Rating the impact of logical representations on retrieval performance

David E. Losada
AILab. Department of Computer Science
University of A Corunna, Spain
losada@dc.fi.udc.es

Alvaro Barreiro
AILab. Department of Computer Science
University of A Corunna, Spain
barreiro@dc.fi.udc.es

Abstract

Logic provides a rich and uniform framework in which Information Retrieval can be modeled. The ability of logical approaches to give rise to more general Information Retrieval models is promising. However no much experimentation has been carried out about the real impact of logical representations on retrieval performance. This work is an attempt to fill this gap. Our work is based on a recent logical model of Information Retrieval in which some classical representations can be modeled and matched. We did experiments on both the basic retrieval task and the task of retrieval with feedback. The results obtained using expressive representations are encouraging for applying the model in real IR systems.

1. Introduction

Although logical approaches to Information Retrieval (IR) are being recognised ones, the practical side needs to be improved. Furthermore, recent compilations [1, 9] about the area of logical models of IR conclude that there is not a paradigm for applying logic to IR. Nevertheless, logic has been used in many areas, leading to expressive models in which reasoning is efficiently run. Nebel described recently [14] a number of domains in which logic-based Knowledge Representation (KR) has been successfully applied. For instance, Description Logics are now used by Lucent in a configuration system and Allen's interval calculus is used to describe document's layouts in order to support automatic mail sorting. The field of IR should also benefit from the ability of KR techniques to deal with knowledge. We strongly believe that logic is a fundamental tool to model IR in a better way. In fact, most of the classical IR models adopt strong assumptions because they do not have adequate tools to deal with knowledge. Documents and queries are knowledge containers and they should be treated as such.

Our research methodology is as follows. First, we have modeled some classical problems within a logical model. This pretends to show that logical approaches can deal with classical IR problems. Second, we stress the advantages of the logical approach. In this respect, we emphasize the ability of the model to manage expressive representations (not handled by classical models). The model we use has been recently implemented in an efficient way. From this implementation we have designed experiments to evaluate the model. Indeed, to implement logical models in an efficient way and to apply them into general domains are challeging issues when merging KR and IR [2]. These experiments involved distinct levels of expressiveness for documents and queries. We experimented with both the basic retrieval task and the retrieval task with feedback.

We chose Propositional Logic as the underlying formalism because its limited expressiveness allows to design efficient methods to compute similarity between documents and queries. Moreover, it is easy to articulate automatic methods for obtaining Propositional Logic representations for documents and queries in standard IR test collections. This facilitates the evaluation of the model against standard IR benchmarks.

The rest of the paper is organized as follows. Section 2 presents an overview of the logical model that stands on the basis of our proposal, the PLBR model (Propositional Logic and Belief Revision model). Section 3 presents the evaluation of the model against some IR test collections. Next, we extend the model to deal with relevance feedback. The evaluation results of the model with feedback are presented in section 5. The paper ends with some conclusions.

2. The PLBR Model

Along this work we will use the model proposed in [11]. This section depicts the basic foundations of this model. The review is intentionally brief and we refer to [11] for a detailed description of the model.

2.1. Representation and matching

Documents and queries are represented as Propositional Logic formulas. The expressiveness of propositional formulas allows us to manage representations which are the logical counterpart of binary-weighted vectors, e.g. $d = information \wedge retrieval \wedge \neg database$. Nevertheless propositional formulas are intrinsically more expressive than vectors. For instance, a document can be represented as $d = (information \wedge retrieval) \vee (data \wedge retrieval)$.

