

# A Learning System for Selective Dissemination of Information

Gianni Amati

Fondazione Ugo Bordoni  
via B. Castiglione, 59  
I-00142 Rome, Italy

Fabio Crestani\*

Dept. of Computing Science  
University of Glasgow  
Glasgow G12 8QQ, Scotland

Flavio Ubaldini

Fondazione Ugo Bordoni  
B. Castiglione, 59  
I-00142 Rome, Italy

## 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 a user's interest 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 present a learning system for information filtering and selective information dissemination. The learning algorithm is described and the effectiveness of the system is evaluated in a true information filtering style.

## 1 Introduction

Information overload is an increasing problem in many domains. New information services (e.g. news services, electronic mail, libraries and databanks) deliver to the user an increasing volume of digital information. Often the information delivered to the user does not match the user interest and it ends up overloading the user, who will have to manually select the interesting information from the "noise". It is true that the user could switch off such automatic delivery of information, but in that case she will have to go after the information herself. In such a dynamic environment it is also difficult for the user, equipped with only conventional search capabilities, to keep up with the fast pace of information generation. Instead of making the user go after the information or having to go through the large amount of incoming information, information should more selectively flow to the interested user. Traditionally, libraries and databanks, provide such kind of service to user. A user is asked to provide a description of the classes of document in which she is interested. Such descriptions are then used as static queries that are submitted to a system that send to a passive user the documents matching

the queries. Such systems are becoming increasingly important and will form an indispensable tool for global information systems.

In this paper we present a system called ProFile that learns by interacting with the user what is the user interest. ProFile constructs and adaptively tunes a representation of the classes of user's interests using a learning algorithm that is derived from the generalised probabilistic model of IR presented in [Amati and van Rijsbergen, 1995]. ProFile selectively distributes documents from a continuous stream to multiple users with several classes of interests. The fast learning and adapting capabilities of ProFile enables it to effectively perform information filtering and dissemination.

## 2 Selective Dissemination of Information

*Document filtering*, also known as *selective dissemination of information*, has a long history, most of it based on the unranked Boolean retrieval model of information retrieval (IR) [van Rijsbergen, 1979]. Most of the recent research on document filtering is based on the assumption that effective IR techniques are also effective document filtering techniques. The TREC conference (see [Harman, 1996] for the last TREC conference) is a good example of this practice.

Recently, the term *information filtering* (IF) has started being used in place of the old style document filtering, to emphasise the possibility of selectively distributing multimedia information. In the context of this paper we will use this term too, since the technique here presented can also be used to perform multimedia document filtering and dissemination.

In IF there is the twofold problem of determining what information is relevant to any user and how this decision can be automatically taken by the information system. Parametric and qualitative descriptions of what information is of her interest must be generated. These factors constitute on the whole what is generally referred to as the *user profile*, but what is needed for expressing them may be difficult to circumscribe. The user profile may consist of a set of keywords, that represent the topics of user's interest, similar to the one generated by the in-

---

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

dexing process in IR. However this representation may result insufficient because the user could also need to know some additional modalities to which information is related, such as its novelty, urgency or purpose. Therefore it is not clear what factors are important for predicting the relevance of information to user's interests. However some limiting hypotheses on the type of user and/or data sources should be done to present a workable model of information filtering. Users may range from casual to specific ones and data types may vary from unstructured (images or "textual" data) to structured ones (such as relational tables). In the following we assume that the system is able to process the data and to provide them with precise (quantitative) descriptors of their contents, that is with a derivative structured information. Such data are called of *semi-structured type* [Belkin and Croft, 1992]. This is what typically happens with the indexing of textual data in IR: by using different techniques it is possible to assign sets of keywords to the data items. These sets are then organised into a matrix made out of vectors of weighted descriptors: weights are computed on the basis of a statistical analysis of the occurrences of words.

