Implementing Document Ranking within a Logical Framework

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

This work deals with the implementation of a logical model of Information Retrieval. Specifically, we present algorithms for document ranking within the Belief Revision framework. Therefore, the logical model that stands on the basis of our proposal can be efficiently implemented within realistic systems. Besides the inherent advantages introduced by logic, the expressiveness is extended with respect to classical systems because documents are represented as unrestricted propositional formulas. As well as representing classical vectors, the model can deal with partial descriptions of documents. Scenarios that can benefit from these more expressive representations are discussed.

1 Introduction

The actual applicability of logical models of Information Retrieval (IR) to the construction of practical systems is controversial. In this work we implement document ranking within a logical framework, contributing this way to show that IR systems can be based on logical models. Moreover, such systems can benefit from the management of incomplete descriptions, which is an inherent characteristic of logic. Generality and formalization of well-known IR notions are additional advantages of the use of logic for IR.

Documents rarely fulfil queries in a complete way. Therefore, a realistic logical model for IR should not decide relevance using the classical entailment $d \models q$. Basically, classical entailment is too strong and cannot represent partial relevance [12]. Classical entailment represents the notion of logical consequence, i.e $\rho \models \phi$ holds iff ϕ is satisfied in all the interpretations satisfying ρ. Van Rijsbergen [19] showed that any logical approach should quantify relevance taking into account the minimal change that must be done in order to establish the truth of the entailment. Several approaches have followed this line of research and recent compendia can be found in [3, 13]. Among several alternatives, we focus here on the framework proposed in [16], where Belief Revision (BR) techniques were used for quantifying relevance between documents and queries. Unfortunately, a direct implementation of the model proposed in [16] would require exponential time to decide relevance. Then, in this work we present an implementation of document ranking within the BR model whose computational complexity is reduced with respect to the direct application of [16]. This way, the actual applicability of the theoretical framework described in [16] is ensured.

Besides classical representation of documents, i.e. sets of terms, more expressive representations are introduced. Consequently, one can build more expressive IR systems having efficient procedures for computing similarity. Those partial representations are very helpful to include retrieval situations in the model [15]. Furthermore, representing documents with general propositional formulas can improve precision, specially in specific domains. From the algorithms presented in this paper, one can also extract an analysis of the tradeoff between the expressiveness of documents and queries and the efficiency of the system.

The rest of the paper is organized as follows. Section 2 presents how the BR framework was used in [16] to obtain a similarity measure. Section 3 shows the algorithm developed for the case when documents and queries represent sets of terms and the algorithm that can deal with more expressive documents and queries. Section 4 depicts the performance results of some experiments we have done with the algorithms. Some points of discussion and future lines of work are presented in section 5. The paper ends with some conclusions.

2 A similarity measure using Belief Revision

Belief Revision addresses the problem of accommodating a new piece of information into a knowledge base. A key issue within the BR theory is to keep consistency in the knowledge base when contradicting new information arrives. A BR operator has to establish a method for selecting the information that must remain after the arrival of contradicting information. The *Principle of Minimal Change*

states that as much old information as possible should be preserved. This principle stands on the basis of any reasonable revision operator. Model-based approaches to BR establish an order among logical interpretations. Next we briefly sketch the similarity measure based on BR described in [16].

First, we present some preliminaries. In this paper we focus on propositional languages. The propositional alphabet is denoted by $\mathcal{P}$. Interpretations are functions from the propositional alphabet, $\mathcal{P}$, to the set $\{true, false\}$. A model of a formula is an interpretation that makes the formula true and $Mod(\psi)$ denotes the set containing all the models of the formula ψ. The symmetric difference between two sets A and B, $A \triangle B$, is defined as $A \triangle B = (A \cup B) \setminus (A \cap B)$, where $\setminus$ is the regular set difference.

Dalal's revision operator [4] defines the difference between two interpretations as the set of propositional letters on which they differ.

$Diff(I, J) = \{p \in \mathcal{P} | I \models p \text{ iff } J \models \neg p\}$

Then, a measure of distance between interpretations, can be obtained from the number of differing propositional letters.

$Dist(I, J) = |Diff(I, J)|$.

In the following we represent an interpretation by the set of propositional letters that it maps into true. Therefore, the difference between two interpretations can be computed using the symmetric difference between their respective sets.

The distance between the set of models of a formula ψ and a given interpretation I is defined as the distance from I to its closest interpretation in $Mod(\psi)$:

$Dist(Mod(\psi), I) = min_{M \in Mod(\psi)} Dist(M, I)$

Given a formula ψ, an order between interpretations can be derived from the closeness of each interpretation to the set of models of the formula ψ. That is, for any formula ψ, Dalal's total pre-order $\leq_\psi$ is defined as:

$I \leq_\psi J$ iff $Dist(Mod(\psi), I) \leq Dist(Mod(\psi), J)$

Given a theory ψ to be revised with a new information μ, $\psi \circ_D \mu$ denotes the theory revised by Dalal's operator. The models of the revised theory are the models of the new information that are the closest to the theory:

$Mod(\psi \circ_D \mu) = Min(Mod(\mu), \leq_\psi)$

This framework was applied to IR as follows. Each propositional letter of the alphabet $\mathcal{P}$ represents one index term. Queries are modeled as propositional formulas that in the BR framework play the role of theories. Documents are propositional formulas and play the role of new informations. Thus, following Dalal's development, a notion of closeness to the query can be obtained within the revision $q \circ_D d$. If a document is represented by a formula that has only one model, each document can be identified by its only model, and the measure of distance from the model of the document to the set of models of a query can be regarded as a measure of distance from the document to the query itself.

With partial document representations, documents have several models. The measure of distance from the document to the query is the average of the distances from each model of the document to the set of models of the query.