Given a document and a query represented by the propositional formulas d and q respectively, we need a procedure to measure the relevance of the document d to the information need expressed by q. The relevance test $d \models q$, where $\models$ stands for the logical entailment, which is fulfilled if all the models of d are also models of q, is too rigid because it is a binary criterion. Losada and Barreiro [11] proposed a method to get a non-binary measure of the entailment $d \models q$ as follows. The entailment $d \models q$ simply tests whether or not $Mod(d) \subseteq Mod(q)$. On the other hand, if we have a method to measure the distance from each model of d to the set of models of q, we would be able to define a non-binary measure of relevance. In the field of Belief Revision (BR) many efforts have been focused on the definition of measures between logical interpretations. The basic BR problem can be defined as follows. Let T be a logical theory and A a new formula to be included in the theory. BR methods define a way to include the new information in the theory. If there is no contradiction between T and A, the solution to the problem is trivial because the new theory, $T \circ A$ ($\circ$ stands for a revision operator), is just $T \wedge A$. However, if contradiction arises some old knowledge has to be deleted in order to get a consistent new theory. Model-based approaches to BR work on the logical interpretations of T and A. Basically, a measure of closeness to the set of models of the theory T is defined and the models of A which are the closest to the models of T are chosen to be the models of the new theory.

In [11] there was found an interesting connection between Dalal's BR operator [3], $\circ_D$, and IR matching functions. Let us regard a query q as a logical theory and a document d as a new information. In the revision process $q \circ_D d$ a measure from a given interpretation to the set of models of the query is defined. Thus, in order to get a non-binary measure of the entailment $d \models q$ we can compute the distance from each model of the document to the set of models of the query and, finally, calculate the average over document's models. This average over document's models was translated into a similarity measure, $BRsim$, in the interval $[0, 1]$.

The important result is that the model subsumes the vector-space model (with binary weights) with the inner product query-document matching function. This is because: a) binary-weighted vectors can be straightforwardly translated into propositional logic formulas (as conjunctions of literals) and b) the measure of matching between documents and queries, $BRsim$, gives the same result than the inner product query-document matching function between the corresponding vectors (see proof in [11]). However, as argued before, the logical formalism is inherently more expressive.

2.2. Implementation

The PLBR model defines the similarity between a document and a query using distances from models of the document to the set of models of the query. A fundamental problem arises if we implement directly this computation. The number of models of a propositional formula grows exponentially with the size of the propositional alphabet and, thus, the computation of the measure of relevance can be endless. Consequently, computations between models should be avoided. In [12] efficient procedures to compute similarity were proposed. A restriction in the syntactical form of the logical formulas involved allows to define polynomial-time algorithms to compute similarity. Specifically, the propositional formulas representing documents and queries have to be in disjunctive normal form (DNF). A DNF formula has the form: $c_1 \vee c_2 \vee \ldots$ where each c_j is a conjunction of literals (also called *clause*): $l_1 \wedge l_2 \wedge \ldots$. A literal is a propositional letter or its negation. Given a document d and a query q, if the formulas d and q are conjunctions of literals (i.e. DNF formulas without disjunctions), the value of $BRsim(d, q)$ is obtained in linear time. On the other hand, if there is some disjunction in either d or q (i.e. either d or q have more than one clause), we can get an approximation to the value of $BRsim(d, q)$ [12]. This approximation is obtained in polynomial time through computations between clauses. Specifically, the algorithm that computes similarity has a complexity of $\mathcal{O}\ (|\mu||\psi|\psi_{max})$, where $|\mu|$ ($|\psi|$) is the number of document (query) clauses and ψ_{max} is the size of the largest query clause.

3. Evaluation

We have used four small test collections for testing the PLBR model, namely CACM, Cranfield, CISI and LISA. We developed a prototype system that obtains automatically DNF formulas for representing documents and queries, matches documents and queries using the algorithms designed in [12] and computes the final retrieval performance results. Our indexing procedures parse the files containing documents and queries and extract sets of terms from the subfields of the document/query (the subfields considered are title .T, abstract .W, authors .A and keywords .K). Then, stopwords are eliminated and stemming is run. The final step of the indexing process implies the actual construction of a DNF formula. In this respect, we followed two main strategies:

1. First, for each document/query all the terms from all its subfields are collected in a conjunction, i.e. a DNF formula with an only clause is built. This strategy is similar to the classical approaches that build a vector in which all the terms from all the subfields are mixed up. In order to have a reference to classical approaches, we used SMART to produce retrieval performance results for the vector-space model with binary weights (along this work we will use the name of BVSP to refer to these experiments). SMART uses the inner product matching function and we indexed all document's subfields. Our prototype system used SMART's list of common words and SMART-enhanced version of the Lovins stemmer. We do not expect to get significant differences between BVSP and the PLBR model using strategy 1 because the similarity measure $BRsim$ subsumes the inner product query-document matching function. The equivalence is obtained when the logical formulas are built under a close world assumption (CWA). For instance, in fig. 1, a CWA policy would assume that the formula representing the document includes the negation of all the terms not appearing in the document. However, in all our tests we take benefit from the ability of the logic to deal with uncertainty and, then, we write formulas like the ones in the figure, where it is assumed that we do not know whether the document actually is (or is not) about the missing terms. The difference between the use of *partial vectors* (conjunctions without CWA in the PLBR model) and the use of total vectors (BVSP) stands on the treatment of the query terms that do not appear within the document. A simple analysis of $BRsim$ [11] shows that, on average, both approaches should produce similar performance results.

2. The second strategy is to separate terms from distinct subfields into distinct conjunctions. Intuitively, different subfields represent aspects of the document/query which are semantically different and, thus, it makes sense to isolate each part in a clause. Since DNF formulas provide us with a method to articulate several views of a document, we use each view to express a subfield. Figure 1 presents an example of the indexing process. At the top of the figure, a document from the CACM collection is shown (only the significant subfields are presented). At the bottom of the figure we present two possible logical representations for the document, depending on whether we index with one (strategy 1) or several clauses (strategy 2).

   It is important to note that we use DNF formulas for the reasons explained in section 2.2. DNF formulas are efficiently matched and we can extract DNF formulas from IR test collections. Hence, we can conduct experimentation in several generic environments. Our aim is not to define a model able to handle complex information from different points of view. This would require the use of more expressive formalisms, such as sublanguages of First Order Logic. In this respect, Description Logics and Conceptual Graphs have been applied to model IR objects in [13] and [15] respectively. But, although the expressiveness of Propositional Logic is limited, several views of a document can be handled using DNF formulas.

The main purpose of the experiments presented in this section is to determine whether or not strategy 2 outperforms strategy 1, i.e. whether or not the use of more expressive representations produces improvements in retrieval performance. Along this work we use the name of AND-tests for the experiments that applied strategy 1 for indexing documents and queries. The name of AND/OR-tests refers to the experiments that applied strategy 2 for indexing documents and queries, i.e. both conjunctions and disjunctions are used.

```
.T
Accelerating Convergence of Iterative Processes
.W
A technique is discussed which, when applied to an iterative procedure for the
solution of an equation, accelerates the rate of convergence if the iteration
converges and induces convergence if the iteration diverges.  An illustrative
example is given.
.A
Wegstein, J. H.
```

Strategy 1: DNF with an only clause

$accel \wedge converg \wedge iterat \wedge process \wedge techn \wedge discuss \wedge apply \wedge procedur \wedge solut \wedge equat \wedge rate \wedge induc \wedge diverg \wedge illustr \wedge wegstein$

Strategy 2: DNF with several clauses

$(accel \wedge converg \wedge iterat \wedge process) \vee (techn \wedge discuss \wedge apply \wedge iterat \wedge procedur \wedge solut \wedge equat \wedge accel \wedge rate \wedge converg \wedge induc \wedge diverg \wedge illustr) \vee wegstein$

Figure 1. Representing a CACM document

Figure 2 summarizes the results of our experiments for the four collections. In each column we show the set of subfields that were considered. We tried out different combinations of subfields and we present the results for the best AND-test and AND/OR-test. BSVP's performance is roughly the same than the performance of the PLBR model when only conjunctive representations are used (AND-tests).