A second assumption upon which our work is based is that the user is *casual*, that is her profile is not pre-defined by some specific keywords chosen by the user in a *controlled language*, that is among a set of possible terms. On the contrary, we assume that the system does not necessarily know any initial definition of the user's profile. Rather, this is long-term defined. The underlying idea is that the system is trained by the user herself and that her profile "converges" to a stable description as soon as the number of the interactions with the information sources has become "large enough". This requirement for the user model makes a non-trivial difference with IR where the system is concerned with a single session at a time. Differently from IF, IR does not depend on time, though user's relevance feedback introduces an implicit temporal factor. Indeed, by adapting a variant of van Rijsbergen's model of IR to IF, as described in [Amati and van Rijsbergen, 1995], we are able to define a new model of information filtering. In particular we will show that the profile of a casual user consists of a vector whose descriptors are in the set of *uncontrolled* terms of the sources data.

### 3 Some Considerations about Related Work

Before considering the crux of the IF, we would like to make some remarks on how statistics may influence an IF model. Useful theoretical tools, that range from elementary probability theory to decision theory and statistical methods, are generally used to draw inferences for processing language and analysing linguistic structures. An example of stochastic model for indexing in IR is that based on the *expected mutual information measure* (EMIM) of van Rijsbergen's [van Rijsbergen, 1979], derived from Shannon's theory of information. This model

was used to find out the word associations in the English language for the last edition of the Collins' dictionary [Church and Hanks, 1989]. Here a word is interpreted as a "message" which carries a measure of uncertainty (entropy) defined in accord to the probability laws. This entropy is maximum when the message is unclear while is null when the message is "free" from any noise. An alternative approach is *vector space model* (VSM) by Salton [Salton and McGill, 1983] based on the Zipf's rank-frequencies law. In [Amati and van Rijsbergen, 1995] we have explored the connection of our model with both the EMIM and the VSM. In this paper we will outline the connections of our IF model with IR models rather than arguing how statistics can be better applied to IF. For the purpose of this paper, which is merely concerned with the modelling user's behaviour in IF, we will not focus on whether information theory or rank-frequencies is the best representation of the document content. Indeed, we assume the system to have the "best" statistical method for drawing linguistic knowledge from a sample of data. Our model is predictive of the behaviour of the user rather than explicative of the content of documents in the sample.

A large number of systems have been currently developed to filter information, and special efforts are devoted to filter the Internet messages. Despite of this explosion, results concerning the evaluation of these systems are rarely available. The evaluation is generally left to the end users. As an example the NewsWeeder [Lang, 1995] has been evaluated by monitoring for a year two users which used a very large amounts of training data. Only two precision values w.r.t. the set of selected documents are given (44% and 59%) and the relative recall values are not available. Hence, very little knowledge is available on whether and how fast the end users experience the filter.

## 4 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 *et al.*, 1995]. Despite being born with the purpose of filtering news, ProFile can be adapted to filter any incoming stream of information, like email, newswires, or newspaper articles.

### 4.1 The ProFile Architecture

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 in ranking documents according to a similarity values between 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 chosed (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 and Carnap, 1953; Hintikka, 1970] and decision theory [Jeffrey, 1965]. ProFile operates according to the following procedure:

*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 description of each conceptual class.

*Training phase.* The initial description of the user interests is utilised as a query by the FIFT service (Fub Information Filtering Tool) [Amati *et al.*, 1995], a customised version of SIFT, a filtering system developed at Stanford (see Section 3). 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 4.2. 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.

*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:

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.
2. *Tuning.* The user can either accept the result of the filtering and let ProFile keep working with the current profiles or otherwise he 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 the filter. 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. We observed that the best training set is obtained by balancing the number of relevant and non-relevant documents present in the training set (see Section 5). Our way of training the system can be assimilated to the uncertainty sampling. In [Lewis and Gale, 1994], Lewis and Gale observed a better performance in using uncertainty sampling instead of relevance sampling [Ghosh, 1991] 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.

## 4.2 The Information Filtering Model

In this section we describe in detail our probabilistic learning model. The model is derived from the generalised probabilistic model of IR presented in [Amati and van Rijsbergen, 1995].

