Propositional logic representations for documents and queries: a large-scale evaluation

David E. Losada[1] and Alvaro Barreiro[2]

[1] Intelligent Systems Group,
Department of Electronics and Computer Science,
University of Santiago de Compostela, SPAIN
dlosada@usc.es
[2] AIlab,
Department of Computer Science,
University of A Coruña, SPAIN
barreiro@dc.fi.udc.es

Abstract. Expressive power is a potential source of benefits for Information Retrieval. Indeed, a number of works have been traditionally devoting their efforts to defining models able to manage structured documents. Similarly, many researchers have looked at query formulation and proposed different methods to generate structured queries. Nevertheless few attempts have addressed the combination of both expressive documents and expressive queries and its effects on retrieval performance. This is mostly due to the lack of a coherent and expressive framework in which both documents and queries can be handled in an homogeneous and efficient way. In this work we aim at filling this gap. We test the impact of logical representations for documents and queries under a large-scale evaluation. The experiments show clearly that, under the same conditions, the use of logical representations for both documents and queries leads to significant improvements in retrieval performance. Moreover, the overall performance results make evident that logic-based approaches can be competitive in the field of Information Retrieval.

1 Introduction

Query structure has been extensively studied in the literature of Information Retrieval (IR). There is evidence that queries involving boolean operators are more effective than weaker query structures. Belkin and others combined manual boolean queries and found improvements in retrieval performance [1]. Hull investigated the impact of boolean structured queries in cross-language information retrieval and noticed that structured queries produce better performance [7]. Kekäläinen and Järvelin studied the effects of query structure in query expansion and found positive effects for expanded queries [8] .

The quest for methods for capturing the internal document structure has also been an active area of research in IR. For instance, a number of investigators proposed different approaches to divide documents into passages [6, 21, 24, 2] and there exists strong evidence that this additional information produces better retrieval performance results.

Although structured queries and structured documents have demonstrated their merits in the context of IR, their combination into the same retrieval model was not evaluated so far. More precisely, expressive document representations are usually matched against flat query expressions and, on the other hand, structured query formulations are often run against non-structured document representations. We claim that it is not sufficient to provide IR systems with powerful query languages if the representation of documents oversimplifies their information content. The reverse argument also holds. Both documents and queries should benefit from the full expressive power of the formalism involved. This was not addressed so far mainly because of the lack of an appropriate framework in which expressive documents and queries are homogeneously handled. This leads to unbalanced models, full of artificial ad-hoc elements, whose results can be hardly generalized.

One of the major advantages which stands on the foundations of logic-based approaches to IR [3] is precisely their ability to produce general and homogeneous retrieval models. In this work we adopt Propositional Logic as the underlying framework and show that better retrieval performance results are obtained when expressive representations are used for both documents and queries. There has been recurrent criticism against logical models of IR focused on complexity and evaluation issues. In this respect, we have taken great care of the actual applicability of the theoretical model. First, the efficiency of the logical approach followed here was recently assured [12, 13, 15]. Second, following the large-scale experimentation presented here, the model appears as a competitive retrieval model under realistic circumstances.

In most of the works on query structure, the formulation of queries was done either manually or assisted by external tools such as thesauri. In our work, we applied simplistic techniques to extract automatically expressive representations from both TREC topics and documents. The development of adequate and generic methods to build automatically expressive representations is indeed a great challenge for logical models of IR. Nevertheless, our simple automatic indexing method facilitates a large-scale evaluation on the impact of logical representations on retrieval performance.

The rest of this paper is organized as follows. In section 2 we briefly sketch the theoretical details of the underlying model. Section 3 reports the experiments conducted and section 4 discusses the evaluation results and other relevant issues. The paper ends with some conclusions.

2 Background

In this work we follow the logical approach for IR suggested by Losada and Barreiro [11, 15, 10]. This model is based on the combined use of Propositional Logic and Belief Revision. Along this paper, we will refer to this model as PLBR model. There are a number of reasons supporting this election. First, the PLBR model was efficiently implemented and polynomial-time algorithms were supplied to match documents and queries [13, 12, 15]. Second, the model was evaluated against four small test collections [16, 14] and the advantages of the use of an expressive formalism became apparent in those experiments. Nevertheless, those experiments could not test the combined effect of expressive documents and expressive queries because of the poor topic structure in those small collections.

Furthermore, the generality of the logical framework is appropriate for the objectives pursued here. Indeed, the PLBR model was successfully used in the past to model documents, queries, feedback information and retrieval situations in an homogeneous way [15]. More recently, the model was extended to include term similarity and inverse document frequency information [17].

2.1 The PLBR model

This section depicts the basic foundations of the PLBR model. The review is intentionally brief because further details can be found elsewhere [15, 10].