$distance(d, q) = \frac{\sum_{m \in Mod(d)} dist(Mod(q), m)}{|Mod(d)|}$

Note that the situation described before, when a document is identified by its only model is a particular case of the previous formula. A similarity measure, $BRsim$, can be directly defined from $distance$ by normalization. Some examples of the computation of this measure can be found in section 3.

In [16] a restricted document representation was proposed in order to get some equivalences with classical measures. Specifically, a classical vector with binary weights was modeled as a conjunction of propositional letters where each letter of the alphabet appears, either positive or negative. In this case document representations have only one model and $BRsim$ is equivalent to the inner product query-document similarity measure. Partial representations of documents have more than one model. It is in this general case, when the logical model implies a significant improvement respect to classical models because a greater expressiveness of the representations is achieved.

3 Implementing Document Ranking

The translation of model-based BR approaches into efficient algorithms has been a problem of great concern in the BR community. In general, it has become hard to develop efficient algorithms for solving large problems. In [8] a lower bound for knowledge base revision was identified: base revision is at least as hard as deciding propositional satisfiability, which is a well known NP-Complete problem. In fact, Dalal showed [4] that his revision is an NP-Complete problem and provided a method for computing it. However, some studies have demonstrated that restricted problems can be solved within limited bounds [2]. Liberatore and Schaerf [14] identified reductions from circumscription into BR and vice versa. However, to rank documents we need the distances used within the BR process and the reduction to a circumscription problem produces directly the final formula of the revised theory.

A direct translation of Dalal's revision into an algorithm requires a table of symmetric differences between all the models of the theory and all the models of the new information. This computation takes exponential time. Then, even when the document has only one model, all the models of the query have to be computed. On the contrary, the algorithms presented in this work do not compute all the models of the theory and the new information. The basic assumption is that both query and document have to be expressed 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 $l_1 \wedge l_2 \wedge \ldots$, where each l_j is a literal, i.e. a propositional letter or its negation. A DNF formula can be represented as a set of clauses $\psi = \{\psi_1, \psi_2, \ldots\}$. Each clause is a set of literals representing their conjunction, i.e. a clause represents a c_j. The whole set represents the disjunction of all the clauses. The important point is that a conjunction of literals can be thought as a partial model, representing the set of models resulting from fixing the truth value of the atoms appearing in the conjunction and combining the truth value of the atoms non appearing in the conjunction.

Instead of a measure of distance between interpretations, a measure of distance between clauses, $CDist$, is defined. The difference between two clauses, ψ_i and μ_j, is the set of literals in ψ_i whose negation is in μ_j:

$CDiff(\psi_i, \mu_j) = \{l \in \psi_i | \neg l \in \mu_j\}$

The distance between two clauses is given by the cardinality of their difference:

$CDist(\psi_i, \mu_j) = |CDiff(\psi_i, \mu_j)|$

3.1 Algorithm for the simple case

The simple case arises when both document and query represent sets of terms. In classical IR this case corresponds to a representation as a vector for both elements. On the logical side, this corresponds to the fact that representations are conjunctions of propositional letters. In this case, query and document are directly in DNF form and can be both represented as a set with one clause, i.e. $\psi = \{\psi_1\}$ and $\mu = \{\mu_1\}$.

Algorithm 1:

Procedure Similarity(ψ,μ)
  Input:query $\psi = \{\psi_1\}$
        document $\mu = \{\mu_1\}$
        Output:BRsim(ψ,μ)

1.Compute $CDist(\psi_1, \mu_1)$
2.$distance = CDist(\psi_1, \mu_1) + \frac{|\psi_1 \setminus \psi_1 \cap \mu_1| - CDist(\psi_1, \mu_1)}{2}$
3.Return $(1 - \frac{distance}{|\psi_1|})$

The value of $CDist(\psi_1, \mu_1)$ represents the number of literals in ψ_1 -query terms- that appear in μ_1 -document terms- with opposite value. This means that any pair of models of ψ and μ will differ on the interpretation for these terms. Then, all the models of μ have to fare at least $CDist(\psi_1, \mu_1)$ to any model of ψ. That is the reason why $CDist(\psi_1, \mu_1)$ is directly added to $distance$. The set $\psi_1 \setminus \psi_1 \cap \mu_1$ contains the literals in ψ_1 that do not belong to μ_1. Therefore, the value $|\psi_1 \setminus \psi_1 \cap \mu_1| - CDist(\psi_1, \mu_1)$ is the number of literals in ψ_1 whose letter does not appear in μ_1, either positive or negative. As these letters do not appear in the representation of μ, half of the models of μ will map the letter into true and the other half will map it into false. On the other hand, and as a consequence of the presence of the literal in ψ_1, all the models of ψ have to map that letters into the same truth value. Therefore, whatever this fixed truth value is, half of the models of μ will have the opposite one. This produces an increment of $\frac{|\psi_1 \setminus \psi_1 \cap \mu_1| - CDist(\psi_1, \mu_1)}{2}$ in the distance. Finally, the distance is transformed into a similarity value in the interval [0,1]. This normalization uses the fact that the greatest value of $distance$ is $|\psi_1|$.

The computation of $CDist(\psi_1, \mu_1)$ in step 1 can be done traversing the literals in ψ_1 and checking whether the opposite literal belongs to μ_1. It can also be done with the reciprocal process, that is, traversing μ_1 and checking in ψ_1. Each check can be done in unit time because an array can be used to store what literals belong to a clause. Then, step 1 can be done in linear time w.r.t the size of ψ_1 or μ_1. Due to similar reasons, the computation of $|\psi_1 \setminus \psi_1 \cap \mu_1|$ in step 2 can also be accomplished in linear time respect to the size of any clause. Consequently, this algorithm can be run in linear time w.r.t the size of either ψ_1 or μ_1. As ψ_1 represents the query and μ_1 the document, ψ_1 is expected to have less literals than μ_1 and the most efficient implementation of the algorithm can be done with complexity $\mathcal{O}(|\psi_1|)$.