Note that Cranfield's best AND-test indexes only the title subfield. Cranfield's abstracts are large (on average 86.1 terms after stopwords and stemming). On the other hand, abstracts in CACM, CISI and LISA have an average length of 48.3, 58.6 and 43.1 respectively. Hence, Cranfield's abstracts likely contain many terms which are not important to determine the actual contents of the documents. If we merge abstracts and titles in a single clause, important and not important terms are mixed and the resulting representation is likely worse. In fact, weights in the model are binary and,

	CACM			Cranfield		
R \ P	**PLBRM** AND-test TWKA	**BVSP** TWKA	**PLBRM** AND/OR-test TWKA	**PLBRM** AND-test T	**BVSP** T	**PLBRM** AND/OR-test TWA
0.00	0.5570	0.5444	0.6004	0.6946	0.6793	0.7430
0.10	0.4468	0.4594	0.4988	0.6542	0.6450	0.6968
0.20	0.3295	0.3495	0.3935	0.5340	0.5175	0.5752
0.30	0.2422	0.2733	0.3227	0.4118	0.3889	0.4860
0.40	0.1916	0.2081	0.2642	0.3291	0.3152	0.4165
0.50	0.1598	0.1803	0.2374	0.2866	0.2758	0.3562
0.60	0.1299	0.1525	0.1739	0.2131	0.1950	0.2766
0.70	0.0795	0.0830	0.1085	0.1372	0.1297	0.1777
0.80	0.0679	0.0610	0.0925	0.1025	0.0947	0.1358
0.90	0.0419	0.0415	0.0513	0.0672	0.0689	0.0949
1.00	0.0337	0.0302	0.0393	0.0654	0.0622	0.0897
Avg. prec.	0.2072	0.2167	0.2530	0.3178	0.3066	0.3680
% change		+4.6%	+22.1%		-3.5 %	+15.8%

	CISI			LISA		
R \ P	**PLBRM** AND-test TWKA	**BVSP** TWKA	**PLBRM** AND/OR-test TWKA	**PLBRM** AND-test TW	**BVSP** TW	**PLBRM** AND/OR-test TW
0.00	0.4826	0.4528	0.5308	0.2095	0.1801	0.2332
0.10	0.2461	0.2225	0.2865	0.1760	0.1137	0.1816
0.20	0.1757	0.1648	0.1997	0.1295	0.0816	0.1332
0.30	0.1451	0.1258	0.1607	0.0859	0.0524	0.0888
0.40	0.1249	0.1090	0.1310	0.0256	0.0142	0.0350
0.50	0.1076	0.0911	0.1080	0.0121	0.0095	0.0158
0.60	0.0936	0.0789	0.0924	0.0062	0.0072	0.0069
0.70	0.0803	0.0677	0.0766	0.0047	0.0045	0.0054
0.80	0.0711	0.0593	0.0671	0.0041	0.0040	0.0044
0.90	0.0597	0.0474	0.0571	0.0035	0.0032	0.0038
1.00	0.0485	0.0366	0.0476	0.0031	0.0027	0.0034
Avg. prec.	0.1487	0.1324	0.1598	0.0600	0.0430	0.0647
% change		-11.0%	+7.5%		-28.3%	+7.8%

Figure 2. Evaluation results for the basic retrieval task

thus, term frequency information cannot be used to measure the relative importance of distinct keywords. On the other hand, Cranfield's titles are not particularly short (on average 7.88 terms after stopwords and stemming, whereas titles in CACM, CISI and LISA have an average length of 5.27, 5.21 and 5.87 respectively) and then they are on their own good representatives for the documents.

In the CACM collection, the AND/OR-test outperforms clearly its corresponding AND-test. In the Cranfield collection similar results are obtained. The AND/OR-test improves significantly the best AND-test, which indexed only the title subfield. This shows that the use of a proper representation allows to include efficiently (in terms of retrieval performance) information from all the subfields. On the contrary, the AND-tests showed that a binary-weighted vector is not appropriate for integrating the information from all the subfields because the AND-test that indexes all the subfields was inferior to the AND-test that only indexes the title subfield.

The conclusions of these experiments are straightforward. It seems clear that the use of more expressive formulas for representing documents is good for an IR system in terms of retrieval performance. In all the collections, the experiments that stored more expressive formulas having conjunctions and disjunctions have led to significant improvements in the retrieval performance of the system. We think that the model might become a paradigm for IR models with binary information. The good retrieval performance results of the PLBR model are very promising for applying it within realistic systems.