Documents and queries are represented as Propositional Logic formulas. Given a document and a query represented by the propositional formulas d and q respectively, it is well known that the application of the notion of logical consequence to decide relevance, i.e. $d \models q$, is too strict [23]. The entailment $d \models q$ simply tests whether or not each logical interpretation that makes d true makes also q true (i.e. each model of d is also a model of q). This is not in accordance with what we expect from an IR measure of relevance. Let us illustrate it through an example. Imagine two documents represented as $d_1 = a \wedge b \wedge \neg c \wedge d$ and $d_2 = \neg a \wedge \neg b \wedge \neg c \wedge d$ and a query represented as $q = a \wedge b \wedge c$. Both documents fail to fulfill the entailment, i.e. $d_1 \not\models q$ and $d_2 \not\models q$. This is because there exist models of d_1 that map the query into false[3]. Similarly, there are also models of d_2 that map the query into false[4]. As a consequence, the application of the logical entailment to decide relevance would assign the same status to both d_1 and d_2 with respect to the query q. This is not appropriate for IR purposes because d_1 is likely more relevant than d_2 (d_1 fulfills partially the query).

In [11] a method to get a non-binary measure of the entailment $d \models q$ was proposed. To define a non-binary measure of relevance the distance from each model of d to the set of models of q is measured. In the field of Belief Revision (BR) measures of distance between logical interpretations are formally defined. 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 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 removed in order to get a consistent new theory. Model-based approaches

[3] Note that any model m of d_1 maps the propositional letter c into false and, hence, m cannot be a model of q.

[4] Note that any model m of d_1 maps the propositional letters a, b, and c into false and, hence, m has to map q into false.

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. As a consequence, BR model-based approaches are suitable for measuring distances from documents to queries when both are represented as logical formulas. Next paragraph sketches the details of this formulation.

In [11] there was found an interesting connection between Dalal's BR operator [4], $\circ_D$, 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 document interpretation to the set of models of the query is defined. An important circumstance is that the semantics of this measure is appropriate for IR. Given a model of the document, the measure represents the number of propositional letters (i.e. index terms) that should be changed in that model in order to satisfy the query. For instance, let us assume a complete document d (i.e. a document having a single model) represented as neural $\wedge$ science $\wedge$ $\neg$network and a query q represented as neural $\wedge$ network. The distance from the document to the query would be equal to one because we would need to change the truth value of one propositional letter in the document (network) in order to satisfy the query. For that hypothetical changed document $d^{'}$, $d^{'} \models q$ would hold.

In the general case, a document representation may be partial and, hence, there might be several interpretations in which the document is satisfied (i.e. several document models). 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 is translated into a similarity measure, $BRsim$, in the interval $[0, 1]$.

Because $BRsim$ is model-based, a direct computation would require exponential time (the number of logical interpretations grows exponentially with the size of the alphabet). In [13, 12] efficient procedures to approximate the computation of $BRsim$ 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 *conjunctive clause*): $l_1 \wedge l_2 \wedge \ldots$. A literal is a propositional letter or its negation. As a result, a document d and a query q can be efficiently matched as long as d and q are in DNF. This restriction is acceptable because the expressiveness of generic propositional formulas and DNF formulas is the same. Indexing procedures have to represent documents as DNF formulas. From the user perspective, the use of DNF formulas does not introduce additional penalties. A translation from a natural language information need into a DNF query can be done automatically (this will be shown in section 3) or, alternatively, users can be asked to write generic propositional formulas and a translation into DNF is automatically done[5].

Let us imagine a document d represented by a DNF formula $dc_1 \vee dc_2 \vee \ldots$ and a query q represented by a DNF formula $qc_1 \vee qc_2 \vee \ldots$, where each dc_i (qc_i) is a conjunctive clause. The distance from the document to the query is measured as the average distance from document clauses to the set of query clauses. The distance from an individual document clause dc_j to the set of query clauses is measured as the minimum distance from dc_j to query clauses. Intuitively, different query clauses represent different requirements in the information need and the distance from dc_j to the query is measured as the distance to the requirement(s) that dc_j best fulfills. The clause-to-clause distance depends on (1) the number of literals appearing as positive literals within one clause and as negative literals within the other clause and (2) the number of literals in the query clause whose propositional letter is not mentioned by the document clause. The clause-to-clause distance helps to determine how good is the document clause for satisfying the query clause. In this respect, a contradicting literal, case (1), produces an increment of 1 to the distance whereas a query literal not mentioned by the document, case (2), increases 0.5 the value of the distance. This is because we do not know whether or not the document clause actually deals with that term[6] (recall that document representations are partial: information about presence/absence is not available for all the terms in the alphabet). The example depicted in fig. 1 helps to clarify the measure of distance applied. Note that the final value of distance is 0 because each document clause completely satisfies one or more query clauses, i.e. any document view satisfies one query requirement[7]. Observe that dc_1 does not include information about the term e and, hence, its distance from qc_1, which asks for e, gets an increment of 0.5.

