Rijsbergen. *Information Retrieval*. Butterworths, London, second edition, 1979.
- [29] T.W. Yan and H. Garcia-Molina. SIFT - a tool for wide-area information dissemination. In *Proceedings of the 1995 USENIX Technical Conference*, pages 177–186, 1995.