Let us analyze the use of the previous algorithm for IR. Classical systems consider representations of documents that have information about the presence or absence for all the index terms. On the logical side, this case corresponds with the fact that documents are total theories, i.e. μ_1 contains all the index terms either positive or negative. As a consequence, all the query terms appear in μ_1 and $distance = CDist(\psi_1, \mu_1)$. As $CDist(\psi_1, \mu_1)$ counts the number of *differing* terms between the query and the document, the result of the algorithm (after the normalization) is equivalent to the inner product query-document matching function. On the other hand, when a document is a partial theory its representation does not store information about all the index terms. This is not a regular assumption in classical systems. Let us think about a query that mentions one of those index terms that do not appear in the document representation. In this case, the model does not assume that the document is (or is not) really about that index term. On the contrary, it considers a value of distance of 0.5 for the query terms not present in the document representation. This behavior is captured in the formula by $\frac{|\psi_1 \setminus \psi_1 \cap \mu_1| - CDist(\psi_1, \mu_1)}{2}$.

It is important to note that algorithm 1 ensures an efficient implementation of the model proposed in [16]. Algorithm 1 deals with the constrained representations proposed in that work and it computes similarity in linear time w.r.t the size of the query.

Example 1: In this example we show the computation of BRsim using tables of symmetric differences between interpretations and the computation of BRsim using Algorithm 1. Let the propositional alphabet $\mathcal{P}$, the documents d_1 and d_2 and the query q be defined as:

$\mathcal{P} = \{a, b, c, d, e\}$
$q = a \wedge c$
$d_1 = \neg a \wedge b$
$d_2 = a \wedge \neg b \wedge c$

The computation of similarity from symmetric differences between interpretations is shown in fig. 1. The following lines depict the computation of the similarity using Algorithm 1.

Document d_1

Input:
  Query (DNF): $\psi = \{\psi_1\},\ \psi_1 = \{a, c\}$
  Document (DNF): $\mu = \{\mu_1\},\ \mu_1 = \{\neg a, b\}$

1. $CDist(\psi_1, \mu_1) = |\{l \in \psi_1 | \neg l \in \mu_1\}| = |\{a\}| = 1$
2. $distance = 1 + \frac{|\{a,c\} \setminus \emptyset| - 1}{2} = 1.5$
3. $1 - \frac{distance}{|\psi_1|} = 1 - \frac{1.5}{2} = 0.25$. Return(0.25).

Document d_2

Input:
  Query (DNF): $\psi = \{\psi_1\},\ \psi_1 = \{a, c\}$
  Document (DNF): $\mu = \{\mu_1\},\ \mu_1 = \{a, \neg b, c\}$

1. $CDist(\psi_1, \mu_1) = |\{l \in \psi_1 | \neg l \in \mu_1\}| = |\emptyset| = 0$
2. $distance = 0 + \frac{|\{a,c\} \setminus \{a,c\}| - 0}{2} = 0$
3. $1 - \frac{distance}{|\psi_1|} = 1 - \frac{0}{2} = 1$. Return(1).

Note that using tables of symmetric differences we need to compute a lot of models and distances between them. On the other hand, algorithm 1 only takes a few steps and it does not need any model.

3.2 Extending the expressiveness of documents and queries

Previous section has shown an efficient implementation of the ranking process within a logical framework. The model subsumes the vector model with binary weights for both query and document. However that equivalence was achieved at the expense of restricting the expressiveness of documents and queries. In this section we consider representations of documents and queries containing both conjunctions and disjunctions. Documents and queries are now represented as general DNF formulas. An important point is that document representations are now enriched to include both connectors what means a substantial improvement in the expressiveness of the model. Queries can have also conjunctions and disjunctions but this is not a distinctive feature of the model. In fact, Boolean Model allows queries with any combination of AND's and OR's. The fact that query representations are DNF formulas does not imply that users have to articulate their information needs in this form, but the system translates user information needs into DNF form. Any propositional formula can be translated into its DNF equivalent.

In this section we develop an algorithm which computes the similarity measure between a document and a query, both in DNF. It is important to note that now the measure is not equivalent to a classical one because document representations are different from those used by classical models.

The algorithm traverses the set of models of the document μ, and for each model computes its distance to the query ψ. These distances are accumulated and, finally, the total number of models of μ is used to get the average of the distances to the query. The advantage stands on the fact that no models of the query are needed. The distance from each model of the document to the query is computed transforming the model to a set of literals and comparing with each conjunction of the query. Reflecting Dalal's semantics, the least distance is selected to be the distance from the model to the query.

Before developing the algorithm some preliminaries are shown. The size of the propositional alphabet will be denoted by S, $S = |\mathcal{P}|$. As it has been said before, an interpretation is denoted by the set of letters mapped into true. Then, given an interpretation m, $LIT(m)$ represents the transformation of m into a set of literals, i.e. $LIT(m) = m \cup \{\neg l | l \in \mathcal{P} \setminus m\}$. The symbols ψ_{min} and ψ_{max} are the size of the smallest and the biggest clause in ψ, respectively. The symbol μ_{max} represents the maximum size of a clause in μ.

