ISSN 1369-1961

Department of Computer Science

Technical Report No. 770

Workshop on Logical and Uncertainty Models for Information Systems

Edited by:
Fabio Crestani &
Mounia Lalmas

QUEEN MARY
AND WESTFIELD COLLEGE
UNIVERSITY OF LONDON

July 1999

Fifth European Conference on Symbolic and Quantitative Approaches to
Reasoning with Uncertainty

Workshop on Logical and Uncertainty Models for Information Systems

University College, London, United Kingdom

5th July 1999

Fabio Crestani and Mounia Lalmas (editors)

Preface

The advent of electronic tools for producing and storing information has resulted in an avalanche of computer readable text. The access to all this information has gone through a slow but steady process to adapt to the growth of availability of electronically stored data, and a large number of tools have been developed to enable users to access and manage these large volumes of information. But what has uncertainty and logic to do with accessing and managing information stored in a computer?

Uncertainty plays a very important role in the representation, access, and retrieval of information. The representation of information objects is often uncertain. For example, the extraction of index terms from a document or a query to represent the document or the query information content is a highly uncertain process. The data describing the redness of a red object present in a picture stored in a multimedia database is subject to a certain degree of uncertainty too. Probability theory is one way of dealing with uncertainty, but there are other approaches, such as, fuzzy logic, the Dempster-Shafer theory of evidence, imaging, neural networks, and so on.

Logic plays a very important role in the representation, access, and retrieval of information. Logic has proved over centuries to be a very powerful modelling and reasoning tool, providing a degree of formality and correctness that can be very useful for manipulating information objects. For the task of retrieving information, logic has been used to build models that provide a rich and uniform representation of information with the aim of improving retrieval effectiveness. Classical logic is one possible approach. Other approaches include belief revision, situation theory, possible world semantics, abductive logic, conceptual graphs, conditional logic, description logic, and so on.

The purpose of the 1999 Workshop on Logical and Uncertainty Models for Information Systems (LUMIS'99) is to promote discussion and interaction among members of the Information Systems community; in particular among those members with research interests in logical and uncertainty models for the treatment of semi-structured and unstructured information. We believe that a very large part of the information that will be available in future will be of this nature. This community is made of people coming from different fields: theoretical computer science, databases, information retrieval, hypermedia, digital libraries, artificial intelligence, to mention just a few areas. We believe we have succeeded in gathering a heterogeneous community that will benefit considerably from exchanging ideas and experiences. Based on our previous experience in organising the 1995 and 1996 Workshops on Information Retrieval, Uncertainty and Logics (WIRUL), we are confident that this exchange of ideas, of which these proceedings are a reminder, will be very stimulating.

10 June 1999 Fabio Crestani and Mounia Lalmas

Organising Committee

Fabio Crestani — University of Glasgow, Scotland
Mounia Lalmas — Queen Mary & Westfield College, England

Programme Committee

Gianni Amati — Fondazione Ugo Bordoni, Rome, Italy
Peter D. Bruza — Queensland University of Technology, Australia
Theo Huibers — DOXiS, The Netherlands
Adrian Muller — IBM, Germany
Theo Huibers — DOXiS, The Netherlands
Jian-Yun Nie — University of Montreal, Canada
Iadh Ounis — IMAG, Grenoble, France
Gabriella Pasi — ITIM-CNR, Milan, Italy

Table of Contents

A logical formulation of the Boolean model and of weighted Boolean models

Gabriella Pasi

Istituto per le Tecnologie Informatiche Multimediali
Consiglio Nazionale delle Ricerche
Via Ampère 56, 20131 Milano, Italy
gabriella.pasi@itim.mi.cnr.it

Abstract. In this paper the role of logic as a formal basis to exploit the query evaluation process of the boolean model and of weighted boolean models is analysed. The proposed approach is based on the expression of the constraint imposed by a query term on a document representation by means of the implication connective (by a fuzzy implication in the case of weighted terms). A logical formula corresponds to a query evaluation structure, and the degree of relevance of a document to a user query is obtained as the truth value of the formula expressing the evaluation structure of the considered query under the interpretation corresponding with a document and the query itself.

1 Introduction

A recent approach to model information retrieval is the logical approach; the main motivation advocated in the literature to model IR in the logical framework is the need for a more general formal discipline, as logic, to reason about the foundational principles of IR [11, 17].

A common basis of the logical models proposed in the literature is to represent both documents and queries as formulae of the language of the adopted logic. The relationship between a document represented by a formula d and a query represented by a formula q is expressed by the formula $d \rightarrow q$, where $\rightarrow$ is the implication connective. The estimate of the relevance of d with respect to q consists in determining the "logical status" of the implication $d \rightarrow q$; to this aim the notion of validity has been mostly applied: d is relevant to q if $d \rightarrow q$ is valid (we recall that a formula is valid if it is true for all the possible interpretations) [11, 17]. Logical consequentiality and truth are other logical notions considered to this aim. To overcome some limitations encountered in adopting classical logic, non-classical logics have been considered and employed to model IR, such as modal logic, logical imaging, and terminological logic [5, 12, 15]. A good review of the logical models of IR can be found in [11, 17].

In [14] the author analyses the role of logic in IR by defining a model of information retrieval based on modal logics, which provides a general framework to define pre-existing IR models. This model is based on the logical interpretation of information retrieval introduced by van Rijsbergen [20], by which the implication of a query by a document is estimated with a given degree of certainty or strenght: $P(d \to q)$. An important point of this approach is that the implication is not the material implication of classical logic. To evaluate the strentgth of the correspondence between a document and a query, Nie considers both the implications $d \to q$ and $q \to d$, the first one being related with the concept of "exhaustivity" of the document to the query, the second one being related to the concept of "specificity" of the document to the query.

In this contribution we analyse the role of logic in IR by following another kind of approach: starting by the boolean model, we formulate its query evaluation by means of first order logic, and we formulate the query evaluation of extended boolean models by fuzzy logic. In our approach the symbol $\to$ denotes the material implication, and it is applied to relate a query term with a term in the document representations: $q(t) \to d(t)$, in which $q(t)$ denotes term t in query q and $d(t)$ denotes term t in document d. The correspondence between a query term and a document is estimated as the degree of truth of the implication under the interpretation consituted by the given query and the considered document. The direction of the implication naturally derives from the modeling of the concept of relevance in the boolean model, in which a document is relevant to a query term if its representation contains the term.

In this paper, then, we do not try to define a new logical model of IR: our aim is to propose a logical formulation of the Boolean model and of weighted Boolean models, in order to outline their connections with logic, and to better understand some query weight semantics proposed in the literature. The proposed approach is based on three main considerations:

1. in an IRS the atomic components of information items are terms; the estimate of relevance of a document to a query is usually decomposed in the estimate of relevance (or probability of relevance, in the probabilistic approaches) of a document to each query term. Terms constitute in fact the elementary constraints in a query, which, by aggregation, are combined to specify a more complex constraint (the overall query expression);
2. the Boolean model is based on classical set theory, which is strictly connected with logic, thus giving raise to a natural logical interpretation;
3. the concept of relevance can be modelled as a gradual concept; it can thus be related to the concept of truth, intended as the conformity of a proposition with respect to a given interpretation.

In the proposed approach, we employ the formal language of logic to represent formulae expressing the *evaluation structure* of a Boolean query; the basic elements of such an expression are the constraints imposed by query terms and aggregated through logical connectives. A document and a query determine a given interpretation, with respect to which the truth value of the *evaluation* formula represents the degree of relevance of the document with respect to the considered query.

After presenting a formulation of the Boolean model in terms of first order logic, we extend our analysis to weighted Boolean models (in which both numeric index term weights and numeric query term weights are employed to discriminate the importance of terms as descriptors) and show that our logical interpretation is naturally extended to formalize weighted Boolean models in the context of fuzzy logic. This extension fully expresses the graduality of the concept of relevance of documents to user queries.

It is important to notice that with our logical formulation only the terms appearing in a query are considered to the aim of query evaluation, while in the validity-based models all possible combinations of terms have to be considered.

The paper is organized as follows: in section 2 we introduce a logical interpretation of the Boolean model. In section 3 we briefly describe the main extensions of the Boolean model by means of fuzzy set theory, and in section 4. we formalize a logical interpretation of these extension as a generalization, in fuzzy logic, of the logic interpretation of the Boolean model.

2 A logical formulation of the Boolean model

In order to express the evaluation structure of a Boolean query in a logical framework, we start by analysing the constraint imposed by a query term, which constitute the basic component of a Boolean query expression. We recall that a Boolean query is an expression in which a set of terms (elementary constraints) are combined by the operators AND, OR and NOT, to formulate a more complex constraint that a document must satisfy to be estimated relevant to the considered query. The concept of relevance in the Boolean model is binary, in the sense that a document is either relevant or not to a considered query. Our logical formulation exploits the bottom-up evaluation of a Boolean query, by means of which first the constraints imposed by query terms are evaluated, and then their satisfaction degrees are combined through the Boolean connectives.

The constraint imposed by a query term t is the requirement of the presence of t as a descriptor in the desired document representations; a document is estimated relevant to t if its representation ”includes” $t : t \in d$. A query constituted by a single query term t can then be seen as the specification of an ”ideal” class of documents, those with $F_d(t) = 1$, in which F_d is the characteristic function of the set constituting the representation of document d. On the terms not specified in the query no constraints are imposed: these terms may or may not belong to the representation of documents, thus not affecting the relevance estimate of the documents with respect to t. The two rows in Table 1 represent two documents; with respect to a query constituted by a term t_1 both documents d_1 and d_2 are judged relevant even if the first is concerned with t_2 while the second is not.

Doc	t_1	t_2
d_1	1	0
d_2	1	.1

Table 1

From a logical point of view the presence of a term in a query implies its membership value in the document representation: $F_{TQ}(t) \rightarrow F_d(t)$, in which F_{TQ} is the characteristic function of the query term-set $TQ = \{t\}$. The truth value of this implication for query terms (terms with $F_{TQ}(t) = 1$) is 1 only in the case in which the term belongs to the document representation ($F_d(t) = 1$). In the case of a negated term, the truth value of the expression $\neg(F_Q(t) \rightarrow F_d(t))$ has to be evaluated as the relevance estimate of document d; this means that if a document satisfies the constraint imposed by t it is not relevant, while if it does not satisfies the constraint it is relevant. In fact, to negate a constraint means to exclude the documents which satisfy that constraint.

An important point to outline is that, by this interpretation, the degree of relevance corresponds to the truth value of the implication explicitating the constraint imposed by a query term t, under the interpretation corresponding to a considered document plus the considered query.

After this first analysis of the logical interpretation of the constraint imposed by a query term, which is the core of our proposal, we present the logical formulation of the Boolean model. We adopt first order logic for the reason that it allows to better structure the information involved in the formalization [4, 18].

We represent the evaluation structure of a Boolean query by a formula of the formal language, which preserves the structure of the Boolean query. The basic components of this expression are the implications which evaluate query terms; the AND, OR, NOT operators correspond to the logical connectives $\wedge$, $\vee$, and $\neg$ respectively. The universe of discourse is constituted by the index terms plus the documents plus the queries.

We use different symbols of variable to designate the three distinct classes of objects of the universe of discourse: d for documents, q for queries and t for terms. We also use the constant symbols $t_1, t_2, ..., t_n$ to designate specific terms.

We introduce the two following predicates: QT is a binary predicate symbol (which stands for query term): $QT(t, q)$ identifies the terms appearing in a given query q; IT is a binary predicate symbol (which stands for index term): $IT(t, d)$ identifies the index terms belonging to the representation of document d.

The core of our logical interpretation of a Boolean query is the formal expression of the constraint imposed by a query term: $QT(t, q) \rightarrow IT(t, d)$. As explained before, this implication has the meaning that a document d can be considered relevant to a query term t if t is also an index term of d. For example, the expression representing the query evaluation structure of the Boolean query $q = (t_1$ AND t_2 OR (NOT t_3)) is the following:

$$(QT(t_1, q) \rightarrow IT(t_1, d) \wedge QT(t_2, q) \rightarrow IT(t2, d)) \vee (\neg QT(t_3, q) \rightarrow IT(t_3, d))$$

in which t_1, t_2 and t_3 are constant symbols identifying query terms.

An interpretation of this formal system corresponds to a real document together with a considered query; for a given interpretation I the variable symbol d is associated with the document and the variable symbol q is associated with the query; the truth value of the implication $QT(t, q) \rightarrow IT(t, d)$ represents the degree of relevance (RSV) of the document d with respect to the constraint imposed by term t, i.e. the degree of satisfaction of the constraint expressed by a query term (in the Boolean model a constraint is fully satisfied or it is not).