4. Relevance feedback

The basic idea underlying relevance feedback is that relevant documents resemble each other and, if we move the query towards *some* relevant documents the performance of the system will be improved. In order to get some relevant documents, interaction with the user is needed. The basic relevance cycle works as follows. A first retrieval is done with the user's original query and the user is presented with a list of the top-ranked documents. The user marks those documents which are relevant and the system uses this information to build a new query. The new query is expected to be a better representation of the user's information need.

4.1. Document oriented feedback

Let us consider the feedback process within the PLBR model. An initial logical query q retrieves a set of documents. Let $D_r = \{d_{r1}, \ldots, d_{ri}\}$ be the set of documents in the top N that are marked by the user as relevant and $D_n = \{d_{nr1}, \ldots, d_{nrj}\}$ be the set of documents in the top N that are not relevant. The new query is expected to retrieve as many relevant documents as possible and, at the same time, move non-relevant documents down in the ranking. If we incorporate the formula $PF = (d_{r1} \vee \ldots \vee d_{ri})$ to the query, we are including models of the relevant documents in the models of the new query and, hence, relevant documents will have a good chance of being retrieved. On the other hand, we should avoid that models of the non-relevant documents belong to the set of models of the new query. This can be achieved through a formula like $\neg d_{nr1} \wedge \neg d_{nr2} \wedge \ldots \wedge \neg d_{nrj}$. However the previous formula can be unsatifiable. If we apply a revision process involving the formulas $\neg d_{nri}$ we can ensure that, if the conjunction of them is satisfiable, the result of the revision is equivalent to the conjunction. Specifically, given two formulas a and b and a revision operator $\circ$, if $a \wedge b$ is satisfiable then the revision $a \circ b$ is equivalent to $a \wedge b$. Furthermore, if the conjunction is not satisfiable then the revised formula $a \circ b$ is satisfiable as long as b is satisfiable. The revision process drops as few beliefs from a as possible and assures the incorporation of b ($(a \circ b) \models b$). As a result, we propose the formula $NF = \neg d_{nrj} \circ \neg d_{nrj-1} \circ \ldots \circ \neg d_{nr1}$ to reject non-relevant documents. Left associativity for the $\circ$ operator is assumed and, hence, the formula favors the unrelevant documents that are in a higher position in the rank (d_{nr1} is supposed to be the highest ranked non-relevant document and d_{nrj} the lowest ranked non-relevant document). In fact, some classical approaches, such as the Ide-Dec-Hi approach [8], apply negative feedback using information coming only from the highest ranked non-relevant document. To incor-

	CACM				Cranfield			
R \ P	BR	DO	TOP	TOPN	BR	DO	TOP	TOPN
0.00	0.320	0.399	0.376	0.422	0.319	0.418	0.408	0.463
0.10	0.254	0.285	0.311	0.351	0.298	0.382	0.383	0.438
0.20	0.190	0.191	0.212	0.237	0.236	0.324	0.343	0.363
0.30	0.151	0.141	0.173	0.195	0.182	0.257	0.254	0.292
0.40	0.103	0.102	0.109	0.135	0.144	0.205	0.203	0.227
0.50	0.084	0.081	0.097	0.114	0.128	0.179	0.182	0.215
0.60	0.070	0.071	0.072	0.091	0.084	0.113	0.130	0.145
0.70	0.051	0.052	0.053	0.058	0.060	0.088	0.092	0.102
0.80	0.040	0.040	0.041	0.045	0.050	0.073	0.080	0.083
0.90	0.022	0.021	0.019	0.023	0.042	0.063	0.072	0.071
1.00	0.016	0.016	0.015	0.017	0.041	0.061	0.071	0.070
Avg.prec.	0.118	0.127	0.134	0.154	0.144	0.197	0.201	0.224
%chg		+7.7%	+13.8%	+20.6%		+36.7%	+40.1%	+56.1%