Algorithm 2:

Procedure Similarity(ψ,μ)
  Input:query $\psi = \{\psi_1, \psi_2, \ldots\}$
        document $\mu = \{\mu_1, \mu_2, \ldots\}$
        Output:BRsim(ψ,μ)

1. $Distance = 0; Total\_Models = 0;$
2. Compute the set of models of μ
3. Extract a new m, model of μ
4. $Distance\_to\_\psi = S$
5. Extract a new $\psi_i \in \psi$
6. Compute $CDist(\psi_i, LIT(m))$
7. If $CDist(\psi_i, LIT(m)) < Distance\_to\_\psi$ then $Distance\_to\_\psi = CDist(\psi_i, LIT(m))$
8. Go to step 5 until no more ψ_is remain
9. $Total\_Models++; Distance += Distance\_to\_\psi$
10. Go to step 3 until no more μ models remain
11. $distance(d, q) = \frac{Distance}{Total\_Models}$
12. Return($1 - \frac{distance(d,q)}{\psi_{min}}$)

Document d_1

Symmetric differences between query and document models:

Document models $\rightarrow$ Query models $\downarrow$	d_1 $\{b\}$	$\{b,c\}$	$\{b,d\}$	$\{b,e\}$	$\{b,c,d\}$	$\{b,c,e\}$	$\{b,d,e\}$	$\{b,c,d,e\}$
$\{a,c\}$	$\{a,b,c\}$	$\{a,b\}$	$\{a,b,c,d\}$	$\{a,b,c,e\}$	$\{a,b,d\}$	$\{a,b,e\}$	$\{a,b,c,d,e\}$	$\{a,b,d,e\}$
$\{a,b,c\}$	$\{a,c\}$	$\{a\}$	$\{a,c,d\}$	$\{a,c,e\}$	$\{a,d\}$	$\{a,e\}$	$\{a,c,d,e\}$	$\{a,d,e\}$
$\{a,c,d\}$	$\{a,b,c,d\}$	$\{a,b,d\}$	$\{a,b,c\}$	$\{a,b,c,d,e\}$	$\{a,b\}$	$\{a,b,d,e\}$	$\{a,b,c,e\}$	$\{a,b,e\}$
$\{a,c,e\}$	$\{a,b,c,e\}$	$\{a,b,e\}$	$\{a,b,c,d,e\}$	$\{a,b,c\}$	$\{a,b,d,e\}$	$\{a,b\}$	$\{a,b,c,d\}$	$\{a,b,d\}$
$\{a,b,c,d\}$	$\{a,c,d\}$	$\{a,d\}$	$\{a,c\}$	$\{a,c,d,e\}$	$\{a\}$	$\{a,d,e\}$	$\{a,c,e\}$	$\{a,e\}$
$\{a,b,c,e\}$	$\{a,c,e\}$	$\{a,e\}$	$\{a,c,d,e\}$	$\{a,c\}$	$\{a,d,e\}$	$\{a\}$	$\{a,c,d\}$	$\{a,d\}$
$\{a,c,d,e\}$	$\{a,b,c,d,e\}$	$\{a,b,d,e\}$	$\{a,b,c,e\}$	$\{a,b,c,d\}$	$\{a,b,e\}$	$\{a,b,d\}$	$\{a,b,c\}$	$\{a,b\}$
$\{a,b,c,d,e\}$	$\{a,c,d,e\}$	$\{a,d,e\}$	$\{a,c,e\}$	$\{a,c,d\}$	$\{a,e\}$	$\{a,d\}$	$\{a,c\}$	$\{a\}$

Cardinalities and computation of the distance:

Document models $\rightarrow$ Query models $\downarrow$	d_1 $\{b\}$	$\{b,c\}$	$\{b,d\}$	$\{b,e\}$	$\{b,c,d\}$	$\{b,c,e\}$	$\{b,d,e\}$	$\{b,c,d,e\}$
$\{a,c\}$	3	2	4	4	3	3	5	4
$\{a,b,c\}$	2	1	3	3	2	2	4	3
$\{a,c,d\}$	4	3	3	5	2	4	4	3
$\{a,c,e\}$	4	3	5	3	4	2	4	3
$\{a,b,c,d\}$	3	2	2	4	1	3	3	2
$\{a,b,c,e\}$	3	2	4	2	3	1	3	2
$\{a,c,d,e\}$	5	4	4	4	3	3	3	2
$\{a,b,c,d,e\}$	4	3	3	3	2	2	2	1
$dist(Mod(q),m_i) = min_{m \in Mod(q)} dist(m,m_i)$	2	1	2	2	1	1	2	1
$distance(d,q) = \frac{\sum_{m \in Mod(d)} dist(Mod(q),m)}{\lvert Mod(d) \rvert}$	$\frac{12}{8} = 1.5$							

Finally, $BRsim(d,q)$ is computed from $distance(d,q)$ using k, the number of literals appearing in the query:

$$BRsim(d,q) = 1 - \frac{distance(d,q)}{k}$$

Therefore, in the example $BRsim(d_1,q) = 1 - \frac{1.5}{2} = 0.25$

Document d_2

Symmetric differences between query and document models:

Document models $\rightarrow$ Query models $\downarrow$	d_2 $\{a,c\}$	$\{a,c,d\}$	$\{a,c,e\}$	$\{a,c,d,e\}$
$\{a,c\}$	$\emptyset$	$\{d\}$	$\{e\}$	$\{d,e\}$
$\{a,b,c\}$	$\{b\}$	$\{b,d\}$	$\{b,e\}$	$\{b,d,e\}$
$\{a,c,d\}$	$\{d\}$	$\emptyset$	$\{d,e\}$	$\{e\}$
$\{a,c,e\}$	$\{e\}$	$\{d,e\}$	$\emptyset$	$\{d\}$
$\{a,b,c,d\}$	$\{b,d\}$	$\{b\}$	$\{b,d,e\}$	$\{b,e\}$
$\{a,b,c,e\}$	$\{b,e\}$	$\{b,d,e\}$	$\{b\}$	$\{b,d\}$
$\{a,c,d,e\}$	$\{d,e\}$	$\{e\}$	$\{d\}$	$\emptyset$
$\{a,b,c,d,e\}$	$\{b,d,e\}$	$\{b,e\}$	$\{b,d\}$	$\{b\}$