This semantic-based interpretation of the relevance concept through logic supports a consideration of the relevance as a matter of degree; as we will see in the next section the graduality of the relevance concept will be naturally modelled by considering the fuzzy extension of this logical formulation.

In Table 2 an example of query evaluation is presented; we consider the query $q = (t_1$ AND t_2 OR (NOT $t_3))$ and the three documents d_1, d_2 and d_3. To simplify the notation we indicate $QT(t_i, q)$ as qt_i and $IN(t_i, d)$ simply as t_i. Each row of the table corresponds to a document; this is possible because of the closed-world assumption of the Boolean model (if a term it is not an element of the document representation it is not concerned with the document).

I	t_1	t_2	t_3	qt_1	qt_2	qt_3	$qt_1 \rightarrow t_1$	$qt_2 \rightarrow t_1$	$qt_3 \rightarrow t_1$	q
d_1	1	1	0	1	1	1	1	1	0	1
d_2	0	1	1	1	1	1	0	1	1	0
d_3	1	0	0	1	1	1	1	0	0	1

Table 2

The terms listed in the columns are the terms appearing in the considered query q, which are the only useful to the aim of query evaluation.

3 Weighted Boolean retrieval models

In the literature, some extensions of the Boolean model have been defined through fuzzy set theory, mainly to the aim of modeling the graduality of the concept of documents relevance. The elements of a fuzzy set are characterised by means of the concept of "graduality" of memebership; with respect to a classical set, a fuzzy set supports a more informative (accurate) description of a class of elements when a sharp boundary of membership cannot be naturally devised [23]. In this section we briefly describe the extensions of the Boolean model necessary to understand how fuzzy (multi-valued) implications can be employed to generalize the logical interpretation of the Boolean model presented in the previous section.

The fuzzy extensions of the Boolean model analysed in this section concern both the representation of documents and the query language. For a more general introduction to the application of fuzzy set theory to IR see [3, 10].

We assume that query evaluation is defined in two subsequent steps. In the first step the constraint imposed by a single term is evaluated by function $E : D \times Q' \rightarrow [0,1]$ in which Q' is the set of queries composed by a single term. Function E computes the degree to which a document d matches a query $q \in Q'$. In the second step, a function E^* is defined: $E^* : D \times Q \rightarrow [0,1]$ (where Q is the set of all the legitimate queries) which evaluates the final RSV of a document, expressing the satisfaction of the whole query. The Boolean operators AND, OR and NOT are interpreted as fuzzy intersection (usually min), union (usually max) and complement respectively.

3.1 A weighted model based on the fuzzy document representation

The first, simplest extension of the Boolean model concerns the document representation; a weighted indexing function is adopted [19] and a document is represented as a fuzzy set of terms: $R(d) = \sum_{t \in T} \mu_d(t)/t$ [10]. For each index-term of a document a numeric membership degree $\mu_d(t) \in [0,1]$ is specified, which expresses the level of concern (aboutness or significance) of the term with respect to the information contained in the document. Usually, the weighting function is defined on the basis of the number of occurrences of t in the document d and in the whole archive D [19]. More recently, other definitions of this function have been proposed, which take into account the document structure, by differently weighting a term occurrence depending on the document subpart in which it appears. To this aim two weghting models have been proposed, one which refers to a logical structure of documents [1], the other which refers to the syntactic structure of documents written in HTML [13]. Like in the vector space model and in probabilistic models, the adoption of a weighted representation makes the retrieval mechanism able to rank the retrieved documents in decreasing order of their relevance with respect to the user query. By this very simple extension the retrieval function evaluating a query term t yields the index term weight: $E(d,q) = F(d,t)$ with $q = t \in Q'$, which is interpreted as the degree of relevance of document d with respect to query term t.

3.2 Extended Boolean Models with weighted query languages

In the Boolean query language the basic selection criteria are terms, which specify constraints on the document representation. To enrich the expressiveness of the Boolean query language, numeric query weights have been introduced as an extension of the basic selection criteria, which become then pairs term-weight. An example of weighted Boolean query is the following: $< t_1, w_1 >$ AND ($< t_2, w_2 >$ OR $< t_3, w_3 >$) in which t_1, t_2, and t_3, are search terms, and w_1, w_2, $w_3 \in [0,1]$ are numeric weights.

A weighted term specifies a constraint on the weighted document representation, which depends on the semantics of the query term weight. In the literature different semantics for query weights have been proposed: the weight can be interpreted as a relative importance indicator, as a threshold on the index term weights, or the specification of an ideal index term weight [3, 10]. The weight semantics determines the definition of function $E : D \times Q' \rightarrow [0, 1]$, in which $Q' = T \times [0, 1]$. The value $E(d, < t, w >)$ is the degree of satisfaction of the selection criterion $< t, w >$ by document d.

In the following, we describe the relative and the threshold semantics for query weights, as they are involved in our logical formulation. The former defines query weights as measures of the "relative importance" of each term with respect to the others in the query [3, 10]. The weights demand for retrieval results conditioned more heavily by the most heavily weighted terms. It is important to notice that this semantics needs an evaluation function which is context dependent, in the sense that two different definitions are given depending on the fact that two weigted terms are connected by the AND or by the OR aggregation operator. In fact, a great query weight for an ANDed term should produce a low evaluation degree able to dominate the min function; on the contrary, a great query weight for an ORed term should produce a high evaluation degree able to dominate the max function. An example of formalization of this semantics is given by the following E function: for conjunctive queries the E function is defined as: $E(F(d, t), w) = max((1 - w), F(d, t))$. In case of disjunctive queries it is defined as: $E(F(d, t), w) = min(w, F(d, t))$ [21]. As this semantics requires a context-dependent evaluation of a query, it could not be modelled by our logical formalization, which requires an evaluation of the constraint imposed by terms independently on the query aggregation structure. However, as any Boolean query can be expressed in Disjunctive Normal Form, we can overcome this problem by constructing a logical formula in DNF, thus employing the unique evaluation function associated with the AND connective, as we will see in the next section.

Other authors proposed a threshold semantics for query weights [10]. By specifying a thresholded term the user is asking to see all documents sufficiently about a topic. The lower the threshold, the greater the number of documents retrieved. The simplest formalization of threshold weights defines the E function as: $E(F(d, t), w) = F(d, t)$ if $F(d, t) \geq w$, 0 otherwise [16].

In this case, the threshold defines the minimally acceptable document. It is important to notice that this modelization of the concept of threshold is such that the documents with an index term weight under the threshold are completely penalized, while those over the threshold have a satisfaction increasing as the index term weight approaches the threshold. Of course other intepretations have been and could be formalized [3, 10].

4 Fuzzy logic to model weighted Boolean models

4.1 Some concepts of fuzzy logic

Fuzzy logic is a logic of vagueness, which has been formalized to the aim of dealing with vague knowledge and supporting approximate reasoning [6, 7, 9, 22]. The concept of vague predicate is a central one in fuzzy logic. In classical logic a unary predicate identifies a subset of the universe of discourse; analogously, in fuzzy logic a vague predicate identifies a fuzzy subset of the universe of discourse. For example, the unary predicate $young(x)$ identifies the fuzzy subset specifying the concept *young* over the possible numeric values for age. In a given interpretation which assigns a numeric value to the variable x, the truth value of $young(x)$ is the degree of membership of x to the fuzzy set *young* ($\mu_{young}(x)$). Then, in the case in which the information in a considered interpretation is precise the truth value of a vague predicate is a number in [0,1]; in the case of vague information the truth value is a fuzzy-truth value which may be approximated by means of two values, a necessity and a possiblity degree [6, 7]. In fuzzy logic the connectives $\vee, \wedge$ and the negation $\neg$ are usually defined as follows $a \vee b = min(a, b)$, $a \wedge b = max(a, b)$ and $\neg a = 1 - a$, in which $a, b \in [0, 1]$.

The fuzzy (multi-valued) implication operator constitues an extension of the implication operator of classical logic, in the sense that its behaviour (under a given interpretation) for the values 0 and 1 reduces to the behaviour of the implication operator in classical logic [8, 9]. It is a binary operator (fuzzy relation) defined on $[0, 1] \times [0, 1]$ and taking values on [0,1]. Several definitions of the fuzzy implication operator have been given in the literature, among which the following ones:

$a \rightarrow_{RG} b = 1$ if $a \leq b$, 0 otherwise
$a \rightarrow_{Gd} b = 1$ if $a \leq b$, b otherwise
$a \rightarrow_{Gg} b = 1$ if $a \leq b$, b/a otherwise
$a \rightarrow_{L} b = 1$ if $a \leq b$, $1 + b - a$ otherwise
$a \rightarrow_{D} b = max(1 - a, b)$

in which *RG, Gd, Gg, RG,* and *D* stand for Rescher-Gaines, Gődel, Goguen, Lukasiewicz, and Dienes respectively.

For some definitions of the implication operator an interpretation of the interaction between the two arguments (antecedent and consequent) has been outlined. By a first interpretation the value a is seen as a threshold which has to be reached by the value b: if the threshold is reached the implication operator produces the value 1. If the threshold is not reached, a "penalty" is applied. Among the implication operators having this behaviour, we recall the Rescher-Gaines, Gődel, Goguen and Lukasiewicz implications; these implications, in the case in which the threshold is not reached produce the values: 0, b, b/a and $(1 - a + b)$ respectively.

A second interpretation is connected to the Dienes implication; the behaviour of this implication is that the lower the a value, the higher the value of the implication (and the higher the a value, the closer to b is the value of the implication). If we consider a formula with implications connected by the AND operator, the a value can be considered as an importance weight for the b value. In fact as 1 is the neutral element for the *min* operator, the implications with a low a value do not have a great influence on the aggregation result.

As we will see later, the distinct behaviour the considered implication operators makes their choice very important in the logical formulation of the weighted Boolean models [2]. This choice is in fact related with the modelled semantics of query weights.

4.2 A logical formulation of weighted Boolean models

In this section we extend the logical interpretation of the Boolean model to weighted Boolean models. We use the same symbols of variables and constant, but we introduce the vague predicates IMP and QW to identify the importances of the index terms in document and in query representations respectively. The vague predicate IMP characterizes the importance of the index terms in a document d, while the vague predicate QW characterizes the weight of the terms in a considered query q.
A query of the type $q = < t_1, w_1 >$ AND $< t_2, w_2 >$ OR $< t_3, w_3 >$, has an evaluation which is formally expressed by the following logical expression:

$$((QW(t_1, q) \rightarrow IMP(t_1, d)) \wedge (QW(t_2, q) \rightarrow IMP(t_2, d))) \vee (QW(t_3, q) \rightarrow IMP(t_3, d)).$$

This formalization reflects the process of "composition" (formulation) of a query: first the index terms which are the useful descriptors are selected and combined through aggregation operators, then weights are associated with each term: if we decompose the query formulation process in this way, with a query expression different weights assignments can be selected; each one plus a document corresponds to an interpretation. In the considered interpretation the weighted query is associated with the formula in which the components are the constant symbols corresponding to terms and the logical connectives correspond to the Boolean connectives.

A document d and the query q constitute an interpetation for this expression, and the truth values of the predicates QW and IMP are the query term weights and the index term weights of the considered interpretation respectively.

The important point to outline is that the choice of the implication operator is crucial in this formalization, as it is strictly connected with the semantics of the query term-weight. In fact, as we have previously seen, a particular choice of the implication operator realizes the explicitation of a particular constraint imposed by the antecedent on the consequent.

The two main semantics associated with the antecedent of a fuzzy implication are those of importance and threshold. Let us first reason about the importance semantics; to this aim we assume that a query be expressed in DNF. If we express the term evaluation constraint by adopting the Kleene-Dienes implication, we obtain the E function presented in section 3.2: in the case of conjunctive queries: $w \rightarrow_{KD} F(d,t) = max(1-w, F(d,t)) = E(d, < t, w >)$.

When associating a threshold weight with a query term a user is introducing a minimum acceptance level of the $F(d,t)$ values. The association of a value w with a query term implies then a classification of the $F(d,t)$ values in the documents to be analyzed: the values under and over w. As we have seen in the previous section, this behaviour is modelled by distinct implication operators, such as the Rescher-Gaines, Gődel, Goguen and Lukasiewicz implications.
In the case in which the E function modeling a threshold semantics for query weights is defined by a fuzzy implication, a complete satisfaction is obtained when $F(d,t)$ is over (or equal to) w. Some threshold semantics for query weights proposed in the literature have a different behaviour for $F(d,t)$ values over w: the higher the F(d,t) the higher the RSV (i.e. $E(d, < t, w >)$). The maximum RSV is obtained when $F(d,t) = 1$. For $F(d,t)$ under w, the $E(d, < t, w >))$ value is penalized, as in the case of the implication operators. These functions express a stricter constraint than that imposed by a fuzzy implication. It follows that these threshold functions are not directly expressible as fuzzy implications; in [2] it has been shown that these functions can be modelled as "modified" fuzzy implications.