	CISI				LISA			
R \ P	BR	DO	TOP	TOPN	BR	DO	TOP	TOPN
0.00	0.412	0.465	0.413	0.523	0.127	0.193	0.241	0.220
0.10	0.201	0.213	0.221	0.242	0.060	0.111	0.120	0.179
0.20	0.161	0.165	0.168	0.183	0.047	0.053	0.045	0.075
0.30	0.136	0.139	0.138	0.147	0.020	0.020	0.018	0.014
0.40	0.115	0.117	0.117	0.119	0.012	0.011	0.013	0.008
0.50	0.102	0.102	0.103	0.102	0.009	0.008	0.010	0.006
0.60	0.086	0.087	0.088	0.085	0.006	0.006	0.007	0.005
0.70	0.073	0.074	0.075	0.069	0.005	0.005	0.005	0.004
0.80	0.063	0.063	0.065	0.057	0.004	0.004	0.004	0.003
0.90	0.050	0.050	0.050	0.043	0.003	0.003	0.003	0.003
1.00	0.038	0.038	0.039	0.035	0.003	0.003	0.003	0.003
Avg.prec.	0.131	0.138	0.134	0.146	0.027	0.038	0.043	0.047
%chg		+5.2%	+2.6%	+11.6%		+41.4%	+59.3%	+76.9%

Figure 3. Evaluation results for the retrieval task with feedback

porate the feedback information into the new query we have to take into account that contradictions can arise between the original query q and the feedback formulas (either PF or NF) and between the feedback formulas themselves. In this respect, we propose to apply the following BR process in order to obtain a new query:

$$NF = \neg d_{nrj} \circ \neg d_{nrj-1} \circ \ldots \circ \neg d_{nr1}$$
$$PF = (d_{r1} \vee \ldots \vee d_{ri})$$
$$q_m = q \circ (NF \circ PF)$$

The importance of the factors involved determined the order of the revision processes. The formula $NF \circ PF$ makes that positive feedback prevails over negative feedback. This is the usual policy in classical approaches. The final revision process favors feedback information with respect to the original query.

The previous revision processes can be efficiently computed as long as the involved formulas are in DNF. As argued in section 3, documents and queries can be indexed in DNF. The formula PF is directly in DNF because it is a disjunction of DNF formulas. However the formula NF involves Conjunctive Normal Form (CNF) formulas [1] (each $\neg d_{nri}$ is a negation of a DNF formula, which is a CNF formula). No efficient procedures have been designed to compute this revision. As a result, the evaluation of this proposal should consider only positive feedback, i.e. $q_m = q \circ PF$. We designed an efficient algorithm that computes the previous revision process following the syntactic characterization proposed by del Val [4, 5]. Along this paper we use the name of document oriented (DO) approach to refer to the feedback process involving the formula $q_m = q \circ PF$.

4.2. Term Selection

Term selection approaches have shown that expanding the query with well-selected terms produces significant improvements in performance. Basically, all the terms from retrieved relevant documents are collected and ordered by a given sorting technique. Some techniques sort terms based on factors such as the total frequency of a term, the idf of a term or the number of retrieved relevant documents containing the term (postings) [6]. More ellaborated methods use ratios or probabilities of terms occurring in relevant vs ocurring in non-relevant documents [7]. Top ranked terms are supposedly the important ones within the set of relevant documents and, thus, they are a better representation for the set of relevant documents. Instead of expanding the query with all the terms from the relevant documents, the query is expanded with the selected terms.

The previous logical formulation of the feedback process takes whole document's representations to revise the original query. We can now refine the formulation to consider term selection. A formula collecting the selected terms is built. Nevertheless, we go one step further. Classical term selection techniques only operate on the retrieved relevant documents. We propose to select terms on the retrieved non-relevant documents as well. Classical models use to disregard the importance of negative feedback. Information from unrelevant documents is often considered in a restricted way. For instance, in the vector space model no new terms are actually added with negative weights. To motivate the importance of negative terms, let us consider a term t appearing in all the retrieved unrelevant documents and in no one retrieved relevant document. It seems reasonable to think that t is a good term to characterize the set of unrelevant documents. In the logical model we can incorporate $\neg t$ to the original query. On the contrary, the vector space model cannot use information from t because t would have a negative weight in the new query and negative weights are considered as 0 weights. We believe that negative terms can be used as a good precision-oriented mechanism.