### 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  $\tau$ . 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 for IF is a triple  $\langle \Omega, \mathcal{A}, \mathcal{P} \rangle$ .  $\Omega$  depends on a temporal parameter  $\tau$ ,  $\Omega(\tau)$  being the set of all documents processed before the time  $\tau$ . Here we assume that  $\Omega$  is the set of documents which have constituted the data stream up to the current moment, so that  $\tau$  can be omitted.  $\mathcal{A}$  is the power set of  $\Omega$ , namely the set of all subsets of  $\Omega$ .  $\mathcal{P}$  is defined by the user starting from the mutually exclusive elementary events, that is the elements  $d$  of  $\Omega$ . 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, provided  $\mathcal{P}(e_2) > 0$  as:

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

Functions defined from  $\Omega$  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) [Salton and McGill, 1983]. In ProFile we use *tf* since *idf* needs a complete information on the set of incoming data which is unrealistic in filtering or require a high expensive processing.

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  $\Omega$ . 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 [Amati and van Rijsbergen, 1995], an IR model is introduced as follows.  $\mathcal{P}$  corresponds to a subjective measure  $R$  of relevance on the event space  $\Omega$ , 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 real numbers in the interval  $[0, 1]$ , 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  $\Omega$ .

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 [Amati and van Rijsbergen, 1995], that is:

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

where  $\lambda_1$  is the "gain" when  $t$  is relevant to the user, while  $\lambda_2$  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

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.  $\lambda_1$  would be then the "award" if he takes  $t$  while  $\lambda_2$  would be the "cost" if he discards  $t$  (with a priori probability  $E_{\neg R}(t)$ ). If "t is relevant", then the user will gain the amount of information of non-relevance of  $t$ : let us denote it by  $Inf_{\neg R}(t)$ . On the other hand, the loss  $\lambda_2$  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 [Hintikka, 1970] 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, that is, 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:  $Inf_{\neg R}(t) = 1 - E_{\neg R}(t) = E_{\neg R}(\neg t)$  and  $Inf_R(t) = 1 - E_R(t) = E_R(\neg t)$ . Substituting the values of the  $\lambda$ 's into (3), we have the *absolute relevance of the term*, which 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 above model derives the probabilistic model of IR [Robertson and Sparck Jones, 1976; van Rijsbergen, 1979]

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

under the hypothesis that: (a)  $R$  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; (b)  $a_i^d$  is the *counting document-term matrix*, that is:  $a_i^d = 1$ , if the  $i$ -th term occurs in  $d$ , and  $a_i^d = 0$  otherwise.

In the formula  $n_R$  denotes the cardinality of the relevant set of documents,  $N$  the cardinality of  $\Omega$ ,  $r^i$  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.

More generally,  $w(t)$  can be used as a weight of relevance of the term  $t$  for the user: 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.

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 by using the weights:

$$w_C(t) = \log \frac{E_{R_C}(t)E_{\neg R_C}(\neg t)}{E_{R_C}(\neg t)E_{\neg R_C}(t)} \quad (6)$$

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. By applying the decision theory we are able to provide a term  $t$  with a weighting formula  $w_C(t)$  (see equation (6)).

3. When a new document  $y = \langle y_t \rangle_{t \in \mathcal{T}}$  is evaluated, the weighting formulas  $w_C(t)$  are easily updated.

4. Finally, the vector space similarity function  $SIM$  is 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 Evaluation Framework and Results

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* [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 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, on average, about 550 words in length. The size of the collection is about 260Mbyte. This is quite a

Recall	0.1	0.3	0.5	0.7	0.9
Precision	0.54	0.31	0.20	0.10	0.3

Table 1: Performance of ProFile for the base run.

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. The topics are complex and long and they can be regarded as almost complete description of the information need of a user. We regarded these topics as examples of relevant documents.

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. *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. Experimentally these measures have proved to be related in such a way that high precision brings low recall and viceversa. In order to give a measure of the learning performance of the filtering algorithm, R and P have been evaluated with different dimension of the set of training examples.

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 relevant and 32 non relevant documents, randomly chosen. The filtering runs shown in Tables 1, 2, 3 and 4 are thus incremental. We did not exploit the *idf* weighting function which would have required the processing of the whole collection in advance. Moreover, we only used a stop list without the stemming. We therefore used a minimum amount of information about the text collection at each run. This is the normal situation in which many filters work, e.g. filtering systems for the net news. We made the hypothesis that the system cannot process in advance the incoming data.

We divided the possible users into three categories: user A demands a high precision performance from the system and is happy with low recall performance (a recall value 0.3, that is 30% of the total number of relevant documents in the collection), the user B requires average levels of recall and precision (a recall of 0.5), and the user C who wants to retrieve most of the relevant information stored in the collection (a recall value of at least 0.7).