Considering Radecki's formalization of threshold semantics, the following expression is obtained: $E(d, < t, w >) = (w \rightarrow_{RG} F(d,t)) * F(d,t)$ where $\rightarrow_{RG}$ is the Rescher-Gaines implication. Function E is defined by modifying a "threshold-based" implication, so as to obtain the highest satisfaction only when $F(d,t) = 1$.

5 Conclusions

In this paper a formulation has been proposed of the Boolean model and of some weighted Boolean models in terms of first order logic and fuzzy logic respectively. The core of this approach is to make explicit the bottom-up structure of the query evaluation process; to this aim the implication connective is employed to express the constraint imposed by a query term on the document representations. A logical formula corresponds to a query evaluation structure, and the degree of relevance of a document to a user query is obtained as the truth value of the formula expressing the evaluation structure of the considered query under the interpretation corresponding with a document and the query itself.

References

1. G. Bordogna and G. Pasi. Controlling Information Retrieval through a user adaptive representation of documents. *International Journal of Approximate Reasoning*, 12 (1995) 317-339.

2. G. Bordogna, P. Bosc, and G. Pasi. Extended Booelan Information Retrieval in terms of Fuzzy Inclusion. *Knowledge Management in Fuzzy Databases* O. Pons, M. A. Vila and J. Kacprzyk eds. Physica Verlag,Heidelberg, in press, 1999.
3. G. Bordogna, P. Carrara, and G. Pasi. Fuzzy approaches to extend Boolean Information retrieval. *Fuzziness in Database Management Systems* P. Bosc, and J. Kacprzyk eds., Series Studies in Fuzziness, Physica-Verlag. Berlin, Germany, (1995) 231–274.
4. C. L. Chang, R. C. Lee. *Symbolic logic and mechanical theorem proving.* Academic Press, New York, 1973.
5. F. Crestani, and C. J. van Rijsbergen. Information retrieval by logical imaging, *Journal of Documentation*, 51(1) (1995) 293–331.
6. D. Dubois and H. Prade. An introduction to possibilistic and fuzzy logics, in *Non-standard logics for automated reasoning*, P. Smets, A. Mamdani, D. Dubois, H. Prade eds., Academic Press, 1988, 287–326.
7. D. Dubois and R. Prade. Fuzzy logics and the generalized modus ponens revisited, *Cybernetics and Systems*, 15 (1984) 293–331.
8. J. Fodor. On fuzzy implications operators, *Fuzzy Sets and Systems*, 42 (1991) 293–300.
9. G. J. Klir, B. Yuan. *Fuzzy Sets and Fuzzy Logic.* Prentice Hall, 1995.
10. D. Kraft, G. Bordogna, G. Pasi. Fuzzy Information Retrieval, in *International Handbook of Fuzzy Sets*, D. Dubois and H. Prade eds., Kluwer Academic Publishers, Vol.3, chapter 6, in press, 1999.
11. M. Lalmas. Logical models in Information Retrieval: introduction and overview, *Information Processing and Management*, 34 (1998) 19–33.
12. C. Meghini, F. Sebastiani, U. Straccia, C. Thanos. A model of information retrieval based on terminological logic, in Proc. *ACM Sigir Conference on Research and Development in Information Retrieval* Pittsburgh, U.S.A., (1995) 298–307.
13. A. Molinari and G. Pasi. A fuzzy representation of HTML documents for information retrieval systems, in Proc. *IEEE International Conference on Fuzzy Systems* New Orleans, U.S.A., 8-12 September 1996, 107–112.
14. J. Y. Nie. An outline of a general model for information retrieval systems, in Proc. *SIGIR 1988* Grenoble, France, June 1988, 495–506.
15. J. Y. Nie. An information retrieval model based on modal logic, *Information Processing and Management* 25(5) (1989) 477–491.
16. T. Radecki. Fuzzy set theoretical approach to document retrieval, *Information Processing and Management* 15(5) (1979) 247–260.
17. F. Sebastiani. On the role of logic in Information Retrieval, *Information Processing and Management*, 34 (1998) 1–18.
18. J. R. Shoenfield. *Mathematical Logic.* Reading : Addison Wesley, 1967.
19. G. Salton, and M. J. McGill. *Introduction to modern information retrieval.* Reading : Addison Wesley, 1983.
20. C. J. Van Rijsergen. A non-classical logic for information retrieval, *Computer Journal*, 29(6) (1986).
21. R. R. Yager. A note on weighted queries in information retrieval systems, *Journal of the American Society for Information Science*, 38(1) (1987) 23–24.
22. L. A. Zadeh. Fuzzy Logic, *IEEE Computer*, April 1988, 82–93.
23. L. A. Zadeh. Fuzzy Sets, *Information and Control*, 8 (1965) 338–353.

A Geometric View of Relevance Effectiveness in Information Retrieval

Sándor Dominich

Department of Computer Science and Information Technology, Buckinghamshire Chilterns University College, High Wycombe, HP11 2JZ, Buckinghamshire, UK, E-mail: sdomin01@buckscol.ac.uk; Department of Computer Science, University of Veszprem, Egyetem u. 10, 8200 Veszprem, Hungary, E-mail: dominich@dcs.vein.hu[1]

Abstract. Relevance is a central concept in Information Retrieval (IR). It is used to work out effectiveness measures for IR systems, i.e. measures to express how well (or bad) an IR system performs; classical measures are precision, recall, fallout. It is shown that the empirical relation $P=NR/x$ (P=precision, R=recall, N=total number of relevant documents, x=the number of retrieved documents) can be formally easily obtained. It is also shown that using the concept of fallout a typical surface can be constructed with the noteworthy properties that it looks similarly for every IR system and each point on this surface corresponds to a 3-tuple (precision, recall, fallout) and thus to one retrieval process. Thus, the name of effectiveness surface is suggested for it. The performance of an IR system can be enhanced by a technique called relevance feedback (used to return documents that are likely to be more relevant). A sequence of repeatedly applied relevance feedbacks, being a sequence of repeated retrievals, corresponds to a sequence of points ('walk') on the effectiveness surface. It is shown that this sequence can be theoretically modelled by an important mathematical structure (recursively enumerable set or Diophantine set), and that it yields a point on the effectiveness surface corresponding to an optimal retrieval situation. Further, the existence of an optimal point is also shown and it is computed as well.

Keywords: Information Retrieval, Relevance Theory, Constrained Nonlinear Optimisation.

1. Introduction

Relevance is a central concept in Information Retrieval (IR), and is mainly a subjective category expressing the user's judgment as to how relevant (or irrelevant) a returned document is. A technique called *relevance feedback* can be used to return documents that are likely to be more relevant: the user is presented with a list of documents first and asked to rank them according to his/her relevance judgment; this is used then to return other documents that are likely to better satisfy his/her information need. Because of the central role played by relevance, it has been and is used to work out *effectiveness measures* for IR systems, i.e. measures to express how well (or bad) an IR system performs. The complexity of this task as well as the compoudness of relevance is also well reflected in recent years research by e.g. Mizzaro (1997), Allan (1996), Belkin (1996), Belkin & Koenemann (1996), JASIS (1994).

Precision and *recall* are two traditional effectiveness measures: precision means the proportion of relevant documents out of those returned, whereas recall that of returned documents out of the relevant ones. Buckland & Gey (1994) attempt to elaborate polynomials

[1] Research partly supported by research grant Pro Cultura Renovanda Tud 98/31

for precision and recall using postulates based on empirical considerations. Their result is that $P = N_{rel} R / x$, where P means precision, R means recall, N_{rel} is the total number of relevant documents and x is the number of retrieved documents.

In the present paper it is shown that this relationship can be easily obtained in a formal (theoretical) way, too. Thus, this may be used as a theoretical introduction to Buckland and Gey's cited work. Moreover, using a third traditional concept for effectiveness, *fallout* (which means the proportion of returned documents out of those nonrelevant), it is shown that a surface can be constructed with the following properties:

a) It looks similarly for every IR system (only its specific position, not its shape, in space varies depending on the total number of documents and the total number of relevant documents given a query).

b) Each point on this surface corresponds to a 3-tuple (precision, recall, fallout) and thus to one retrieval process.

A sequence of repeatedly applied relevance feedbacks corresponds to a sequence of points on this surface. Assuming that the sequence of relevance feedbacks (e.g. in a probabilistic retrieval) improves the effectiveness of the IR system, i.e. it improves precision, recall and fallout, the question of whether the corresponding sequence of points tends towards a specific point (corresponding to an optimum) arises, and if so which is that point? In other words, the sequence of retrievals means a 'walk' on the surface. In mathematical terms, the above question means the following: is there an optimal (minimal) point that this sequence of points leads to? The answer to this question will be yes: it will be shown that a repeatedly applied sequence of relevance feedbacks in probabilistic retrieval can be conceived as a recursive process that can be theoretically modelled by an important mathematical structure (recursively enumerable set or Diophantine set). A noteworthy property of such a structure is that it has a fixed point. This point can be thought of as corresponding to a retrieval situation which cannot be improved anymore, hence the existence of an optimal point is granted and it can be computed. This computation will also be performed.

The paper suggests that this surface be called an *effectiveness surface* which thus offers a geometric view of relevance effectiveness of an IR system. At the same time, it also reflects a theoretical dynamics among precision, recall and fallout during repeated relevance feedbacks.

2. Definitions of the main mathematical concepts used

1. *Recursion* means to define a process (function, procedure, language) with reference to itself. Formally, recursion is a process whereby a function f, called a primitive recursive function, with n+1 variables is defined as follows: $f(x_1, x_2, ..., x_n, 0) = \alpha(x_1, x_2, ..., x_n)$, $f(x_1, x_2, ..., x_n, y+1) = \beta(x_1, x_2, ..., x_n, y, f(x_1, x_2, ..., x_n, y))$ where α is a function with n variables and β a function with n+2 variables. In words, function f is given an initial value first represented by function α, and then every next value of f is defined by function β using the previous value of function f. An example for recursion is the usual arithmetical addition: $f(x, 0) = x$, $f(x, y+1) = f(x, y) + 1$. Let us consider the particular case when n=1. Then $f(x, 0) = \alpha(x)$, $f(x, y+1) = \beta(x, y, f(x, y))$, where $x = x_1$ and hence the index can be omitted. This particular case is used in Theorem 1.

2. *Fixed point.* Given a recursion with a corresponding f (as above). Then there exists a situation when the value of f coincides with the value on which f is computed; symbolically $f(x) = x$. This value is referred to as a fixed point. There are several fixed points theorems in recursion theory (e.g. Rogers Fixed Point, First Recursion and Second Recursion Theorems). For this paper, it is important that a fixed point exists, and this corresponds to a situation where a retrieved set of documents is the same as that retrieved in the next step (after relevance feedback). This may, in principle, be interpreted as a case when effectiveness cannot be enhanced anymore. This concept and interpretation of a fixed point is used in part 4.

3. *Diophantine set* (or recursively enumerable set, abbreviation; r.e. or Diophantine structure). A subset $A \subseteq B$ is called a recursively enumerable or Diphantine set relative to B if there exists a procedure (process, algorithm, program, language) that, when presented with an input $b \in B$, outputs 'yes' if and only if $b \in A$. When $b \notin A$ then the procedure does not end (undefined). For example, the set of C programs that halt on a given input is a r.e. set (relative to the set of all C programs). It can be shown that a set whose elements are given (generated) by a primitive recursive function is a Diophantine set. This result is used in Part 4.

4. *Level surface.* Given a function f: $\mathbf{R}^3 \rightarrow \mathbf{R}$, $f(x, y, z) \in \mathbf{R}$, $(x, y, z) \in \mathbf{R}^3$, and a constant c

$\in$ **R**. The set of points (x, y, z) in space for which $f(x, y, z) = c$ is called a *level surface*. For example, let $f(x, y, z) = x^2 + y^2 + z^2$ and $c = 4$. Then the level surface $x^2 + y^2 + z^2 = 4$ is the sphere having its centre in the origin and radius equal to 2. All level surfaces in this example are spheres but with different radii. This result is used in Part 3.