An extension of the PLBR model was defined to include idf and term similarity information [17]. New efficient algorithms were designed and the experiments against small collections revealed that the model can be competitive with the vector-space model with the tf/idf weighting scheme.

3 Experiments

In our experiments, we used a subset of the TIPSTER/TREC collection consisting in about 173.000 documents. Specifically, we considered all Wall Street Journal (WSJ) documents (years 87-92) in TIPSPER/TREC volumes 1&2.

[5] Although a translation from a propositional formula into DNF can take in the worse case exponential time, queries have usually few terms and, then, the translation time is acceptable.

[6] This decision is theoretically supported by the fact that half of the models of the document clause map the term into true and half of the models of the document clause map the term into false or, alternatively, half of the models of the document clause *agree* with the query clause and half of the models of the document *disagree* with the query clause.

[7] Since query requirements are combined through logical disjunctions the satisfaction of one single requirement is enough to satisfy the query.

$\mathcal{P} = \{a, b, c, d, e\}$
$d = (a \wedge b \wedge d) \vee (a \wedge \neg b \wedge \neg d \wedge e)$, $q = (a \wedge e) \vee (a \wedge d)$
document $d = dc_1 \vee dc_2$, $dc_1 = (a \wedge b \wedge d)$, $dc_2 = (a \wedge \neg b \wedge \neg d \wedge e)$
query $q = qc_1 \vee qc_2$, $qc_1 = (a \wedge e)$, $qc_2 = (a \wedge d)$
Distance from dc_1 ***to*** q
Distance from dc_1 ***to*** qc_1
#contradicting literals = 0
#terms in q clause not mentioned by the doc clause = 1 (e)
Distance(dc_1,qc_1)= 0 + 1/2 = 0.5
Distance from dc_1 ***to*** qc_2
#contradicting literals = 0
#terms in q clause not mentioned by the doc clause = 0
Distance(dc_1,qc_2)= 0 + 0/2 = 0
Distance from dc_1 to q = 0
Distance from dc_2 ***to*** q
Distance from dc_2 ***to*** qc_1
#contradicting literals = 0
#terms in q clause not mentioned by the doc clause = 0
Distance(dc_2,qc_1)= 0 + 0/2 = 0
Distance from dc_2 ***to*** qc_2
#contradicting literals = 1 (d)
#terms in q clause not mentioned by the doc clause = 0
Distance(dc_2,qc_2)= 1 + 0/2 = 1
Distance from dc_2 to q = 0
Distance from d to q = (0+0)/2 = 0

Fig. 1. Distance from a DNF document to a DNF query

In order to index this collection, we used GNU mifluz [18]. GNU mifluz provides a C++ library to build and query a full text inverted index. Mifluz was developed by Senga [22], which is a development group focused on IR software. The flexibility of mifluz routines allowed us to create an inverted file in which, for each term, we store both document and clause information. Recall that we store documents as DNF formulas and conventional inverted files were not designed to store clause information. Mifluz is very flexible and allows to define explicitly the structure of the inverted file. As a consequence, we could design and build an inverted file able to efficiently store documents as DNF formulas.

A total of 50 TREC topics were used in this experimentation. Topics #151 - #200 from TREC-3 adhoc retrieval task [5] were used to generate automatically DNF queries for representing user needs. We used a stoplist of 571 words and terms were stemmed using Porter's algorithm [19].

3.1 Evaluating the PLBR model

Two main strategies were applied to define logical queries. First, a baseline with flat query structure is built as follows. All query terms are extracted and, after stopword and stemming, the query terms are collected into a single clause, i.e. a DNF formula with a single conjunctive clause is built. A second class of tests are based on expressing queries as DNF formulas having several clauses. Each query clause is formed from a subfield of the TREC topic. Figure 2 shows an example of both strategies for topic No. 160.

It is important to observe that, although simplistic, this approach is able to build automatically structured queries for TREC topics. Most of the works aforementioned [1, 7] are based on structured queries built manually. Kekäläinen

<title> Topic: Vitamins - The Cure for or Cause of Human Ailments
<desc> Description:
Document will identify vitamins that have contributed to the cure for human diseases or ailments or documents will identify vitamins that have caused health problems in humans.
<narr> Narrative:
A relevant document will provide information indicating that vitamins may help to prevent or cure human ailments. Information indicating that vitamins may cause health problems in humans is also relevant. A document that makes a general reference to vitamins such as "good for your health" or "having nutritional value" is not relevant. Information about research being conducted without results would not be relevant. References to derivatives of vitamins are to be treated as the vitamin.