Cardinalities and computation of the distance:

Document models $\rightarrow$ Query models $\downarrow$	d_2 $\{a,c\}$	$\{a,c,d\}$	$\{a,c,e\}$	$\{a,c,d,e\}$
$\{a,c\}$	0	1	1	2
$\{a,b,c\}$	1	2	2	3
$\{a,c,d\}$	1	0	2	1
$\{a,c,e\}$	1	2	0	1
$\{a,b,c,d\}$	2	1	3	2
$\{a,b,c,e\}$	2	3	1	2
$\{a,c,d,e\}$	2	1	1	0
$\{a,b,c,d,e\}$	3	2	2	1
$dist(Mod(q),m_i) = min_{m \in Mod(q)} dist(m,m_i)$	0	0	0	0
$distance(d,q) = \frac{\sum_{m \in Mod(d)} dist(Mod(q),m)}{\lvert Mod(d) \rvert}$	0			

Finally, $BRsim(d,q)$ is computed from $distance(d,q)$ using k, the number of literals appearing in the query:

$$BRsim(d,q) = 1 - \frac{distance(d,q)}{k}$$

For the current example, $BRsim(d_2,q) = 1 - \frac{0}{2} = 1$

Figure 1. Computation of similarity from symmetric differences between interpretations

Step 3 extracts each model m of μ. The variable $Distance\_to\_\psi$ stores the distance from the model m to the set of models of the query. However, it is not necessary to compute any model of the query. Instead, the representation of each model of μ as a conjunction ($LIT(m)$) is compared with every conjunction ψ_i of the query and the smallest distance is stored in $Distance\_to\_\psi$. $Distance\_to\_\psi$ is initialized to its greatest value, the size of the propositional alphabet. The variable $Distance$ stores the sum of the distances from each model of μ to ψ. Finally, dividing by the total number of models, the distance from the document to the query is obtained. The similarity measure normalized in the interval $[0, 1]$ is obtained using the fact that the distance from a model to the set of models of the query ψ is at most the size of the smallest ψ_i. The reason is that the distance from a model m to the set of models of a query ψ takes the value of the distance from m to the closest ψ model. For any clause ψ_i of ψ

Document d_1

Symmetric differences between query and document models:

Document models $\rightarrow$	d_1		
Query models $\downarrow$	$\{a, b, c\}$	$\{a, b, c, d\}$	$\{a, b, d\}$
$\{a, c\}$	$\{b\}$	$\{b, d\}$	$\{b, c, d\}$
$\{a, b, c\}$	$\emptyset$	$\{d\}$	$\{c, d\}$
$\{a, c, d\}$	$\{b, d\}$	$\{b\}$	$\{b, c\}$
$\{a, b, c, d\}$	$\{d\}$	$\emptyset$	$\{c\}$
$\{a, d\}$	$\{b, c, d\}$	$\{b, c\}$	$\{b\}$
$\{a, b, d\}$	$\{c, d\}$	$\{c\}$	$\emptyset$

Cardinalities and computation of the distance:

Document models $\rightarrow$	d_1		
Query models $\downarrow$	$\{a, b, c\}$	$\{a, b, c, d\}$	$\{a, b, d\}$
$\{a, c\}$	1	2	3
$\{a, b, c\}$	0	1	2
$\{a, c, d\}$	2	1	2
$\{a, b, c, d\}$	1	0	1
$\{a, d\}$	3	2	1
$\{a, b, d\}$	2	1	0
$dist(Mod(q), m_i) = min_{m \in Mod(q)} dist(m, m_i)$	0	0	0
$distance(d, q) = \frac{\sum_{m \in Mod(d)} dist(Mod(q), m)}{\lvert Mod(d) \rvert}$	0		

Finally, $BRsim(d_1, q)$ is computed from $distance(d_1, q)$ using ψ_{min} the size of the smallest conjunction in ψ:

$$BRsim(d, q) = 1 - \frac{distance(d,q)}{\psi_{min}}$$

The similarity measure between d_1 and q is $BRsim(d_1, q) = 1 - \frac{0}{2} = 1$.

Document d_2

Symmetric differences between query and document models:

Document models $\rightarrow$	d_2					
Query models $\downarrow$	$\{b, c\}$	$\{a, b, c\}$	$\{b, c, d\}$	$\{a, b, c, d\}$	$\{a, b\}$	$\{a, b, d\}$
$\{a, c\}$	$\{a, b\}$	$\{b\}$	$\{a, b, d\}$	$\{b, d\}$	$\{b, c\}$	$\{b, c, d\}$
$\{a, b, c\}$	$\{a\}$	$\emptyset$	$\{a, d\}$	$\{d\}$	$\{c\}$	$\{c, d\}$
$\{a, c, d\}$	$\{a, b, d\}$	$\{b, d\}$	$\{a, b\}$	$\{b\}$	$\{b, c, d\}$	$\{b, c\}$
$\{a, b, c, d\}$	$\{a, d\}$	$\{d\}$	$\{a\}$	$\emptyset$	$\{c, d\}$	$\{c\}$
$\{a, d\}$	$\{a, b, c, d\}$	$\{b, c, d\}$	$\{a, b, c\}$	$\{b, c\}$	$\{b, d\}$	$\{b\}$
$\{a, b, d\}$	$\{a, c, d\}$	$\{c, d\}$	$\{a, c\}$	$\{c\}$	$\{d\}$	$\emptyset$

Cardinalities and computation of the distance:

Document models $\rightarrow$	d_2					
Query models $\downarrow$	$\{b, c\}$	$\{a, b, c\}$	$\{b, c, d\}$	$\{a, b, c, d\}$	$\{a, b\}$	$\{a, b, d\}$
$\{a, c\}$	2	1	3	2	2	3
$\{a, b, c\}$	1	0	2	1	1	2
$\{a, c, d\}$	3	2	2	1	3	2
$\{a, b, c, d\}$	2	1	1	0	2	1
$\{a, d\}$	4	3	3	2	2	1
$\{a, b, d\}$	3	2	2	1	1	0
$dist(Mod(q), m_i) = min_{m \in Mod(q)} dist(m, m_i)$	1	0	1	0	1	0
$distance(d, q) = \frac{\sum_{m \in Mod(d)} dist(Mod(q), m)}{\lvert Mod(d) \rvert}$	0.5					

Finally, $BRsim(d_2, q) = 1 - \frac{distance(d_2,q)}{\psi_{min}} = 1 - \frac{0.5}{2} = 0.75$

Figure 2. Computation of similarity from symmetric differences between interpretations

Size of the propositional alphabet: S=4

1. Document d_1

   Input:

   Query in DNF form: $\psi = \{\psi_1, \psi_2\},\ \psi_1 = \{a, c\}, \psi_2 = \{a, d\}$

   Document d_1 in DNF form: $\mu = \{\mu_1, \mu_2\},\ \mu_1 = \{a, b, c\}, \mu_2 = \{a, b, d\}$

   1. $Distance = 0; Total\_Models = 0;$

   2. $Mod(\mu) = \{\{a, b, c\}, \{a, b, c, d\}, \{a, b, d\}\}$

   3. $m = \{a, b, c\},\ LIT(m) = \{a, b, c, \neg d\},$ 4. $Distance\_to\_\psi = 4$

      5. $\psi_1 = \{a, c\},$ 6. $Dist(\psi_1, LIT(m)) = 0,$ 7. $Distance\_to\_\psi = 0$

      5. $\psi_2 = \{a, d\},$ 6. $Dist(\psi_2, LIT(m)) = 1$

   9. $Total\_Models = 1; Distance = 0$

   3. $m = \{a, b, c, d\},\ LIT(m) = \{a, b, c, d\},$ 4. $Distance\_to\_\psi = 4$

      5. $\psi_1 = \{a, c\},$ 6. $Dist(\psi_1, LIT(m)) = 0,$ 7. $Distance\_to\_\psi = 0$

      5. $\psi_2 = \{a, d\},$ 6. $Dist(\psi_2, LIT(m)) = 0$

   9. $Total\_Models = 2; Distance = 0$

   3. $m = \{a, b, d\},\ LIT(m) = \{a, b, \neg c, d\},$ 4. $Distance\_to\_\psi = 4$

      5. $\psi_1 = \{a, c\},$ 6. $Dist(\psi_1, LIT(m)) = 1,$ 7. $Distance\_to\_\psi = 1$

      5. $\psi_2 = \{a, d\},$ 6. $Dist(\psi_2, LIT(m)) = 0,$ 7. $Distance\_to\_\psi = 0$

   9. $Total\_Models = 3; Distance = 0$

   11. $distance(d, q) = \frac{0}{3} = 0$

   12. Return(1)

2. Document d_2

   Input:

   Query in DNF form: $\psi = \{\psi_1, \psi_2\},\ \psi_1 = \{a, c\}, \psi_2 = \{a, d\}$

   Document d_2 in DNF form: $\mu = \{\mu_1, \mu_2\},\ \mu_1 = \{b, c\}, \mu_2 = \{a, b\}$

   1. $Distance = 0; Total\_Models = 0;$

   2. $Mod(\mu) = \{\{b, c\}, \{a, b, c\}, \{b, c, d\}, \{a, b, c, d\}, \{a, b\}, \{a, b, d\}\}$

   3. $m = \{b, c\},\ LIT(m) = \{\neg a, b, c, \neg d\},$ 4. $Distance\_to\_\psi = 4$

      5. $\psi_1 = \{a, c\},$ 6. $Dist(\psi_1, LIT(m)) = 1,$ 7. $Distance\_to\_\psi = 1$

      5. $\psi_2 = \{a, d\},$ 6. $Dist(\psi_2, LIT(m)) = 2$

   9. $Total\_Models = 1; Distance = 1$

   3. $m = \{a, b, c\},\ LIT(m) = \{a, b, c, \neg d\},$ 4. $Distance\_to\_\psi = 4$

      5. $\psi_1 = \{a, c\},$ 6. $Dist(\psi_1, LIT(m)) = 0,$ 7. $Distance\_to\_\psi = 0$

      5. $\psi_2 = \{a, d\},$ 6. $Dist(\psi_2, LIT(m)) = 1$

   9. $Total\_Models = 2; Distance = 1$

   3. $m = \{b, c, d\},\ LIT(m) = \{\neg a, b, c, d\},$ 4. $Distance\_to\_\psi = 4$

      5. $\psi_1 = \{a, c\},$ 6. $Dist(\psi_1, LIT(m)) = 1,$ 7. $Distance\_to\_\psi = 1$

      5. $\psi_2 = \{a, d\},$ 6. $Dist(\psi_2, LIT(m)) = 1$

   9. $Total\_Models = 3; Distance = 2$

   3. $m = \{a, b, c, d\},\ LIT(m) = \{a, b, c, d\},$ 4. $Distance\_to\_\psi = 4$

      5. $\psi_1 = \{a, c\},$ 6. $Dist(\psi_1, LIT(m)) = 0,$ 7. $Distance\_to\_\psi = 0$

      5. $\psi_2 = \{a, d\},$ 6. $Dist(\psi_2, LIT(m)) = 0$

   9. $Total\_Models = 4; Distance = 2$

   3. $m = \{a, b\},\ LIT(m) = \{a, b, \neg c, \neg d\},$ 4. $Distance\_to\_\psi = 4$