3. Effectiveness surface

Let $\Delta \neq 0$ denote the total number of relevant documents to a query q, $|\Re(q)| = \kappa \neq 0$ denote the number of retrieved documents in response to q, and α denote the number of retrieved and relevant documents. It is reasonable to assume that $|D| = M > \Delta$.

The meaning of the usual relevance effectiveness measures are recalled now as follows.

DEFINITION 1. *Recall* ρ is defined as $\rho = \alpha / \Delta$.
DEFINITION 2. *Precision* π is defined as $\pi = \alpha / \kappa$.

One can easily see that:

PROPOSITION 1. *The ratio of recall and precision varies linearly with* κ.

(Proof. $\alpha = \rho\Delta = \pi\kappa \Rightarrow \rho / \pi = \kappa / \Delta$.)

The result reported by Buckland and Gey (1994) is $P = N_{rel} R / x$. This rewrites as $R / P = x / N_{rel}$ which coincides with that of Proposition 1.

A third traditional measure is as follows:

DEFINITION 3. *Fallout* φ is defined as $\varphi = (\kappa - \alpha) / (M - \Delta)$.

As a noteworthy property, it can be shown now that recall, precision and fallout satisfy the following relation:

PROPOSITION 2.

$$\frac{\varphi\pi}{\rho(1-\pi)} = \frac{\Delta}{M-\Delta}$$

Proof.

$$\varphi = \frac{\kappa-\alpha}{M-\Delta} = \frac{\kappa-\rho\Delta}{M-\Delta} = \frac{\rho\Delta/\pi - \rho\Delta}{M-\Delta} = \frac{\rho\Delta(1-\pi)}{\pi(M-\Delta)}$$

Because recall ρ, fallout φ and precision π can take on different values e.g. in a series of relevance feedbacks for the same query q, they can be thought of as values of variables in general as follows: x for fallout, y for precision and z for recall. Thus the left hand side of the expression in Proposition 2 corresponds to a three-variables function $f(x, y, z) = xy / (z(1 - y))$. The right hand side of the above expression is constant for a query q under consideration. Thus, one can consider all those values of function f, i.e. points in the three dimensional Euclidean space, that are equal to the right hand side. This is equivalent to defining a surface as follows.

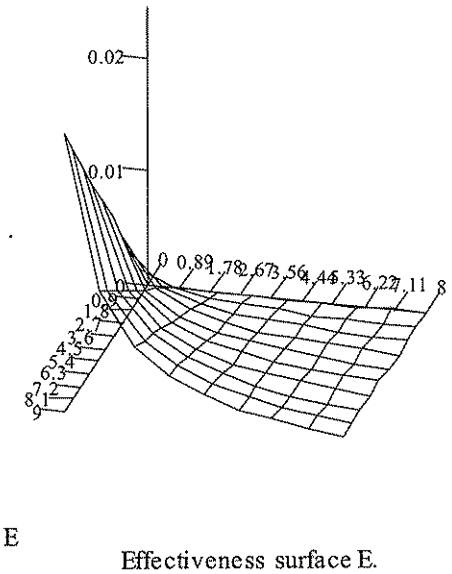
DEFINITION 4. The level surface Σ

$$\Sigma = \{ (x, y, z) \in \mathbf{R}^3 : \frac{xy}{z(1-y)} = \frac{\Delta}{M - \Delta} \}$$

is called the *effectiveness surface* of *IR* (corresponding to q).

The figure below shows a plot of the effectiveness surface.

The vertical axis corresponds to fallout, the axis to its right to precision, and the third axis corresponds to recall. It can nicely be seen that fallout increases when precision is low and recall is high. The lower on the surface, the higher precision and/or recall. The shape of the effectiveness surface is the same for every IR system hence it is a typical surface. Its specific position in space varies with the total number of documents in the IR system and the total number of relevant documents given a query. The surface remains the same for the same query, and repeated relevance feedbacks for the same query means a walk on this surface (which plays the role of a constraint in the optimisation of effectiveness).



Effectiveness surface E.

4. Relevance feedback as a Diophantine structure

Let q denote a query. In Probabilistic *IR* (*PIR*), the set $\Re(q)$ of retrieved documents $d \in D$

in response to query q contains those documents whose conditional probability of relevance $P(R|(q, d))$ exceeds that of nonrelevance $P(I|(q, d))$ – Bayes' Decision Rule – and a real valued threshold (or cut off) $\tau \geq 0$, i.e. $\Re(q) = \{ d \in D: P(R|(q, d)) \geq P(I|(q, d)), P(R|(q, d)) \geq \tau \}$.

A basis to estimate the probabilities $P(.|(q, d))$ is offered by Bayes' Formula:

$$P(\bullet|(q, d)) = \frac{P((q, d)|\bullet)\, P(\bullet)}{P(q, d)}$$

where the symbol • stands for relevance R or irrelevance I.

Bayes' Formula requires that for the estimation of $P(\bullet|(q, d))$ an initial set $\Re_0(q)$ be known first, based on which $P((q, d)|\bullet)$ can be estimated. The denominator $P(q, d)$ simplifies, $P(\bullet)$ is the a priori probability of criterion . in general and is constant. The estimation of $P(\bullet|(q, d))$ can be iterated using each time the previous $\Re(q)$ to re-estimate (relevance feedback) the probabilities $P((q, d)|\bullet)$.

We show now the following connection between *PIR* and a Diophantine structure.

THEOREM 1. *Repeatedly applying PIR yields a Diophantine structure.*

Proof. Given a query q. An initial set $\Re_0(q)$ of retrieved documents is obtained first. *PIR* is repeatedly applied in consecutive steps $s = 1, 2, ...$ At any step s, the set $\Re_{s-1}(q)$ of the previous step is used to estimate the probabilities $P((q, d)|\bullet)$ based on which the probabilities $P(\bullet|(q, d))$ can be calculated – using Bayes' Formula – and a new set $\Re_s(q)$ of retrieved documents is obtained. Let $f(x, y)$ mean the newly retrieved set of documents $\Re_s(q)$ at step s, where x is an integer variable corresponding to query q and y is an integer variable symbolising step s when probabilities $P(\bullet|(q, d))$ are computed. Let the process of calculating, based on relevance feedback, the new probabilities $P(\bullet|(q, d))$ and of retrieving a new set $\Re_{s+1}(q)$ of documents, at step s+1, be represented by a function $\beta(x, y, f(x, y))$. One can consider a series $\Re_0(q)$, $\Re_1(q)$, $\Re_2(q)$, ..., $\Re_s(q)$, ... of retrieved documents. Thus, using the above introduced functions, one can define function f recursively as follows: $f(x, 0) = \alpha(x)$ and $f(x, y + 1) = \beta(x, y, f(x, y))$ with the following meaning: at the initial step $s = 0$, i.e. $f(x, 0)$, an initial set $\Re_0(q)$ is retrieved (using e.g. a vector or interaction or another method), represented by $\alpha(x)$; then, at every next step $s + 1$, a new set $\mathfrak{R}_{s+1}(q)$ is obtained, i.e. $f(x, y + 1)$, after repeatedly computing, based on relevance feedback using the previous $\mathfrak{R}_s(q)$, the probabilities $P(\bullet|(q, d))$ and perfoming a retrieval opeartion again, i.e. $\beta(x, y, f(x, y))$. Because, formally, function f is recursively defined (primitive recursive function), the series $\mathfrak{R}_0(q)$, $\mathfrak{R}_1(q)$, $\mathfrak{R}_2(q)$, ..., $\mathfrak{R}_s(q)$, ... forms a recursively enumerable (r.e.) set (relative to the power set $\wp(D)$ where D denotes the set of dcouments to be searched), and, as such, it is a Diophantine set.

The function f, being recursive, is computable, hence it has a fixed point (Rogers Fixed Point Theorem, Philips, 1992). This means that there exists an index s such that the same $\mathfrak{R}_s$ is obtained in a next step as that of the previous one, i.e. there is not any improvement and thus this can be inerpreted as an optimal situation. If one assigns now to the sets $\mathfrak{R}_s$ points on a the effectiveness surface, a fixed point corresponds to an (local/global) optimum (minimum, maximum) on this surface.

5. Optimal Information Retrieval

Ideally, an *IR* should be such that $\varphi = 0$ and $\rho^{\cdot} = 1$. Because $\varphi = 0$ implies $\pi = 1$, the following two optimalities can be defined.

DEFINITION 6. An *IR* is called ρ-*optimal* if $\rho = 1$, and φ-*optimal* if $\varphi = 0$.

It is easy to see that in a ρ-optimal *IR* we have $\pi < 1$, whereas in a φ-optimal *IR* $\rho < 1$. Effectiveness (or performance) is now defined as a relative position, e.g. Euclidean distance or a cosine, to the ideal $I^* = (\varphi^*, \rho^*, \pi^*) = (0, 1, 1)$ as follows.

DEFINITION 7. *Effectiveness* (*performance*) ε of an *IR* is defined as the Euclidean distance between a current point (φ, ρ, π) and the ideal $I^* = (0, 1, 1)$:

$$\varepsilon = (\varphi^2 + (1 - \pi)^2 + (1 - \rho)^2)^{1/2}$$

The smaller ε the more effective *IR*. It is easy to see that $\varepsilon = 1 - \rho$ for φ-optimality, and $\varepsilon = (\varphi^2 + (1 - \pi)^2\ )^{1/2}$ for ρ-optimality. The concept of a global optimality is now defined as follows (its existence follows from Theorem 1):

DEFINITION 8. An *optimal IR* is one with φ, π and ρ such that $\min_{\varphi,\pi,\rho\in\Sigma} \varepsilon$.

In other words, an *IR* is optimal if its recall, precision and fallout are a solution of the following nonlinear minimization problem with constraints:

$$\min\ (\varphi^2 + (1-\pi)^2 + (1-\rho)^2)^{1/2}$$

subject to the following constraints:

$$\frac{\varphi\pi}{\rho(1-\pi)} = \frac{\Delta}{M-\Delta} \qquad 0 < \rho \leq 1 \qquad 0 \leq \pi \leq 1$$

Alternatively, instead of the Euclidean distance, other measure for ε may also be used, e.g. the cosine of the angle between the ideal $\boldsymbol{I}^* = (0, 1, 1)$ and the current vector $\mathbf{o} = (\varphi, \pi, \rho)$:

$$\varepsilon = \frac{(\boldsymbol{I}^*, \mathbf{o})}{\|\boldsymbol{I}^*\|\ \|\mathbf{o}\|} = \frac{\pi + \rho}{2\cdot(\varphi^2 + \pi^2 + \rho^2)^{1/2}}$$

In this case, the higher ε, the more effective *IR* (the smaller the angle). The concept of a global optimality in this case is defined as follows. An *optimal IR* is one with φ, π and ρ such that $\max_{\varphi,\pi,\rho\in\Sigma} \varepsilon$.

In other words, an *IR* is optimal if its recall, precision and fallout are a solution of the following nonlinear maximization problem with constraints:

$$\max \frac{\pi + \rho}{2\cdot(\varphi^2 + \pi^2 + \rho^2)^{1/2}}$$

subject to the following constraints:

$$\frac{\varphi\pi}{\rho(1-\pi)} = \frac{\Delta}{M-\Delta} \qquad 0 < \rho \leq 1 \qquad 0 \leq \pi \leq 1$$

Solutions for both problems are as follows (using MathCAD 8.01 Plus Professional). Both methods give, practically, the same global optimum.

a) **Euclidean effectiveness**:

$\varepsilon(\phi,\rho,\Pi) := \sqrt{\phi^2 + \left(1-\Pi^2\right) + \left(1-\rho^2\right)}$ Let $\Delta := 500$ $M := 500000$

Guess values (near to global optimum): $\phi := 0.1$ $\Pi := 0.9$ $\rho := 0.9$

Minimization with constraints:

Given

$$\frac{\phi\cdot\Pi}{\rho\cdot(1-\Pi)} = \frac{\Delta}{M-\Delta} \qquad \rho>0 \qquad \rho\le 1 \qquad \Pi\ge 0 \qquad \Pi\le 1$$

$$\text{Minimize}(\varepsilon,\phi,\rho,\Pi) = \begin{bmatrix} 1.275\cdot 10^{-7} \\ 1 \\ 1 \end{bmatrix}$$

b) **Cosine angle effectiveness:**

$$\varepsilon(\phi,\rho,\Pi) := \frac{\Pi+\rho}{2\cdot\sqrt{\phi^2+\Pi^2+\rho^2}}$$

Maximization with constraints:

Given

$$\frac{\phi\cdot\Pi}{\rho\cdot(1-\Pi)} = \frac{\Delta}{M-\Delta} \qquad \rho>0 \qquad \rho\le 1 \qquad \Pi\ge 0 \qquad \Pi\le 1$$

$$\text{Maximize}(\varepsilon,\phi,\rho,\Pi) = \begin{bmatrix} 7.625\cdot 10^{-7} \\ 1 \\ 0.999 \end{bmatrix}$$

6. Conclusion

It was shown that a typical surface can be constructed on which each point corresponds to a three-tuple (recall, precision, fallout). The name effectiveness surface was suggested for it. A series of repeatedly applied relevance feedbacks corresponds to a sequence of points on this surface. It was shown that the corresponding series of sets of retrieved documents forms a Diophantine set which, as a recursive structure, has a fixed point. This point may be interpreted as being an optimal point for optimal effectiveness values. These values can be calculated, they being the solutions of a constrained nonlinear optimisation mathematical problem. In other words, relevance effectiveness enhancement is formulated as a constrained nonlinear optimisation problem controlled by an effectiveness surface. Thus, relevance feedback, as a means to enhance effectiveness, is equivalent to a mathematical process of minimizing (maximizing) a nonlinear function subject to nonlinear constraints.