Strategy 1: DNF with a single clause
vitamin ∧ cure ∧ caus ∧ human ∧ ailment ∧ document ∧ identifi ∧ contribut ∧ diseas ∧ health ∧ problem ∧ relevant ∧ provid ∧ inform ∧ indic ∧ prevent ∧ make ∧ gener ∧ refer ∧ good ∧ nutrit ∧ research ∧ conduct ∧ result ∧ deriv ∧ treat

Strategy 2: DNF with several clauses
(vitamin ∧ cure ∧ caus ∧ human ∧ ailment) ∨ (document ∧ identifi ∧ vitamin ∧ contribut ∧ cure ∧ human ∧ diseas ∧ ailment ∧ caus ∧ health ∧ problem) ∨ (relevant ∧ document ∧ provid ∧ inform ∧ indic ∧ vitamin ∧ prevent ∧ cure ∧ human ∧ ailment ∧ caus ∧ health ∧ problem ∧ make ∧ gener ∧ refer ∧ good ∧ nutrit ∧ research ∧ conduct ∧ result ∧ deriv ∧ treat)

Fig. 2. Representing a TREC topic

and Järvelin work on automatic queries but query structure produces only better results after expansion [8]. The small-scale experiments of the PLBR model reported in [14, 16] do not provide a detailed study of the effect of query structure because of the poor variety of subfields in the topics.

The first aim of these experiments is to determine whether or not the separation of query information into several clauses is beneficial in terms of retrieval performance. Note that, intuitively, each subfield represents a different view of the information need and it seems sensible to think that a separate representation is adequate.

In order to isolate the effect of expressive queries from the effect of expressive documents, we first ran experiments on flat document representations with varying *degree of expressiveness* for queries. Specifically, we first considered documents as conjunctions of terms (i.e. DNF formulas having a single conjunctive clause) where all terms from different document subfields are represented into the same document clause, i.e. no structure information is handled for documents. In table 1 we present performance results for this first pool of experiments. Tests with and without idf information were run. The use of expressive query representations leads to spectacular improvements in retrieval performance. Observe that the test using structured queries with no idf is even better than the test using idf on flat queries. This supports the idea that IR needs flexible query languages able to express user information needs in a more adequate way. Recall that DNF formulas having several clauses involve the use of both logical disjunctions and logical conjunctions whereas DNF formulas with a single clause involve only the use of logical conjunctions. Negations were not used in this evaluation. From the evaluation results obtained, it appears that the variety of logical connectors to formulate queries is a good property of the query language.

	no idf		idf	
Recall	**1 clause in docs 1 clause in qs**	**1 clause in docs several clauses in qs**	**1 clause in docs 1 clause in qs**	**1 clause in docs several clauses in qs**
0.00	0.3235	0.4922	0.5260	0.5173
0.10	0.1535	0.2730	0.2792	0.3474
0.20	0.0896	0.2402	0.2010	0.3112
0.30	0.0485	0.1785	0.1407	0.2589
0.40	0.0304	0.1478	0.0978	0.2024
0.50	0.0169	0.1110	0.0692	0.1581
0.60	0.0087	0.0932	0.0454	0.1356
0.70	0.0020	0.0737	0.0269	0.1178
0.80	0.0006	0.0475	0.0162	0.0871
0.90	0.0001	0.0352	0.0055	0.0652
1.00	0.0001	0.0171	0.0042	0.0248
Avg.prec. (non-interpolated)	0.0451	0.1316	0.1055	0.1792
% change		+191.8%		+69.9%

Table 1. Effect of expressive queries on retrieval performance

In a second pool of experiments we considered documents as DNF formulas having several conjunctive clauses and queries as DNF formulas having a single conjunctive clause. As for queries, to get DNF representations for WSJ documents we used the subfield structure of the WSJ documents. In the experiments reported here, we considered the subfields HL, TEXT and LP which corresponds to headlines, main text and lead paragraphs, respectively. Terms from each subfield are collected into a conjunctive clause and the document representation is composed of the disjunction of all these clauses. We also considered an additional clause which is composed of all the terms from all the subfields. In this way, we have an additional view which represents the full document. This was inspired by some works on Passage Retrieval [21, 24] that use both local (document passages) and global (full document) information. Nevertheless, it has been traditionally difficult to decide which view is adequate for a particular retrieval. The logical formalism is flexible enough and can cope with alternative views of the documents and all of them are considered at retrieval time.