We propose to revise the query using the selected terms from the relevant documents as positive terms and the selected terms from the unrelevant documents as negative terms. We build a revising formula which is the conjunction of all the terms, either positive or negative:

$$q_m = q \circ (t_{p1} \wedge \ldots \wedge t_{pn} \wedge \neg t_{n1} \wedge \ldots \wedge \neg t_{nm}),$$

where $t_{p1}, \ldots, t_{pn}$ are the selected terms from the retrieved relevant documents and $t_{n1}, \ldots, t_{nm}$ are the selected terms from the retrieved non-relevant documents. We will use

[1] A CNF formula has the form: $c_1 \wedge c_2 \wedge \ldots$ where each c_j is a disjunction of literals: $l_1 \vee l_2 \vee \ldots$.

the name of term oriented feedback to refer to this new approach.

5. Evaluating feedback

We evaluated both feedback strategies in CACM, Cranfield, CISI and LISA. A residual evaluation methodology was applied [16]. The top ten documents were used for relevance feedback. Not all the original queries can be considered for this evaluation because some of them retrieve all their relevant documents in the top ten and some of them retrieved no relevant documents in the top ten. Specifically, we used 47 CACM queries, 190 Cranfield queries, 60 CISI queries and 19 LISA queries.

In the evaluation of the term selection approach, we applied the postings method to select terms. The posting of a term is the number of relevant (non-relevant) documents in which it occurs. Our interest is to test the impact of the use of expressive representations when changing the query and not to compare several sorting methods.

Figure 3 presents the retrieval performance results for the feedback approaches. The first column of each table presents the base residual run (BR), i.e. the initial run without feedback and with the top ten documents removed. The second column presents the results for the document oriented approach (DO) and the last columns show the results for experiments using the term oriented approach. We tried out several expansions of the query varying the number of positive terms. In the third column (TOP) we show the results for the best run. Once the set of positive terms used for expansion is fixed, we expand the query with those positive terms and a set of negative terms. We varied the number of negative terms and, in the fourth column (TOPN), we show the performance results for the best run.

Significant improvements are obtained when feedback is introduced. In terms of average precision, the document oriented approach improved the base residual in all the tests. The run on the CISI collection presented the poorest improvement (5.2 % in terms of average precision). This might be due to the fact that CISI has many relevant documents per query and, then, it is difficult to approximate the query to that big set of documents. A big set of documents is likely more heterogeneous. If a query has many relevant documents, the similarity among them decreases and the impact of feedback is reduced. Besides, CISI has big queries and, then, there is not much room for improvement.

In general, term selection performed better than the document oriented approach. The selection with only positive terms outperformed the document oriented approach in three collections. Only in the CISI collection the document oriented approach was better than the selection with only positives. Again, the intrinsic difficulty of this collection might have produced this situation.

The selection of both positive and negative terms presented very good performance. It was the best approach in all collections. The improvements over the approach that only selects positive terms are quite significant. It is especially attractive the case of the CISI collection, in which the selection of positives did not outperform the document oriented approach but, on the contrary, the selection of both positives and negatives was significantly better than the document oriented approach. In this sort of environments (like CISI), where it is hard to approximate the query to the set of relevant documents, it becomes especially useful to have a method to move the query away from the unrelevant documents.

6. Conclusions

The PLBR model can constitute the basis of a working IR system. AND-tests have made evident that a restricted case of the PLBR model presents similar performance results than the vector-space model with binary weights. Indeed, this had already been predicted theoretically. Furthermore, more expressive document's representations have been applied leading to significant improvements in retrieval performance. We have also shown that Relevance Feedback can benefit from the use of negations. Our experiments have presented better performance ratios when negated terms are included in the new query.