The paper also shows a connection between IR and Recursion Theory as an abstract mathematical structure.

References

Allan, J. (1996) Incremental Relevance Feedback. In SIGIR '96 *Proceedings of the 19th International Conference on Research and Development in Information Retrieval.* Zurich, Switzerland, 270-278.

Belkin, N.J. et al. (1996) Using relevance feedback and ranking in interactive searching. In Harman, D. (ed.) *TREC-4 Proceedings of Fourth Text Retrieval Conference*. Washington, D.C., 181-209.

Belkin, N.J. & Koenemann, J. (1996) A case for interaction: A study of interac-tive information retrieval behavior and effectiveness. In *Proceddings of the ACM SIG CHI Conference on Human Factors in Computing Systems*, New York, 205-212.

Buckland, M. & Gey, F. (1994) The Relationship Between Recall and Precision. *Journal of the American Society for Information Science*, **45(1)**: 12-19.

JASIS (1994) Special topic issue: Relevance research. *Journal of the American Society for Information Science*, 45.

Mizzaro, S. (1997) Relevance: The Whole History. *Journal of the American Society of Information Science*. 48, 810-832.

Philips, I.C.C. (1992) Recursion Theory. In: Abramsky, S. & Gabbay, D.M. & Maibaum, T.S.E. (eds.) *Handbook of Logic in Computer Science*. Vol. 1, Oxford Science Publications, Clarenden Press.

Text Passage Classification Using Supervised Learning

Y. Bi [1], F. Murtagh [2], S. McClean [1] and T.Anderson [1]

(1) Faculty of Informatics, University of Ulster
Shore Road, Newtownabbey, Co. Antrim
BT37 0QB, Northern Ireland
Email: Y.Bi@ulst.ac.uk

(2) Department of Computer Science, The Queen's University,
Belfast BT7 1NN, Northern Ireland, UK
Email: F.Murtagh@Queens-Belfast.AC.UK

Abstract. In this paper, we describe a method for text passage classification or extraction by means of supervised machine learning and analytically identifying passages. The underlying characteristic of the method lies in the utilization of the resulting classification, which leads to the classification of the portion of a document in a high dimensional feature space into a low dimensional space which is composed of the features drawn from the document itself. We also present a preliminary experiment for evaluating the performance of this method.

1. Introduction

The concept of passage is treated, in information retrieval, as a self-contained portion of the document, in which a clear topic should be contained, and there is a topic shift between passages [2, 10, 16, 21, 25]. We adopt this concept in this paper. With respect to text passage classification, it is the problem of associating the passage (segment) with a topic drawn from a larger text document into one or more predefined categories. More formally, given a set of ***n*** categories and a document, the task is to split the document into smaller segments and map them into the corresponding categories. In this paper, we presume that the case described is one category assignment.

Compared to passage retrieval, although passage classification is slightly different from passage retrieval since the latter aims at retrieving passages in response to an arbitrary query, rather than the predefined categories, the similarities between classification and retrieval can be seen via the probability ranking principle. In information retrieval, for the document case, the probability ranking principle means that the optimal way to present documents to the user is to rank them by decreasing order of $P(R_Q \mid D)$, the probability of relevance to the query given a document D. The document classification problem is to decide whether D belongs to a category C_1 or its complement C_2 (two category case). The optimal way to decide whether a document belongs to a category is given by Bayes decision rule: decide C_1 if $P(C_1 \mid D) > P(C_2 \mid D)$, otherwise decide C_2. Therefore, the primary goal of both document retrieval and classification is to compute $P(C \mid D)$. Here, the case of passage retrieval and classification is similar to the document case.

Text passage classification can be viewed as a special case of text document classification. Alternatively it also can be viewed as a further step in document classification. The underlying distinction between both classifications lies in the fact that the passage constitutes a basic unit to be classified into the predefined categories instead of the document being considered as a unit. There are several apparent advantages when individual text passages are independently classified into known categories. First, the efficiency of text utilization may be improved because the user is no longer faced with large masses of documents covering a diversity of topics, but instead can concentrate on the most immediately relevant passage categories [18, 21]. Second, the known categories may make the pieces of text understandable, recognizable, manageable and capable of being reassembled. Consequently, besides the standard applications to information retrieval and extraction, passage classification may be effectively applied in establishing links between pairs of the passages within documents, in particular, hyperlinks on the Web; automatic text summarization [19]; and automatically tagging text body in coding XML (eXtensible Markup Language) documents.

In this paper, we propose an approach for passage classification, that is to take the Naïve Bayes learning approach as the first step to classify documents via ranking the posterior probabilities of features (referring to words or terms) of the documents, and then refine the classified documents to identify passages which are closer to the known categories than others using the likelihood of the features. The basic insight supporting this approach is that the words related to the same topic are closer to each other [9]. This approach effectively combines inductive learning in a high dimensional feature space with the analytical deduction in a relatively low dimensional feature space. It can guarantee that the objects from the training examples are consistent with the objects to be classified, and techniques developed to work for document classification will work, with suitable modifications, for the task of passage classification.

2. Approach to passage classification

The approaches to carrying out passage classification can broadly be divided into two categories. One can follow the way of standard document classification, treating each passage as a mini-document, and then classifying the passage by using learning methods, as in document classification. In contrast with document classification, where a training set consists of total documents, for passage classification, the training set should be composed of complete passages derived from documents, ensuring that the object to be classified is consistent with the sample from the training data set. Note that this training set requires that individual passages of each category have to be manually identified from the document examples. Such hand-crafted work would be expensive to achieve. Even given such a training set, the basic characteristics of each passage are self-contained and lack the corresponding global context. In addition, for this training set, the computational costs will increase with increasing number of passages. For example, given *100* categories, and a document comprising *100* passages, the number of comparisons of categories and passages increases from *100* to 100^2 compared with the document classification.

However, with respect to the issues of the passage training data and the global context above, it could be argued that they might appear to be not important when a text document is represented using the traditional vector model – the "bag of words" which ignores the document structures and word ordering in each document [6, 11]. One might expect that a training set of domain-specific documents could be generalized for both classification cases – an entire document and the portions of the document. Unfortunately such generalization can not be supported from the results of information retrieval [25], and it remains unclear how to use such an approach for passage classification.

The second strategy, which is adopted in this work, is to take the approach used for document classification as the first step, classifying a document into one of a set of predefined categories, and then to identify the passage which is the closest to this category via analyzing the likelihood of the features of the document. As opposed to standard document classification, this approach employs features which are as informative as possible, regardless of the optimum of feature selection, in order to retain the integrity of the document and the context in which the passage resides. The motivation for this approach comes from the complementary advantages of combining statistically inductive and deductive methods. On the one hand, the inductive method offers the advantage that it requires no explicit knowledge and learns a function or rule for generalization cases based solely on the training data. This results in a given document being classified into one predefined category. On the other hand, the identified category can be then used as explicit and specific knowledge for uncovering a passage which the local vicinity of the features concentrates more on this category than others, without taking account of the training data. This method leads to an approach to using some results derived from a text classifier for further exploring passage recognition.

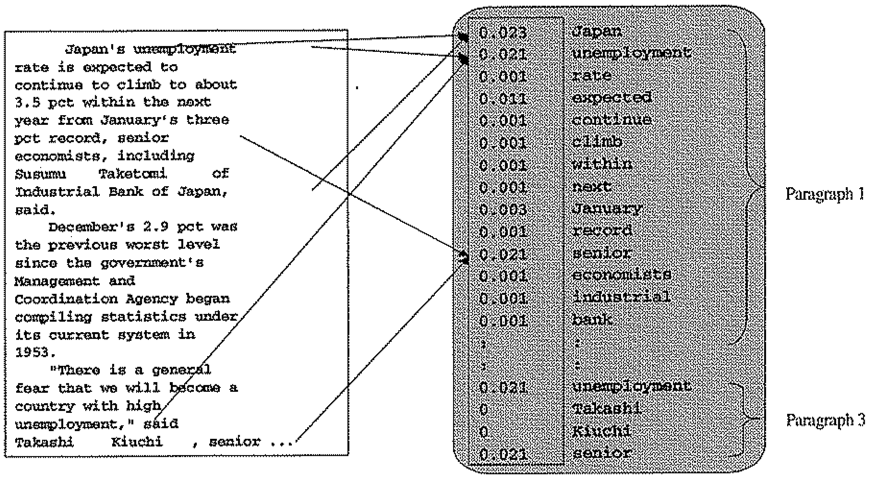
An apparent disadvantage for this method, however, is that the deduction can be misled when given an incorrect category, e.g. knowledge. Given such an uncertainty of statistical classification, we take a principle used in [11] that minimizing the number of errors of the statistical classification guarantees that the accuracy of classification is highest, and we ignore the case of no category assignment.

In inductive methods for text document classification, feature selection (reduction) has received considerable attention for improving the effectiveness and efficiency of performance of learning algorithms. The argument in favor of feature reduction is that the learning model can operate with the relatively small size of input features, and non-informative features can be removed. Thus reduced dimensionality of features makes the learning algorithm have relatively little computational cost, and the resulting features are interpretable. More recently, Yang and Pedersen [27] compare a number of methods for feature reduction, and argue that feature reduction has the potential to exceed the performance in text classification before the features are reduced.

Yet, there are strong practical reasons to expect that more features should be retained before carrying out document classification. A number of the experiments on feature reduction show that the optimum number of features for document classification varies [8]. It is likely that many of the best informative features are eliminated via a "semi-unified threshold". Also, a term which may appear to be a very non-informative or poor feature on its own may turn out to be a good discriminative feature in combination with other terms [24]. Subsequently, it is unavoidable that feature reduction may destroy the integrity of documents – the context of a topic.

3. Passage representation

A common method for inductive text classification is to use partial or limited features of documents to uncover the entire characteristics of documents, e.g. measure the associations between documents and categories being via the limited words, particularly in the "bag of words" model. Some empirical evidence shows the amount of features can be eliminated up to 90% or more of the unique terms with either an improvement or no loss in classification (as measured by average precision) [27]. However, such methods can not effectively be applied in passage classification since they may not be concerned with the distribution of these features within documents. Also, there is little reason to believe that the occurrence of one word or two words in a paragraph can reflect the entire characteristics of the paragraphs (moderate size). Consequently, using the standard vector model of text representation, it is hard to employ information inherent in the document structure such as the distribution of terms. In order to rank passages of the documents, we require a text representation that has the capacity of representing more features, and facilitates use of ordering and location of features. Figure 1 shows the simple text representation adopted from [17].



Japan's unemployment rate is expected to continue to climb to about 3.5 pct within the next year from January's three pct record, senior economists, including Susumu Taketomi of Industrial Bank of Japan, said.

December's 2.9 pct was the previous worst level since the government's Management and Coordination Agency began compiling statistics under its current system in 1953.

"There is a general fear that we will become a country with high unemployment," said Takashi Kiuchi , senior ...

0.023	Japan
0.021	unemployment
0.001	rate
0.011	expected
0.001	continue
0.001	climb
0.001	within
0.001	next
0.003	January
0.001	record
0.021	senior
0.001	economists
0.001	industrial
0.001	bank
:	:
:	:
0.021	unemployment
0	Takashi
0	Kiuchi
0.021	senior

Figure 1: A representation of the document

4. Passage generation

Methods have been designed and implemented in the past for the use of text passages in information retrieval. These methods aim either at using the number and concentration of query words included in the individual sentences and paragraphs to generate the score of the portions of documents, or infer boundaries by shift of topic. Salton [21] suggested that a right step to passage retrieval may be based on the use of text paragraphs rather than sentences for the construction of text passages. Kaszkiel et al. [10] proposed a passage extraction mechanism allowing any sequence of words of any length starting at any word in the document. Obviously this method is impractical. Furthermore, the any length is restricted to a set of fixed passage length – from 50 to 600 words in increments of 50. The experimental results show that compared to whole-document ranking, the effectiveness of passage retrieval can be improved by 21% via ranking of arbitrary passages with fixed-length.