In table 2 performance results obtained from expressive document representations are presented. All these results were obtained using queries having a single conjunctive clause. The effect of expressive document representations is negative when no idf information is available and positive when idf information is considered. Unfortunately, following these results we cannot reach a clear conclusion about the effect of expressive document representations when flat query expressions are used. In table 3 we show the performance ratios obtained when both documents and queries are represented as DNF formulas having several clauses. We also show results for conjunctive representations for both documents and queries. The improvements found in retrieval performance from the use of generic DNF formulas for both documents and queries are huge. Clearly, expressive formulas appear as an important tool to improve retrieval performance. However, the effect of expressive document representations when flat queries are used is unclear. This experimentation provides no clear evidence about the adequacy of expressive document representations when the query language is poor. This might indicate that it is not sufficient to apply expressive document representations if the representation of queries oversimplifies their information content. This idea is supported by the fact that the best performance of the logical model is obtained when the full expressive power is applied to both doc-

	no idf		idf	
Recall	**1 clause in qs 1 clause in docs**	**1 clause in qs several clauses in docs**	**1 clause in qs 1 clause in docs**	**1 clause in qs several clauses in docs**
0.00	0.3235	0.3188	0.5260	0.4988
0.10	0.1535	0.1279	0.2792	0.2753
0.20	0.0896	0.0866	0.2010	0.2192
0.30	0.0485	0.0439	0.1407	0.1634
0.40	0.0304	0.0276	0.0978	0.1014
0.50	0.0169	0.0170	0.0692	0.0715
0.60	0.0087	0.0101	0.0454	0.0507
0.70	0.0020	0.0041	0.0269	0.0320
0.80	0.0006	0.0009	0.0162	0.0208
0.90	0.0001	0.0003	0.0055	0.0088
1.00	0.0001	0.0003	0.0042	0.0038
Avg.prec. (non-interpolated)	0.0451	0.0425	0.1055	0.1104
% change		-5.8%		+4.6%

Table 2. Effect of expressive documents on retrieval performance

uments and queries. In the discussion section we provide an additional analysis about the effects of the logical approach on retrieval performance.

	no idf		idf	
Recall	**1 clause in docs & qs**	**several clauses in docs & qs**	**1 clause in docs & qs**	**several clauses in docs & qs**
0.00	0.3235	0.6231	0.5260	0.6445
0.10	0.1535	0.4489	0.2792	0.5023
0.20	0.0896	0.3485	0.2010	0.4128
0.30	0.0485	0.2755	0.1407	0.3387
0.40	0.0304	0.2182	0.0978	0.2646
0.50	0.0169	0.1666	0.0692	0.2106
0.60	0.0087	0.1376	0.0454	0.1783
0.70	0.0020	0.0929	0.0269	0.1342
0.80	0.0006	0.0743	0.0162	0.1009
0.90	0.0001	0.0396	0.0055	0.0695
1.00	0.0001	0.0138	0.0042	0.0206
Avg.prec. (non-interpolated)	0.0451	0.1980	0.1055	0.2378
% change		+339.0%		+125.4%

Table 3. Effect of expressive documents and expressive queries on retrieval performance

3.2 Comparison with the Vector-Space model

In this section we compare the results obtained with the PLBR model against results obtained with the Vector-Space model. The latter results were obtained using the Lemur toolkit [9]. Lemur supports the construction of text retrieval systems using popular IR models such as Vector-Space and Okapi or newer ones such as Language Modeling approaches. It is designed to facilitate research in IR using large-scale databases. Lemur was developed by the Computer Science Department of the University of Massachusetts and the School of Computer Science at Carnegie Mellon University in the framework of the so-called Lemur Project. This does not pretend to be a strict comparison because the PLBR model can only deal with binary term frequency information and, on the other hand, the VSP model can not handle documents and queries divided into parts. However, it is interesting to see the absolute retrieval performance of the logical approach against the retrieval performance of a popular IR model.

The procedure to obtain VSP retrieval performance results was as follows. First, we ran Lemur routines to build a classical inverted file for the WSJ collec-

	no idf			idf		
Recall	**VSP bin tf**	**VSP raw tf**	**PLBR**	**VSP bin tf**	**VSP raw tf**	**PLBR**
0.00	0.3386	0.4500	0.6231	0.6235	0.6699	0.6445
0.10	0.1473	0.2383	0.4489	0.3520	0.3988	0.5023
0.20	0.0863	0.1726	0.3485	0.2858	0.3460	0.4128
0.30	0.0465	0.1379	0.2755	0.2096	0.2967	0.3387
0.40	0.0303	0.1086	0.2182	0.1567	0.2563	0.2646
0.50	0.0156	0.0699	0.1666	0.1092	0.2001	0.2106
0.60	0.0085	0.0443	0.1376	0.0858	0.1565	0.1783
0.70	0.0021	0.0261	0.0929	0.0584	0.1115	0.1342
0.80	0.0006	0.0148	0.0743	0.0363	0.0741	0.1009
0.90	0.0001	0.0042	0.0396	0.0165	0.0330	0.0695
1.00	0.0001	0.0015	0.0138	0.0071	0.0118	0.0206
Avg.prec. (non-interpolated)	0.0450	0.0946	0.1980	0.1532	0.2104	0.2378
% change		+110.2%	+340.0%		+37.3%	+55.2%

Table 4. PLBR model vs Vector-Space Model

tion[8]. As in the experiments with the PLBR model, we indexed the HL, TEXT and LP subfields (headlines, full text and lead paragraph, respectively) and terms were stemmed using Porter's algorithm [19]. The stoplist was the same used in the tests of the PLBR model. Note that evaluation is done at the document level. Although documents have several clauses, there are not relevance assessments for particular clauses (only whole documents have their relevance assessment).