      5. $\psi_1 = \{a, c\},$ 6. $Dist(\psi_1, LIT(m)) = 1,$ 7. $Distance\_to\_\psi = 1$

      5. $\psi_2 = \{a, d\},$ 6. $Dist(\psi_2, LIT(m)) = 1$

   9. $Total\_Models = 5; Distance = 3$

   3. $m = \{a, b, d\},\ LIT(m) = \{a, b, \neg c, d\},$ 4. $Distance\_to\_\psi = 4$

      5. $\psi_1 = \{a, c\},$ 6. $Dist(\psi_1, LIT(m)) = 1,$ 7. $Distance\_to\_\psi = 1$

      5. $\psi_2 = \{a, d\},$ 6. $Dist(\psi_2, LIT(m)) = 0,$ 7. $Distance\_to\_\psi = 0$

   9. $Total\_Models = 6; Distance = 3$

   11. $distance(d, q) = \frac{3}{6} = 0.5$

   12. Return($1 - \frac{0.5}{2}$)

Figure 3. Computation of similarity with Algorithm 2

Size of the alphabet	Number of doc. terms	Number of query terms	Run time (microsec.)	Number of doc. models
25	25	1	18	1
25	25	5	31	1
25	25	10	45	1
25	10	1	15	2^{15}
25	10	5	20	2^{15}
25	10	10	31	2^{15}
50	50	1	19	1
50	50	5	47	1
50	50	10	89	1
50	25	1	17	2^{25}
50	25	5	33	2^{25}
50	25	10	59	2^{25}
50	10	1	16	2^{40}
50	10	5	22	2^{40}
50	10	10	34	2^{40}
500	50	1	23	2^{450}
500	50	5	55	2^{450}
500	50	10	92	2^{450}
500	250	1	160	2^{250}
500	250	5	319	2^{250}
500	250	10	470	2^{250}
500	500	1	263	1
500	500	5	474	1
500	500	10	799	1

Figure 4. Run Time Performance of Alg.1

with polynomial complexity analysis. Therefore, the definition of a similarity measure of this kind and the analysis of its appropriateness for IR modeling, appear as future lines of work.

5 Discussion and Future Work

Some IR models consider representations of documents with information about presence/absence for all the keywords of the indexing vocabulary. In systems with such a *total information assumption* a logical approach hardly improves performance. In these cases, logical approaches provide a formal and homogeneous framework where classical models can be analyzed but their impact is somewhat reduced. This is the case of the model of [16] with documents as total theories. In that case, the measure $BRsim$ is equivalent to the inner product query-document similarity measure with binary weights. However, a more natural assumption is to consider documents as partial descriptions [20]. It is in this case when logical models can produce a significant improvement with respect to classical ones.

The efficiency of the first algorithm presented in this work permits its application for large-scale IR systems. As well as representing classical vectors, Algorithm 1 allows us to express *partial* vectors. This kind of representations can be very helpful when considering *retrieval situations*. We use the name of retrieval situations to refer to several aspects affecting the relevance judgment and not captured by a simple matching between topics. User's knowledge and intentions, pragmatics of the language, etc. are factors that should be taken into account by a system when deciding relevance. This necessity was already pointed out by Nie and other researchers [17], who outlined the use of counterfactual conditional logic for modeling situational aspects. We can imagine a realistic scenario where a partial description d of each document is maintained. When a user articulates a query q, the system takes the representation of the current retrieval situation S, and *revises* it with the document representation, i.e. it makes $S \circ d$. The result of $S \circ d$ represents the adaptation of the document to the current retrieval situation and it is likely *less partial* than d. Then, $S \circ d$ is matched with the query representation q in order to decide relevance. Let us consider an example. A document about the ml programming language could be indexed by the index term ml without mentioning the relationship between ml and computer science (cs). A common user may not know that the language ml has something to do with computer science but an experienced user would probably now that they are related concepts. That knowledge would be part of their respective user profiles. Then, if both users articulate a query asking about cs documents and the document representation is contrasted with the user profiles, the first user would not access the document while the second one would. This goes in the line that unexperienced users receive generic documents while experienced users receive specific ones. In fact, it has not much sense to present a ml document to a user that does not know what it is. He/she is probably looking for more general documents about computer science. Then, in some extent, the precision of the set of retrieved documents would be improved.

An important point is that the operation of revision between a retrieval situation and a document can be accomplished using a BR operator. Note that the use of BR would be different here than in [16]. In that work, the measures between logical interpretations within the BR process $q \circ d$ were used to build an estimation of $P(d \models q)$ but the final result of the revision was left aside. For the new application, the result of the revision, $S \circ d$, would be used to compute its similarity with respect to the query q. For that task, the techniques presented in [16] can be generalized, so that the BR process $q \circ (S \circ d)$ gives us a measure of $P((S \circ d) \models q)$. There are some important results in the field of BR that can contribute to establish BR as a paradigm for IR. In particular, del Val [6, 5] identified some particular cases that do not need to measure distances between logical interpretations in order to obtain the revised theory. Therefore, the computation of $S \circ d$ can be executed in polynomial time. Del Val uses a syntactic characterization whose basic idea is to manage clauses of literals instead of managing interpretations. We have used this technique to construct the algorithms presented in section 3. The management of retrieval situations and partial descriptions of documents and the adequacy of several revision methods for combining them has been thoroughly studied in a recent work [15].