To yield a set of candidate passages, the following approach takes two aspects of the fixed length of words and paragraphs into account. For the first passage, starting at the beginning of a document, the first 100 words are considered as a candidate one. For the second passage, starting at an increment in 30 words of the document, a segment with 100 words is taken as the second one, and so on. Correspondingly, the feature vector of each passage is also formed following the same procedure, and viewed as a linear concatenation of three segments where the each of these consists of 30 features, each segment approximating one paragraph in the document (see Figure 2). The association between an identified category and the passage will be ranked without considering the number of paragraphs the passage spans, and then the passage with highest-rank will be examined as to how many paragraphs are included in this passage. As a result, the length of this passage may vary. Note that there is no theoretical basis for optimal choice for the above values of 100 and 30. However, various length of passages will be further investigated. Due to the requirement of making the computation for a larger size of features operational, we choose a simple learning algorithm – the naïve Bayes as a document classifier.

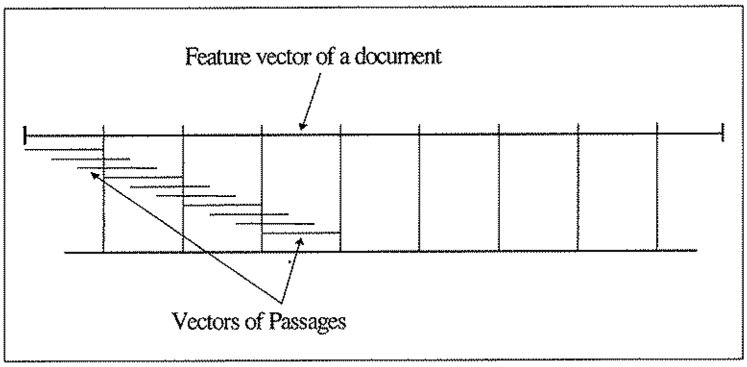


Figure 2: Passage generation

5. Naïve Bayes classifier

In machine learning, we are often interested in determining the best hypothesis *h* from a hypothesis space *H*, given observed training data *D*, and this can be generalized to unseen cases. In general, we suppose *H* consists of a set of possible hypotheses with prior knowledge: h_1, …, h_r. Instead of finding the best hypotheses, we compute the probability for all the hypotheses. These probabilities would reflect how much we trust (or like) each of the hypotheses. Bayes theorem provides a direct method for computing the probability of a hypothesis based on its prior knowledge (probability), the probability of observing various data, given the hypothesis and the training data itself [17].

Adapting this theorem to text classification, a number of text classification systems have been built on this probabilistic framework [13, 14, 17]. The basic idea is to use the conditional (posteriori) probabilities of categories given a word to estimate the probabilities of categories given a document. More precisely, given a set of predefined categories (C_1, …, C_q), and a set of documents (D_1, …, D_r), where each document can be represented by a feature vector $\vec{x} = (x_1, \ldots, x_k)$, in which x_i is term and the length *k* is not fixed. The conditional probabilities can be then computed via the following formula:

$$P(C = C_j \mid \vec{x}) = \frac{P(\vec{x} \mid C = C_j) * P(C = C_j)}{P(\vec{x})} \qquad (1)$$

In order to make the computation of $P(\vec{x} \mid C = C_i)$ (often called the likelihood of $\vec{x}$ given C_i) practical, a common strategy is to assume that each feature drawn from the features x_1, .., x_k is conditionally independent of every other feature on a given C_i. In addition, division by $P(\vec{x})$ is in practice unnecessary since it is a constant for each C_i and has no effect on the classification. Thus Equation 1 is simplified as follows:

$$P(C = C_j \mid \vec{x}) = P(C = C_j) \cdot \prod_{i \leq k} P(x_i \mid C = C_j) \qquad (2)$$

$P(C = C_j)$ can be estimated from the fraction of documents in the corresponding categories, and $P(x_i \mid C = C_j)$ can be computed based on the *m*-estimate or the Laplace law [17]. Thus, the estimate for $P(x_i \mid C = C_j)$ will be given as:

$$P(x_i \mid C = C_j) = \frac{n_{ij}}{n_j + |\,\text{Vocabulary}\,|} \qquad (3)$$

where n_j is the total number of terms in the category C_j, n_{ij} is the number of occurrences of term x_i in category C_j, and |Vocabulary| is the total number of distinct terms found within the training examples.

Given a feature vector of a new document $\vec{d}$ = (d_1, …, d_k), the aim is to seek a maximum posterior probability of $\vec{d}$ in all categories C_i. For any d_i in this vector, its value will be zero if the corresponding feature does not occur in the training set.

6. Identifying passages

As stated above, the occurrences of distinctive words in documents usually are not conditionally independent. Huge number of cases can prove that such an assumption of conditional independence is not true in practice. Although the independence assumption almost never holds for textual data, the performance of its application in text classification has been shown to be comparable to other learning algorithms such as decision tree [13]. Thus, we retain this independence assumption in identifying passages, ensuring the consistency of the utilization of feature values between the document classification and identifying passages.

Once a new document is categorized, the task is to identify a part of the document which is more closely associated with the topic of the corresponding category. Formally, let a new document $\vec{d} = (d_1,...,d_k)$, the categorical assignments of the document is category $C_1^{'}$, we aim at seeking a sub-vector $\vec{d}_i = (d_{i1},...,d_{ig})$ representing a passage, where not only $\vec{d}_i \subseteq \vec{d}$ and the ordering of d_{ij} coincide with the original ordering in the document, but also the association between $\vec{d}_i$ and $C_1^{'}$ is closer than others.

Given the above setting, our interest here lies in seeking such a method that can make best use of an obtained fact – knowledge that the association between $\vec{d}$ and $C_1^{'}$ is probabilistically biggest in the collection of the document. This fact simplifies the problem of identifying a passage from a high dimensional D to a relatively small feature space D' which consists of the features of a document. Thus instead of measuring the association between $\vec{d}_i$ and $C_1^{'}$, we rank the passages based on the number and concentration of features for representing passage content. A straightforward implementation is to rank the sum of feature values (likelihood see Equation 2) to identify a $\vec{d}_i = (d_{i1}, ..., d_{ig})$, where the biggest sum of the likelihood of the passage $\vec{d}_i$ is defined as the one with the closest association between $\vec{d}_i$ and C_1'. The justification of this intuition here stems from the well-known cluster hypothesis in information retrieval [20]. We adapt this notion for identifying a passage, e.g. a dense part, in this small feature space D'.

In order to find $\vec{d}_i$ using a maximum likelihood estimate, we need to maintain as many informative features over the training data as possible, ensuring in each paragraph that relatively adequate features exist. Given a set of candidate passages and the methods of passage ranking, let us look at what situation may lead to a passage with the highest ranking. Figure 3(a) depicts an ideal situation in achieving the highest ranking. One might conjecture such a situation is one where most of the features in the passage consist of words with middle frequency and concentrate on one region composed of less paragraphs. This notion could be explained by the empirical evidence described by van Rijsbergen [23] that 'middle frequency terms are most useful in retrieval particularly if the distribution of terms is skewed', that is because such terms are informative for text classification tasks [27]. However, the same result – the highest ranking – also can be achieved if terms with middle frequency scatter over more paragraphs. This situation is shown in Figure 3(b). The last situation is which the passage is made of a mixture of low and high frequency terms. In this extreme case, the terms may span less or more paragraphs with the bigger standard deviation. If nothing special is known about the distribution of the features under consideration, we have to use a mechanism to judge the variances of the three situations mentioned above in ranking passages.

$$Score_{pi} = \frac{1}{N} \mu_{pi} (1 - \sigma_{pi}) \qquad (4)$$

Equation 4 is proposed for calculating the score of ranking passages. In this equation, μ_{pi} is a mean of feature likelihood of Passage $_i$, $N = |Passage_i|$ is number of paragraphs that the Passage $_i$ spans, σ_{pi} is the standard deviation of feature likelihood of the Passage $_i$. N and σ_{pi} are two coefficients which are used to control the impact of the latter two situations stated above on the score. That is, if the number of paragraphs that the passage spans is more, and the standard deviation of feature likelihood of the passage is larger, then it is hard to achieve a biggest score for this passage. Therefore, this equation should express the three situations considered above to some extent.

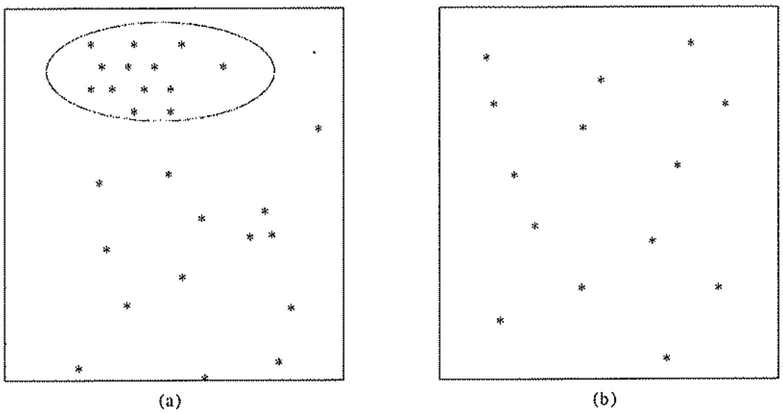


Figure 3: Two examples of feature distribution within the document ((a) there is a dense region, (b) there is not)

7. A preliminary experiment

In this section, we describe a preliminary experiment for evaluating the performance of the above method. For our experiments, we aim at evaluating the accuracy of identifying the passage, rather than the performance of document classification. That is because we somewhat directly adapt the Naïve Bayes Classifier described in [17] into the document classification in this work with little variation on text representation. The evaluation of performance of the variations of this learning algorithm has been described in [4, 13, 17]. In this context, however, it would be apparent that the identifying passage can fail or be misled when the incorrect text category knowledge is given, and this method is suspect or powerless if the knowledge is incorrect or unavailable. Therefore, identifying a passage is based on such a simple assumption that a given document at least can be classified into one predefined category.

Evaluating the accuracy of identifying a passage is problematic, since established evaluation methods, such as precision-recall, may not be appropriate to this task. To get an idea of the effectiveness of our identifying passage method, we designed the following experiments, which are mainly concerned with determining the boundaries of the passages. We use the Equation (3) to rank a set of candidate passages. The passage with the biggest score is defined as one which is closest to the identified category, and it also is expected that there would be variations between the passage and the preceding and succeeding passages.

Since we did not have an appropriate training dataset to evaluate the method, we used the new version of Reuters to randomly construct 3 test documents. We used 214 stories of the Reuters with three categories, Consumer Price Index (CPI), Unemployment (JOBS) and Gross National/Domestic Product (GNP), as a training dataset which consists of 3110 distinctive words (features) after removing the stop words and high frequent words such as "PCT". Of these, 2200 words are from CPI category, 1800 words come from JOBS category and 2660 from GNP. There are 16,039 unique terms in the collection [24, 27]. To approximate a practical situation in computing probabilities of features, we presume the training dataset contains 5000 unique terms. Note that what is reported here is mainly based on the GNP category.

The constructed documents are a simple concatenation of stories from different categories without considering story boundaries. The length of each document is over 500. The first document is a concatenation of two GNP and Stock stories, which is split into a set of 17 passages; and the second one is composed of two JOBS and GNP stories which include 14 passages; and the third document is made up of three Trade, GNP and Wheat stories, which includes 24 passages.

Figure 4 shows a result of this experiment. In this experiment, we did not take the length of paragraphs as a main measuring factor since the paragraphs of most of the stories are only made up of 20 or 30 words before removing stop words. This may not be a representative situation in a larger document. There is also a side-effect in ranking the scores of the passages.

Now, let us see the first document, an identified passage is the Passage 7 with the score of 0.0035. There is an apparent boundary between two stories. This is an ideal case. For the second one, the biggest score is 0.0028 which is associated with Passage 14, and the boundary shift between two stories is not apparent. Interestingly, there is a similar one – Passage 1 with small difference with the biggest one – Passage 14. We notice, however, that there is big change between Passage 1 and Passage 2. This implies that the first thirty terms play a crucial role in obtaining this score. Getting back to the stories, we found there are overlaps among several high frequency terms such as *Japan, bank, rate,* etc. Thus the optimal choice of the passage should be Passage 14.