Table 4 depicts the results obtained for the WSJ collection using several weighting schemes[9] and figure 3 shows the corresponding precision vs recall graph. For comparison, we also show the performance results obtained with the PLBR model when both documents and queries are represented as DNF formulas having several clauses.

These experiments allow us to extract a number of conclusions. First, when no idf information is available, the PLBR model is always superior to the VSP[10]. Even though the VSP uses raw tf, the PLBR model keeps being better (19.8% average non-interpolated precision vs 9.46% average non-interpolated precision). Recall that the PLBR model can only deal with binary term frequency information. Nevertheless, the positive effect obtained from expressive representations is superior to the negative effect related to the lack of a non-binary term frequency notion. When idf information is available, the same tendency holds. If the notion of term frequency is binary the PLBR model performs better than the VSP (55.2% better in average non-interpolated precision). The raw tf/idf VSP experiment is slightly inferior to the PLBR model. However, it is well known that important improvements can be obtained with the VSP if weighting schemes such as BM25 [20] are applied. This suggests that additional investigation is needed to determine whether or not the PLBR model can be competitive in terms of absolute ratios of retrieval performance. However, we still do not know the limits of the PLBR model because the full expressive power was not utilized.

[8] In this step, we introduced minor changes in Lemur source code to be able to select which document subfields were indexed.

[9] We also had to introduce minor additions in Lemur source code to handle some of the weighting schemes depicted in the table.

[10] Observe that neither the VSP model nor the PLBR model were tested using normalization. Indeed the incorporation of some kind of normalization (maybe clause-based) in the PLBR model is a future line of work.

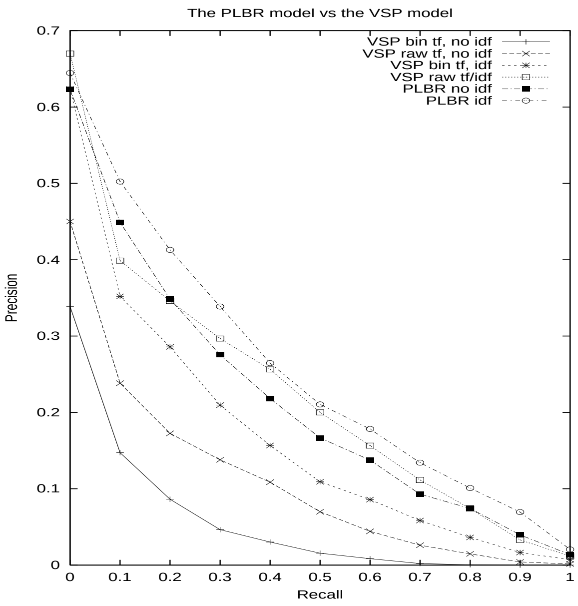

Fig. 3. The PLBR model vs the VSP model

Negations were not considered in these experiments. As it was mentioned before, the quest to design techniques that obtain automatically better logical representations of texts is a major challenge for logical approaches to the IR problem. Furthermore, the PLBR model used so far does not apply any normalization factor. In order to ascertain the real limits of a logical approach such as ours, it is very important to investigate on formal ways to encompass non-binary term frequency information and methods to apply clause-based normalization. Anyway, the results of this experimentation are clear: IR models can obtain large benefits if structure is handled for both documents and queries. Under the same conditions, the structured version was always significantly better.

4 Discussion

In the experiments reported in this paper logic appears as a tool to enhance retrieval precision. In this section we look deeply into the characteristics of the matching process trying to find explanations for that good behaviour. Specifically, we look at query expressiveness, whose benefits in retrieval performance are especially large.