Now we discuss the usefulness of representing documents with unrestricted propositional formulas. Conventional IR systems giving very good performance even with huge amounts of data, stand on representations of documents as sets of terms. However, several applications using IR techniques as their underlying technology require more expressive document representations. For instance, OntoSeek [10] is a system specially designed for yellow pages and product catalogs that considers structured representations and uses linguistic resources such as WordNet. Fuhr [7] also pointed out that new IR applications, dealing with structured documents, hypertext, multimedia objects, etc. need more expressive formalisms. For instance, spatial and temporal relationships within multimedia objects or complex terminologies cannot be represented with a simple set of terms. Another recent example can be found in [18], where hierarchies of concepts were used to organize a set of documents. Furthermore, in environments like image retrieval, where a total content analysis cannot be attainable, it is interesting to have a method to express several alternatives. For instance, a process of image recognition can produce that a shape is something like a deer or a horse and a vector representation would impose a simplification. In fact, the application of text retrieval techniques for digital pictures is limited [9] and using a set of captions to express image content is inappropriate. A document description using more expressive languages allows the articulation of sophisticated queries, which are more closer to user's view, and provides an increment in precision and reasoning support.

A major challenge is that classical IR systems, working with unstructured text, get benefit from the more expressive representations proposed in this work. Then, a very important line of future work is the design of experiments that, starting from plain text build automatically a representation of the document as an unrestricted propositional formula. In this sense, some works in Passage Retrieval [11, 1] have followed different approaches to split plain texts into different chunks. Instead of matching a query against a vector representing the whole document, the query is matched against each individual vector representing a part of the document. The evaluation of these methods showed improvements in precision. These results are encouraging for our model of documents as DNF formulas.

6 Conclusion

In this work we have implemented document ranking within the logical framework of BR. This way it has been shown that this logical approach can constitute the theoretical basis of a realistic IR system. The model subsumes a classical vector-space model with binary weights and the inner product query-document similarity measure is a particular case of the similarity measure $BRsim$ computed within the logical framework. Specifically, we have developed two algorithms. The first one computes similarity between logical representations of a document and a query that are equivalent to binary vectors. The second algorithm computes similarity between a document and a query, both represented as general DNF propositional formulas. The efficiency of the first algorithm and its capability to work with partial vectors are very helpful to introduce retrieval situations. That way, a system can maintain partial representations of documents and revise them in the light of the current retrieval situation, just before matching with queries. Furthermore, unrestricted propositional formulas are promising representations for future applications. Indeed, representations more expressive than simple sets of terms have been recently claimed for specific domains. In this sense, the automatic attainment of unrestricted propositional formulas for representing documents appears as a challenge in order to make a proper evaluation of our model.

Acknowledgements: This work was supported in part by project PGIDT99XI10201B from the government of Galicia, *Xunta de Galicia*, (Spain) and by project PB97-0228 from the government of Spain.

References

[1] J. Callan. Passage-level evidence in document retrieval. In *Proc. of SIGIR-94, the 17th ACM Conference on Research and Development in Information Retrieval*, pages 302–310, Dublin, July 1994.

[2] T. S.-C. Chou and M. Winslett. Immortal:a model-based belief revision system. In *Proc. of KR-91, the Second Conference on Principles of Knowledge Representation and Reasoning*, pages 99–110, Cambridge,Massachusetts, April 1991.

[3] F. Crestani, M. Lalmas, and C. Rijsbergen, editors. *Information Retrieval, Uncertainty and Logics: advanced models for the representation and retrieval of information*. Kluwer Academic Publishers, Norwell, MA, 1998.

[4] 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.

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

[6] 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.

[7] N. Fuhr. Probabilistic Datalog: Implementing logical information retrieval for advanced applications. *Journal of the American Society for Information Science*, 51(2):95–110, 2000.

[8] P. Gärdenfors and H. Rott. Belief revision. In D. Gabbay, C. Hogger, and J. Robinson, editors, *Handbook of Logic in*

Artificial Intelligence and Logic Programming, volume 4, Epistemic and Temporal Reasoning, pages 35–175. Clarendon Press, Oxford, 1995.

[9] C. Globe and S. Bechhofer. "Fetch me a picture representing triumph or similar": Classification based navigation and retrieval for picture archives. In *Proc. of Seminar on Searching for Information: Artificial Intelligence and Information Retrieval Approaches*, pages 4/1–4/4, Glasgow, UK, 1999.

[10] N. Guarino, C. Masolo, and G. Vetere. OntoSeek:content-based access to the web. *IEEE Intelligent Systems*, 14(3):70–80, 1999.

[11] M. Hearst and C. Plaunt. Subtopic structuring for full-length document access. In *Proc. of SIGIR-93, the 16th ACM Conference on Research and Development in Information Retrieval*, pages 59–68, Pittsburgh, June 1993.

[12] M. Lalmas. Logical models in information retrieval: introduction and overview. *Information Processing & Management*, 34(1):19–33, 1998.

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

[14] P. Liberatore and M. Schaerf. Reducing belief revision to circumscription (and vice versa). *Artificial Intelligence*, 93(1-2):261–296, 1997.

[15] D. Losada and A. Barreiro. Retrieval situations and belief change. In *Proc. of LUMIS-2000, the 2nd International Workshop on Logical and Uncertainty models for Information Systems (to appear)*, Greenwich,UK, September 2000.

[16] 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.

[17] J.-Y. Nie, M. Brisebois, and F. Lepage. Information retrieval as counterfactual. *The Computer Journal*, 38(8):643–657, 1995.

[18] M. Sanderson and B. Croft. Deriving concept hierarchies from text. In *Proc. of SIGIR-99, the 22nd ACM Conference on Research and Development in Information Retrieval*, pages 206–213, Berkeley, August 1999.

[19] C. J. Van Rijsbergen. A non-classical logic for information retrieval. *The Computer Journal*, 29:481–485, 1986.

[20] C. J. van Rijsbergen. Towards an information logic. In *Proc. of SIGIR-89, the 12th ACM Conference on Research and Development in Information Retrieval*, pages 77–86, Cambridge, MA, June 1989.