In the third document, the part of the passages of GNP is from Passage 13 to 18. The expected passage is number 13 and the score is 0.002447. From Figure 4, we can not identify a passage with GNP category so that its score is biggest among the set of the passages. But we could recognize that there were shifts between the preceding and succeeding passages to some extent. With respect to accuracy of identifying the passage for this case, however, the justification should be given from the method and the training data. It should be clear that there were more similarities between two types of stories since many stories are simultaneously labeled with two labels Trade and GNP in Reuters. The strict separation between two types of topics is hard to achieve. On the other hand, this method may be not able to automatically discriminate such cases based solely on statistical measurements. It is important that the extent to which the score of each passage shifts from the preceding and succeeding passages may provide significant evidences for refining passages via incorporating inference. This remains for further investigation.

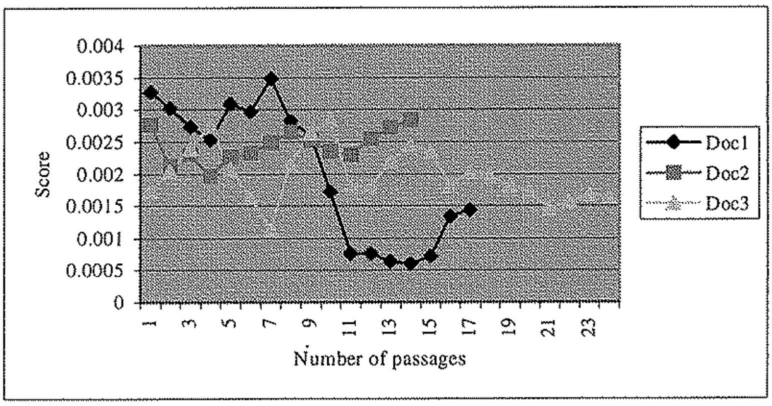


Figure 4: The scores of the passages of the documents

8. Conclusion

In this paper, we provide a possible approach for passage classification using the combination of supervised learning and statistical analysis. In order to explore the structure of a document, in particular, detect a dense part contained in the document, it would be worthwhile investigating in this direction. That is how we transform a passage classification in a high dimension of features into a small itself feature space, in which the results obtained from the classifier can be further used as knowledge to refine the document. In the experiment, due to insufficient training and test data, the experiment of the role of the number of spanning paragraphs has not effectively been carried out. Thus, the expectations which are described (see Section 6) can not be proved. This suggests that extending the scale of the experiment and the classification of various length of passages remain to be investigated.

References

1. C. Apté, F. Damerau, and S. M. Weiss. Towards Language Independent Automated Learning of Text Categorization Models. SIGIR'94. Annual ACM SIGIR Conference on Research and Development in Information Retrieval. P24-30, 1994.
2. J. P. Callan. Passage-Level Evidence in Document Retrieval. SIGIR'94. Annual ACM SIGIR Conference on Research and Development in Information Retrieval. P302-10, 1994.
3. T. G. Dieterich. Machine Learning Research: Four Current Directions. Draft of May 23, 1997. http://www.cs.wisc.edu/~shavlik/cs760.html
4. N. Fuhr. Models for Retrieval with Probabilistic Indexing. Information Processing and Management. 25(1), p55-72, 1989.
5. F. V. Jensen. An Introduction to Bayesian Networks. UCL, 1996.
6. K. S. Jones and P. Willett (editors). Readings in Information Retrieval. Morgan Kaufmann. 1997.
7. K. S. Jones and M. Kay. Linguistics and Information Science. Academic Press. New York. 1973.
8. T. Joachims. Text categorization With Support Vector Machines: Learning With Many Relevant Features. In Proceedings 10th European Conference on Machine Learning (ECML), Springer Verlag, 1998.
9. D. Koller and M. Sahami. Hierarchically Classifying Documents Using Very Few Words. International Conference on Machine Learning - 97. http://Walrus. Stanford.EDU/diglib/pub/reports/
10. M. Kaszkiel and J. Zobel. Passage Retrieval Revisited. SIGIR'97. Annual ACM SIGIR Conference on Research and Development in Information Retrieval. P178-185, 1997.
11. D. D. Lewis. Naive (bayes) at forty: The Independence Assumption in Information Retrieval. In European Conference on Machine Learning, 1998.
12. D. D. Lewis. Reference List to Accompany SIGIR'97 Tutorial on Machine Learning for Information Retrieval. http://www.research.att.com/lewis

13. D. D. Lewis and M. Ringuette. A Comparison of Two Learning Algorithms for Categorization. In Symposium on Document Analysis and Information Retrieval, pages 81-93, Las Vegas, NV, 1994.

14. L. S. Larkey and W. B. Croft. Combining Classifiers in Text Categorization SIGIR'96. Annual ACM SIGIR Conference on Research and Development in Information Retrieval. P289-297, 1996.

15. D. Merkl. Exploration of Text Collections with Hierarchical Feature Maps. SIGIR'97. Annual ACM SIGIR Conference on Research and Development in Information Retrieval. P186-195, 1997.

16. M. Melucct. Passage Retrieval: A Probabilistic Technique. Information Processing & Management. Vol. 34, No. 1, pp43-68, 1998.

17. T. M. Mitchell. Machine Learning. McGraw-Hill, 1997.

18. H. T. Ng, W. B. Goh, and K. L. Low. Feature Selection, Perception Learning, and a Usability Case Study for Text Categorization. SIGIR'97. Annual ACM SIGIR Conference on Research and Development in Information Retrieval. P67-73, 1997.

19. G. Salton, A. Singhal, M. Mitra and C. Buckley. Automatic Texts Structuring and Summarization. Information Processing & Management, Vol.33, No. 2, pp. 193-207, 1997.

20. G. Salton, J. Allan, C. Buckley and A. Singhal. Automatic Analysis, Theme Generation, and Summarization of Machine-Readable Texts. Science. Vol. 264, pp.1421-1426, 3 June 1994.

21. G. Salton, J. Allan and C. Buckley. Approaches to Passage Retrieval in Full Text Information Systems. SIGIR'93. Annual ACM SIGIR Conference on Research and Development in Information Retrieval.

22. H. Schütze and C. Silverstein. Projections for Efficient Document Clustering. SIGIR'97. Annual ACM SIGIR Conference on Research and Development in Information Retrieval. 1997.

23. C. J. van Rijsbergen. Information Retrieval (second edition). Butterworths, 1979.

24. E. Wiener, J. O. Pedersen and A. S. Weigend. A Neural Network Approach to Topic Spotting. Proc. SDAIR '95, pp. 317-332, Las Vegas, NV, 1995.

25. R. Wilkinson. Effective Retrieval of Structured Documents. SIGIR'94. Annual ACM SIGIR Conference on Research and Development in Information Retrieval. P311-17, 1994.

26. Y. Yang. An Evaluation of Statistical Approaches to Text Categorization. Information Retrieval Journal, 1998 (to appear).

27. Y. Yang, J. P. Pedersen. A Comparative Study on Feature Selection in Text Categorization Proceedings of the Fourteenth International Conference on Machine Learning (ICML'97), 1997.

A Logical approach to Query Reformulation motivated from Belief Change

G. Amati[1] and P.D. Bruza[2]

[1] Fondazione Ugo Bordini v. B. Castiglione, 59, Rome, Italy gba@fub.it
[2] Distributed Systems Technology Centre, University of Queensland, Australia. 4072 bruza@dstc.edu.au

Abstract. This article attempts to draw parallels between query reformulation and the theory of belief change. We argue that query expansion and query revision can be explained in terms of the revision of the user's beliefs in query terms. Query contraction is argued as being under-pinned by a process of belief expansion. Although the connection between belief change theory and query reformulation has intuitive appeal, there are problems when adopting the theory as a formalization of the query reformulation process. For example, negation has a different character in query reformulation than it does in belief change theory. The connection between belief change theory and nonmonotonic reasoning is discussed using known results yielding insights into a query reformulation logic.

Keywords: Belief Revision, Nonmonotonic Logics, Information Retrieval.

1 Introduction

Consider a user who wishes to find out about the latest developments in the field of information retrieval. The user types in *information retrieval* and receives a result from the IR system. While viewing this result, the user cannot find any documents which satisfy the information need. As a consequence, the user decides to reformulate the initial query. Having seen that some of the documents in the result set deal with the retrieval of text, the user enters *text retrieval* as the new query.

It is well known that users have difficulty in expressing their information need (Cleverden 1991). Poorly specified queries lead to imprecise query results which necessitates query reformulation in order to improve its precision. In other words, information retrieval is a process which often involves the initial query being reformulated several times until satisfactory precision is achieved in the result set. Note that the query is not reformulated in an arbitrary way. The user is careful in the way in which (s)he modifies the query. We propose that query reformulation can be viewed as a process of belief change. The theory of belief change not only allows us to understand query reformulation in terms of user beliefs about query terms, but also sheds light on the logic behind it. The ultimate goal is to provide intelligent support for the user to aid them in the specification of queries.

2 Query Reformulation and its relation to Belief Change

A recent study of query reformulation on the internet analysed query log-files of a internet meta-search engine and classified the query manipulations during an IR session (Bruza and Dennis 1997). The most frequently occurring manipulations were:

- adding a term to the query (*query expansion*), e.g.,

  windows95 → windows95 help

- replacing a term in the query with another term (*query revision*) , e.g.,

  information retrieval $\dot{\rightarrow}$ text retrieval

– deleting a term from the query (*query contraction*), e.g.,

internet security → internet

Other manipulations included variations of spelling, punctuation, case etc. As such manipulations occurred much less frequently than the above three, we will not consider them here.

We argue that query expansion, revision and contraction are driven by processes involving the expansion, revision and contraction of beliefs about query terms.

2.1 Belief Change

The theory of belief change attempts to explain how a rational agent adjusts and assimilates its beliefs about some real or imagined world (Gärdenfors 1988). A rational agent is assumed to have a partially ordered set K of beliefs. The ordering captures the intuition that the agent holds some beliefs more strongly than others. Beliefs are assumed to be represented by sentences in propositional logic. Due to the dynamics of the agent's environment, an agent may have to *expand*, *revise* or *contract* its beliefs.

Belief expansion refers to the process when a new sentence ϕ is assimilated into the agent's belief system K, together with the logical consequences of the addition (regardless of whether the new belief system is consistent, or not).

Belief revision refers to the process whereby the belief system is adjusted in response to a new sentence that is inconsistent with K. In order to maintain consistency in the resulting belief system, some sentences in K are retracted (deleted).

Belief contraction is a process in which the agent rejects a belief thereby removing it from its belief system. In order for the resulting belief system to be closed under logical consequence, some sentences from K may have been retracted.

The theory of belief change has been heavily studied within the field of artificial intelligence as an underlying theory for agent technology. In addition, the link between belief change and nonmonotonic reasoning has been bridged allowing belief change to be characterized from a logical perspective.

When a user enters a query, (s)he harbours a belief about how useful a term, or set of terms, will be for locating relevant documents. In addition, certain beliefs will be more entrenched than others. In the example above, the initial query *information retrieval* reflects the user most strongly believes that these two terms will lead to relevant documents. Possibly the user is aware of related terms such as *text*, but the belief in these terms is less entrenched and are therefore not used in the initial query. When the user views a query result, (s)he is confronted with new information. This can lead to revision of beliefs, for example, loss of belief in the search term *information* and a corresponding increase in belief of the term *text*.

The user may also hold beliefs about terms that will not be beneficial in satisfying the information need. For example, if the user is interested in *text retrieval*, (s)he may reject terms such as *image*, *multimedia* etc., as they are contained in documents irrelevant to the information need.

In short, we propose that the user holds beliefs about terms that will be beneficial or detrimental to locating documents which could satisfy the information need. These beliefs form a part of the belief system K of the user during an IR session. (Other beliefs arise from the user's knowledge of the retrieval engine, background knowledge etc., but we will not highlight them in this article).

2.2 Formalization of Beliefs

We propose that the beliefs are constructed from a set T of terms. Individual terms will be denoted by the letters $A, B, C \ldots$ etc. Beliefs are drawn from a language $\mathcal{L}_T$ obtained from the power set $\wp(T)$ of T and a new constructor $\neg$ in the following way

$$\begin{array}{ccc} \wp(T) & \subset & \mathcal{L}_T \\ \text{if } \Gamma \in \wp(T) & \text{then} & \neg\Gamma \in \mathcal{L}_T \\ \mathcal{L}_T \text{ is the smallest set} & \text{satysfying} & \text{the above properties} \end{array}$$

For simplicity, $\neg\{A\}$ will be written $\neg A$. This formalism allows us to use only sets of terms and negated sets of terms to express our beliefs and disbeliefs.