Consider documents as DNF formulas with a single clause and queries as DNF formulas having several clauses. In this case, the PLBR model behaves clearly better than the PLBR model with flat representations. If queries have a single conjunctive clause then all the terms appearing in the TREC topic (even in different subfields) are mixed up into that clause. This clause is used to match the document clause. On the other hand, if we can represent queries with several conjunctive clauses then we will be able to separate distinct parts of the information need into distinct conjunctive clauses. As argued in section 2, the distance from the document clause to the query is measured as the distance to the closest query clause(s). Let us imagine a topic whose title is “Dog maulings” and a relevant document d_r which mentions both terms. If all the query terms are collected into a single conjunctive clause the position in the rank of d_r will depend on how many query terms appear in the document. Alghough d_r mentions “dog” and “maulings”, it might be the case that it receives a low retrieval score because most of the other query terms are not present in d_r (e.g. because the relevant document is short). Intuitively, if the query language forces us to store all the terms into the same flat structure then, the meaning of the information need is blurred. Think that, the longer the query is, the more chance to have generic terms which are not very important to decide relevance (and, hence, the more chance for long documents to match the query). If we represent the title into a single conjunctive clause and the rest of the topic is separated into distinct clauses, then the retrieval score of d_r will be maximum (because one query clause - the title query clause - fares 0 from the document), no matter how far the rest of the query clauses are. This means that the satisfaction of a single query clause is enough to assign a high rank to the document. Although a given document does not share many terms with a query, it can receive a good retrieval score because it fulfills completely one of the query views. As a consequence, the semantics of the distance that PLBR uses helps to move relevant documents towards higher positions in the rank.

One can reasonably argue that a similar behaviour might be obtained in the VSP model if we assign weights for query terms taking into account the subfield of the topic in which the terms are mentioned. This would allow to measure the relative importance of the query terms but, as structure is not handled, we could not recognize whether or not a part of the query is fully satisfied.

When documents are DNF formulas having several clauses the retrieval performance of the PLBR model gets further improvements. The separation of the document information into several parts helps to refine the matching process and, for each document clause, its closest query clause(s) is located. This means that an ellaborated matching is done that takes into account matches between portions of the document and portions of the query. The practical advantages in retrieval performance of this formulation are clear. On the other hand, if queries have a single clause, there is no evidence that the separation of documents into several parts is beneficial. More experimental work is needed to shed light on this issue. Anyway, the use of expressive representations for both documents and queries was always significantly better than any other approach and, thus, there is no doubt about the role of expressiveness for enhancing retrieval systems. On the contrary, that circumstance supports the idea that representational power should be fully provided to both documents and queries.

Observe that the logical approach followed in this work captures only a binary notion of term frequency (tf). The reader might wonder why the model does not include the tf factor. The idf factor and a measure of similarity between terms are *global* notions, i.e. they do not depend on a particular document but are characteristics of the whole collection (furthermore, the notion of term similarity is not collection-dependent because we can even get a measure of similarity between terms from a thesaurus, from other collections, etc.). These notions introduce additional information about the involved terms which is considered by the distance measured at retrieval time. However, our representational formalism keeps being the same: Propositional Logic. The tf factor, which is determined by the number of occurrences of a term within a document, is not a global notion but it is associated to a particular document. At matching time, we can use the idf factor and term similarity information for measuring the distance between two interpretations because they are global factors and, hence, we do not need to know which document/query is being handled. On the contrary, to apply the tf factor we would need to know which document/query corresponds to the interpretations being handled [15]. If we want to adhere to the theoretical formalism, this would not be possible because a given Propositional Logic interpretation can be a model of many documents and queries. Hence, the notion of interpretation would have to incorporate term frequencies giving rise to a totally different model. As a consequence, the PLBR model cannot consider term frequency information.

5 Conclusion

The most popular IR models have been traditionally driven by efficiency rather than expressiveness. This leads to IR systems which retrieve large amounts of documents very quickly but whose representational power is poor. As a result, generalization is hardly possible and the structure of documents and queries receives a marginal role. It is difficult to get increasingly better performance results based on such models. Research on weighting schemes, normalization, etc. has made a tremendous effort to enhance IR but they are limited by the characteristics of the underlying representational apparatus. We claim that IR systems should consider formalisms able to capture an enhanced notion of document and query. We are not sure about which the best framework is but we are pretty confident that the expressive power is a fundamental tool to improve retrieval performance.

The performance results obtained in this work support the intuitions reflected in the last paragraph. Huge benefits were found when documents and queries are represented as expressive formulas. Under the same conditions, the logical approach was always superior to the classical vector-space model. The combined use of split representations and a matching process driven by the closest query clause appear as adequate tools to model IR systems.

Previous experiments using the PLBR model against small collections [16, 14] anticipated its good behaviour but the full expressive power was not utilized. Following the evaluation reported here, we can say without doubt that the more expressive the model is, the better it does retrieval. This suggests that IR systems should allow to match expressive documents against expressive queries. Moreover, the size of the collection utilized here assures the good operation of the PLBR model under realistic circumstances.

Note also that significant improvements were obtained with coarse techniques for separating a document/query into several clauses. In the future we plan to apply more complex procedures to divide documents/queries into clauses. Moreover, in the experiments presented here we did not make use of negations. The incorporation of negated terms into queries in a relevance feedback loop was recently evaluated with very good performance results [14]. We believe that negations can play an important role as a precision-oriented mechanism.

Although the expressiveness of Propositional Logic is limited, further extensions of the PLBR model towards more expressive logics such as First Order Logic can be undertaken. As logical models are more general, newer models can inherit results obtained previously.