Table 2 and Table 3 show that the learning must be restricted to only high frequent terms in the training data. They 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 positive documents compared with a balanced training set; negative examples are neither harmful nor useless when combined with positive information (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

User	8R	16R	32R-AT	32R-TT	32R-HF
A	+ 6.3	+9.9	+10.6	+13.8	+15.5
B	+ 5.4	+9.6	+3.2	+13.5	+ 14.3
C	+17.7	+18.3	+22.5	+26.9	+ 27.3

Table 2: Precision increment in percentage 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.

User	4R-4N	8R-8N	16R-16N AT	16R-16N TT	16P-16N HF
A	+6.7	+8.6	+11.0	+8.9	+13
B	+7.5	+8.8	+12.5	+13.2	+ 16
C	+18.9	+19.3	+24.6	+24	+ 23

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

documents).

Table 4 shows that a training set made up of only negative examples do not contribute significantly in the tuning phase. The initial improvement with respect the base run does not growth by increasing the size of the training data. Notice that the performance of the run which does not add extra term to the base run profile (denoted by 32N-TT), is equal to the performance of the run (32N-AT) in which also the terms occurring in the training are weighted. This result shows that negative counterexamples contribute to eliminate the noise brought about by non-discriminant terms of the topic, but new negative terms do not help to discriminate the relevant documents.

Even though the topics were long and complex, the results show that few training documents improve substantially the performance of the system, hence a short tuning phase is indeed necessary especially when the document sources are different and not known in advance. Nevertheless, results shown that a relatively small subset of the relevance judgments works quite well with respect to the full set.

## 6 Conclusions and Future Work

We presented a learning algorithm to perform effective information dissemination and filtering. Our future work will be devoted to further experimentation in order to

User	8N	16N	32N-AT	32N-TT	32N-HF
A	+ 6.1	+5.9	+6.2	+6.1	+6.1
B	+8.8	+9.1	+9.3	+9.5	+ 9.5
C	+20.9	+20.9	+21.1	+21.2	+21.2

Table 4: Precision increment in percentage w.r.t. the base run using only non relevant documents

determine the best possible learning strategy among the many that can be performed using the proposed model.

## References

- [Amati and van Rijsbergen, 1995] 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.
- [Amati *et al.*, 1995] G. Amati, D. D’aloi, and V. Giannini. A framework for dealing with email and news messages. In *Proceedings of AICA 95*, pages 27–29, Cagliari, Italy, September 1995.
- [Bar-Hillel and Carnap, 1953] Y. Bar-Hillel and R. Carnap. Semantic information. *British Journal of the Philosophy of Science*, 4:147–157, 1953.
- [Belkin and Croft, 1992] 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.
- [Church and Hanks, 1989] K.W. Church and P. Hanks. Word association norms, mutual information and lexicography. In *Proceedings of ACL 27*, pages 76–83, Vancouver, Canada, 1989.
- [Ghosh, 1991] G. Ghosh. *A brief history of sequential analysis*. Marcel Dekker, New York, USA, 1991.
- [Harman, 1996] D. Harman. Overview of the fifth text retrieval conference (TREC-5). In *Proceeding of the TREC Conference*, Gaithersburg, MD, USA, November 1996.
- [Hintikka, 1970] J. Hintikka. On semantic information. In *Information and inference*. Synthese Library, Reidel, Dordrecht, The Netherlands, 1970.
- [Jeffrey, 1965] R.C. Jeffrey. *The logic of decision*. McGraw-Hill, New York, USA, 1965.
- [Lang, 1995] K. Lang. NewsWeeder: learning to filter netnews. In *Proceedings of ML 95*, pages 331–339, 1995.
- [Lewis and Gale, 1994] 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.
- [Robertson and Sparck Jones, 1976] 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.
- [Salton and McGill, 1983] G. Salton and M.J. McGill. *Introduction to modern Information Retrieval*. McGraw-Hill, New York, 1983.
- [van Rijsbergen, 1979] C.J. van Rijsbergen. *Information Retrieval*. Butterworths, London, second edition, 1979.