The model presented here can be extended to deal with term similarity and inverse document frequency [10]. At this point, our intention was not to establish a strict comparison against other IR models. We think that this comparison should be done using the extension of the PLBR model that includes term similarity and inverse document frequency. Furthermore, so far, we have separated distinct subfields into distinct clauses of a DNF formula (AND/OR tests). This is a direct approach to test the effect of expressive representations on retrieval performance. However, more ellaborated separations might be applied. Negations should also be used when representing documents and queries because they are good precision-oriented tools. In summary, once we make extensive use of the expressive capabilities of the model, appropriate comparisons against other IR models will be scheduled.

7. Acknowledgments

This work was partially supported by projects refs. PB97-0228 and PGIDT99XI10201B.

References

[1] F. Crestani, M. Lalmas, and C. J. van Rijsbergen (editors). *Information Retrieval, Uncertainty and Logics: advanced*

models for the representation and retrieval of information. Kluwer Academic, Norwell, MA., 1998.

[2] W. B. Croft. Knowledge-based and statistical approaches to text retrieval. *IEEE Expert*, pages 8–12, April 1993.

[3] M. Dalal. Investigations into a theory of knowledge base revision: Preliminary report. In *Proc. of AAAI-88, the 7th National Conference on Artificial Intelligence*, pages 475–479, 1988.

[4] A. del Val. *Belief Revision and Update*. PhD thesis, Stanford University, 1993.

[5] A. del Val. Syntactic characterizations of belief change operators. In *Proc. of IJCAI'93, the Thirteenth International Joint Conference on Artificial Intelligence*, pages 540–545, Chambery, France, 1993.

[6] D. Harman. Towards interactive query expansion. In *Proc. SIGIR-88, the 11th ACM International Conference on Research and Development in Information Retrieval*, pages 321–331, Grenoble, France, June 1988.

[7] D. Harman. Relevance feedback revisited. In *Proc. SIGIR-92, the 15th ACM International Conference on Research and Development in Information Retrieval*, pages 1–10, Copenhagen, Denmark, June 1992.

[8] E. Ide. New experiments in relevance feedback. In *G. Salton (editor), The SMART retrieval system - Experiments in automatic document processing*, pages 337–354, Englewood Cliffs, NJ, 1971. Prentice Hall Inc.

[9] M. Lalmas and P. D. Bruza. The use of logic in information retrieval modeling. *Knowledge Engineering Review*, 13(3):263–295, 1998.

[10] D. E. Losada. *A logical model of information retrieval based on propositional logic and belief revision*. PhD thesis, University of A Corunna, 2001(to appear).

[11] D. E. Losada and A. Barreiro. Using a belief revision operator for document ranking in extended boolean models. In *Proc. of SIGIR-99, the 22th ACM Conference on Research and Development in Information Retrieval*, pages 66–73, Berkeley, California, August 1999.

[12] D. E. Losada and A. Barreiro. Efficient algorithms for ranking documents represented as DNF formulas. In *Proc. SIGIR-2000 Workshop on Mathematical and Formal Methods in Information Retrieval*, pages 16–24, Athens, Greece, July 2000.

[13] C. Meghini, F. Sebastiani, U. Straccia, and C. Thanos. A model of information retrieval based on a terminological logic. In *Proc. of SIGIR-93, the 16th ACM Conference on Research and Development in Information Retrieval*, pages 298–307, Pittsburgh, PA, June 1993.

[14] B. Nebel. Knowledge representation and reasoning - the theoretical side of AI. In *Proc. of ECAI'2000, the 14th European Conference on Artificial Intelligence*, page 763, Berlin, Germany, 2000.

[15] I. Ounis and J. Chevallet. Using conceptual graphs in a multifaceted logical model of information retrieval. In *Proc. of DEXA-96, the 7th International Conference on Database and Expert Systems Applications*, pages 812–823, Zurich, Switzerland, September 1996.

[16] G. Salton. Evaluation problems in interactive information retrieval. *Information Storage and Retrieval*, 6(1):29–44, 1970.