Acknowledgements

The work reported here was co-funded by "Ministerio de Ciencia y Tecnología" and FEDER funds under research project TIC2002-00947 (R&D program: "Tecnologías de la Información y las Comunicaciones"). The first author is supported in part by "Ministerio de Ciencia y Tecnología" and in part by FEDER funds through the "Ramón y Cajal" R&D program.

References

1. N.J. Belkin, C. Cool, W.B. Croft, and J.P. Callan. The effect of multiple query representations on information retrieval system performance. In *Proc. of SIGIR-93, the 16th ACM Conference on Research and Development in Information Retrieval*, pages 339–346, Pittsburgh, PA, June 1993.
2. J.P. Callan. Passage-level evidence in document retrieval. In *Proc. SIGIR-94, the 17th ACM Conference on Research and Development in Information Retrieval*, pages 302–310, Dublin, UK, July 1994.
3. 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.
4. M. Dalal. Investigations into a theory of knowledge base revision:preliminary report. In *Proc. AAAI-88, the 7th National Conference on Artificial Intelligence*, pages 475–479, Saint Paul, USA, 1988.
5. D. Harman. Overview of the third text retrieval conference. In *Proc. TREC-3, the 3rd text retrieval conference*, 1994.
6. M. Hearst and C. Plaunt. Subtopic structuring for full-length document access. In *Proc. SIGIR-93, the 16th ACM Conference on Research and Development in Information Retrieval*, pages 59–68, Pittsburgh, USA, June 1993.

7. D.A. Hull. Using structured queries for disambiguation in cross-language information retrieval. In *Proc. of AAAI spring symposium on cross-language text and speech retrieval*, Stanford, CA, March 1997.
8. J. Kekäläinen and K. Järvelin. The impact of query structure and query expansion on retrieval performance. In *Proc. of SIGIR-98, the 21st ACM Conference on Research and Development in Information Retrieval*, pages 130–137, Melbourne, Australia, August 1998.
9. The lemur toolkit. http://www.cs.cmu.edu/ lemur.
10. D. E. Losada. *A logical model of information retrieval based on propositional logic and belief revision*. PhD thesis, University of A Corunna, 2001.
11. D. E. Losada and A. Barreiro. Using a belief revision operator for document ranking in extended boolean models. In *Proc. SIGIR-99, the 22nd ACM Conference on Research and Development in Information Retrieval*, pages 66–73, Berkeley, USA, 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. D. E. Losada and A. Barreiro. Implementing document ranking within a logical framework. In *Proc. SPIRE-2000, the 7th Symposium on String Processing and Information Retrieval*, pages 188–198, A Coruña, Spain, September 2000.
14. D. E. Losada and A. Barreiro. An homogeneous framework to model relevance feedback. In *Proc. of SIGIR-2001, the 24th ACM Conference on Research and Development in Information Retrieval (poster session)*, pages 422–423, New Orleans, USA, September 2001.
15. D. E. Losada and A. Barreiro. A logical model for information retrieval based on propositional logic and belief revision. *The Computer Journal*, 44(5):410–424, 2001.
16. D. E. Losada and A. Barreiro. Rating the impact of logical representations on retrieval performance. In *Proc. DEXA-2001 Workshop on Logical and Uncertainty Models for Information Systems, LUMIS-2001*, pages 247–253, Munich, Germany, September 2001.
17. D. E. Losada and A. Barreiro. Embedding term similarity and inverse document frequency into a logical model of information retrieval. *Journal of the American Society for Information Science and Technology*, 2003 (to appear).
18. GNU mifluz. http://www.gnu.org/software/mifluz.
19. M.F. Porter. An algorithm for suffix stripping. In K.Sparck Jones and P.Willet, editors, *Readings in Information Retrieval*, pages 313–316. Morgan Kaufmann Publishers, 1997.
20. S.E. Robertson, S. Walker, S. Jones, M.M. HancockBeaulieu, and M. Gatford. Okapi at TREC-3. In D.Harman, editor, *Proc. TREC-3, the 3rd Text Retrieval Conference*, pages 109–127. NIST, 1995.
21. G. Salton, J. Allan, and C. Buckley. Approaches to passage retrieval in full text information systems. In *Proc. SIGIR-93, the 16th ACM Conference on Research and Development in Information Retrieval*, pages 49–58, Pittsburgh, USA, June 1993.
22. Senga: Information retrieval software. http://www.senga.org.
23. C.J. van Rijsbergen. A non-classical logic for information retrieval. *The Computer Journal*, 29:481–485, 1986.
24. R. Wilkinson. Effective retrieval of structured documents. In *Proc. SIGIR-94, the 17th ACM Conference on Research and Development in Information Retrieval*, pages 311–317, Dublin, UK, July 1994.