The belief set K is founded on a subset of $\mathcal{L}_T$. Greek letters Γ, Δ denote individual beliefs in K. Beliefs are ordered according to $\leq, \leq \subseteq K \times K$ such that

$\Gamma \leq \Delta$ means that Δ *is as least entrenched as* Γ
$\Gamma < \Delta$ means that Δ *is more entrenched than* Γ

The belief system $\langle K, \leq \rangle$ is a poset, the maximal element(s) of which correspond to the current query Q. (For simplicity Q will denote both the current query and the maximal i.e., most deeply entrenched belief(s)). The maximal element Q reflects the intuition that the most deeply entrenched belief corresponds to the query being issued by the user. Although we use sets of terms and the negation restricted to sets of terms, these signed sets can be mapped into formulas of the standard propositional logic language. Hence the entrenchment is actually defined between formulas.

The next most deeply entrenched beliefs after Q correspond to sets of term sets which the user believes are useful for retrieving relevant documents, but not as useful as the terms in Q. For this reason, they were not expressed by the user as a part of the query. These beliefs, denoted $\mathcal{C}$, are, however, likely to be drawn upon for the purposes of query expansion, or query revision.

$$\mathcal{C} = \{\Gamma \in K | \Gamma < Q \wedge \neg\exists_{\Delta \in K}[\Gamma < \Delta \wedge \Delta < Q]\}$$

Finally, there are the sets of terms that the user believes are detrimental towards finding relevant documents. The set of terms rejected by K is the maximal subset R_K of $\mathcal{L}_T$ satisfying the properties $\Delta \in R_K, \neg\Delta \in K$ and $\neg\Delta \in R_K, \Delta \in K$. Hence $K \cap R_K = \emptyset$. These are rejected terms, however, such terms are a part of the belief system in the sense that the user believes such terms to have a negative effect in the quest to locate relevant documents.

A term set Δ which contradicts a belief Γ in K is denoted $\Gamma \perp \Delta$. Placed in a logical perspective, the user believes in the negation of the term. In other words, for all $\Delta \in R_K, \neg\Delta \in K$. Note carefully that negation is dependent on the current state of the be belief system, i.e., it is context sensitive.

Figure 1 depicts a belief set whereby the arrows represent the entrenchment relation, so, for example, {internet,security} > {firewall} > {network}. Q, which corresponds to the most deeply entrenched belief is {internet,security}. The next most deeply entrenched belief set $C =$ {{firewall}, {digital,cipher}}. Examples of the elements of R_K are

{{demographics}, {cyberporn}, {chat,room}}

as the user is interested specifically in the security aspects of the internet, and thus not in its demographics, pornography etc. The figure depicts three levels of beliefs, but in general there could be more.

The expansion, revision and contraction of a query is a concrete reflection of the user's expansion, revision and contraction of beliefs about search terms. We explore this theme in more detail in the following subsections.

2.3 Query Expansion

Query expansion constitutes adding term(s) Γ to query Q. We argue that a process of belief revision underpins query expansion, the new belief system being denoted by $K^*_{Q\cup\Gamma}$. Obviously, the user believes that Γ will improve Q, so Γ would normally be deeply entrenched in K, perhaps drawn from the set $\mathcal{C}$. The term set $\Gamma \cup Q$ (the expanded query) represents the new most deeply entrenched belief. Γ disappears from K as it is promoted to form a part of the new Q. Similarly Q disappears as apparently it was not sufficient. Figure 2 illustrates the state of the belief system after query "internet security" has been expanded into "internet security firewall".

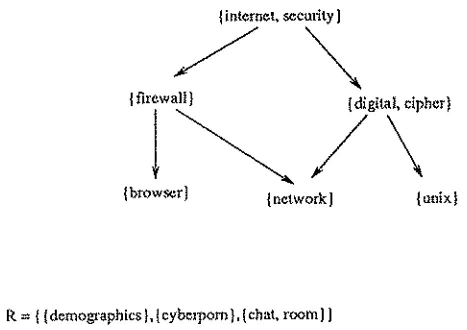


Fig. 1. A belief set K

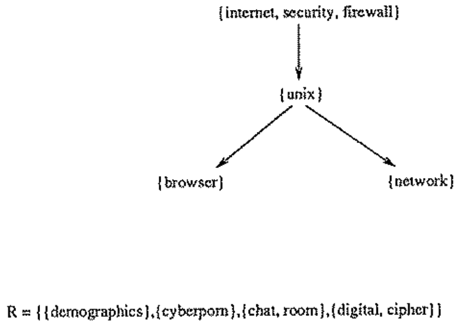

{internet, security, firewall}
{unix}
{browser}
{network}
R = {{demographics},{cyberporn},{chat, room},{digital, cipher}}

Fig. 2. Belief System after query expansion

As the user is interested in firewall security, belief in."unix" is promoted as the user knows that firewalls are often implemented on the unix platform. "Digital ciphers" is rejected as the user is specifically interested in firewalls. It seems reasonable to assume that during successive query expansions the set of rejected beliefs R grows monotonically. This parallels a decrease in the size of the belief system:

$$K^*_{\Gamma \cup Q} \subset K \cup \{\Gamma \cup Q\}$$

The above example illustrates that a belief revision process underpins query expansion. Observe that the expansion of the query has led to the retraction of some beliefs. Let K^-_Γ denote the process of contracting the belief set K by retracting the term (set) Γ. After contraction a new belief set is obtained which has the property:

$$\Gamma \notin K^-_\Gamma$$

In the theory of belief change, contraction relies on negation $\neg$. Indeed, in order to obtain a contraction K^-_ϕ of ϕ from K, different from K itself, we need to decide whether the expansion $K \cup \neg\phi$ is inconsistent or not. This is why, in belief change, contraction is not a primitive process but is definable from expansion and revision by introducing the connective of negation (Gärdenfors 1988). According to belief change theory, the processes of contraction and revision are related by the two relations:

$$K^-_\phi = K \cap K^*_{\neg\phi} \text{ and } K^*_\phi = (K^-_{\neg\phi})^+_\phi$$

In other words belief contraction and belief revision are inter-definable concepts. We can always use one of them as primitive in the theory of belief change.

The mapping of the above equalities into a query reformulation perspective is not straightforward. Unlike logic, negation is a somewhat subtle issue in information retrieval. As suggested earlier, negation manifests itself as a perceived incompatibility between terms when the user is searching. (For example, when searching for documents about *text retrieval*, terms such as *image*, *multimedia* may be viewed by the user as clashing, or being incompatible, with search terms (s)he believes are beneficial for locating relevant documents). There is no easy way to define this sort of incompatibility. It is not a pure syntactic inconsistency as defined in logic. It is the context that gives the truth value to the concept generated by the union of two or more terms. For example "lemon" may generate inconsistency with both "car" and "tree", though "lemon, car", in the sense of a bad bargain, and "lemon,tree" are acceptable. We can't a priori exclude that we are looking for a document talking of "a car crashing into a lemon tree". Hence, we propose a simple but useful semantics for context-dependent negation. Let us consider the collection D of relevant documents.

Definition 1 (Context-dependent negation). *The belief $\Gamma, \Gamma \in K$ contradicts the term $\neg A$, denoted $\Gamma \perp \neg A$ iff for all (or most of) documents $d \in D$, if all elements of Γ are true in d, then A is true.*

We have defined inconsistency via a fuzzy characterization because the lack of occurrence of A given the occurrence of the elements of Γ in a relatively few documents can be due to incomplete description of documents in D (we call this phenomenon the *error frequency of A given Γ*). Note that this definition of negation is context sensitive with respect to the set D. We can't conclude from a single document $d, d \in D$ whether two or more terms are incompatible or not. Negation is indeed a modality of a term, analogous to negation in intuitionistic logic. Observe that due to the error frequency it is not feasible to define $\Gamma \perp A$ along the lines given above. It is an open question how to define this inconsistency in terms of the document space.

The requirement that the documents in D are relevant models the intuition that the incompatibilities are a product of the information need. The user has beliefs about what terms are beneficial to the information need. Many of these terms will be elements of the set Γ. Hence, if $\Gamma \perp \neg A$ we need to retract from Γ in a minimal way those terms such that, for any document d, if all elements of the contraction of Γ are true in d, A is contingent, that is it can occur or not with similar frequency (greater than the error frequency of A and $\neg A$).

The way minimality is formalised is central to IR. Statistical analysis is in general used to solve this problem. For example if we can retract either t_1 or t_2 from Γ in order to retract also A in the above sense, then we may prefer to maintain the more informative term between t_1 and t_2, e.g. that one which has a greater inverse document frequency. For example between $information$ and *text* we could prefer to keep *text* as it is a more informative term.

2.4 Query Contraction

Query contraction is the dual of query expansion. When a user contracts a query "internet security", to "internet", what is the intention behind this contraction? It is clear that the query is too specific, but there are at least two possibilities with regard to the broadening of the scope of the query (Bruza and Van Linder 1998):

1. The user has become interested in all aspects of the internet, *including* security, or,
2. The user has become interested in all aspects of the internet, *excluding* security.

In both cases, query contraction will correspond to an expansion of beliefs. For example, terms which were initially rejected such as "chat room", "pornography" etc. (when the query was "internet security"), may now be accepted because the user is broadening the search to the internet in general (with or without the security aspects). Belief expansion simply involves adding the new belief:

$$K_{\Gamma}^{+} = K \cup \Gamma$$

When contracting the query none of the newly accepted beliefs are maximal, but they may extend the set C.

2.5 Query Revision

There are two aspects of query revision: First, the user selects a term A in the query to be replaced with a term B. Secondly, the user chooses the replacement term B.

A consequence of the first point is that, in the eyes of the user, some query terms are more important than others. Moreover, we cannot assume that this ordering is stable during the IR session. For example, at one instant in the session the user may view term A to be more important than term B, and at another instant in the session, term B may be viewed to be more important than A.

The second point raises a number of complex issues. In practice factors determining the choice of replacement term can be many and varied, for example, the user's general knowledge, the user's knowledge of how search engines function, etc. As a consequence, the relationship between the replacement term and the term being replaced is hard to predict. For example, the replacement term may be more specific, more general, similar, tangential, etc. to the term being replaced.

Due to the above mentioned complexity, it is difficult to formalize query revision precisely. However, the theory of belief change does provide some explanatory power:

- The user rejects term A. This can be modelled as a belief contraction: K_A^-. It is an open question whether all beliefs implied (or strongly related to A) are rejected as well. In the example above, the user rejects "information" in the query "information retrieval" and replaces it with the term "text". It can be argued, "information retrieval" is also rejected as "information" is implied by "information retrieval".
- The replacement term B probably corresponds to a deeply entrenched belief, for example, it is drawn from the set C.
- After rejecting the term A, term B is used to expand the left over part of the query. That is $Q \setminus \{A\}$ is expanded with term B.

In summary, query revision can be viewed as a belief contraction, followed by belief revision. If this view is accepted, the resulting belief system corresponding to revising query Q by replacing term A in Q with term B is formalized by:

$$(K_A^-)^*_{(Q\setminus\{A\})\cup B}$$

2.6 Requirements for a query reformulation logic

By viewing query reformulation in the light of belief revision, the following requirements materialize:

- the logic must be conservatively monotonic with respect to query expansion because the terms chosen for expansion are not arbitrary, but are terms in which the user has some fairly high belief.
- A query reformulation logic should suggest "good" replacement terms during query revision.
- The logic should respect the entrenchment of beliefs etc.

What logic satisfies these requirements? This question will be addressed in the next section.

3 From Belief Change to a Query Reformulation Logic

3.1 Preliminaries

In the following, the symbols A, B, C etc. will be used to denote terms (either from a document, or a query) and Γ, Δ set of terms.

Definition 2. *The logical entailment relation* $\vdash$ *satisfies the following properties:*

$$\Gamma \vdash \Gamma \text{ (Reflexivity)} \tag{1}$$

$$\frac{\Gamma \vdash \Delta}{\Gamma, \Gamma' \vdash \Delta} \text{ (Weakening left)} \tag{2}$$

$$\frac{\Gamma \vdash \Delta \quad \Gamma \vdash \Delta'}{\Gamma \vdash \Delta \cup \Delta'} \text{ (Compositionality)} \tag{3}$$

The notation $\Gamma \mathrel{\mid\!\sim} \Delta$ will be shorthand for "Δ is a relevance entailment of Γ".

Definition 3. *The relevance entailment relation* $\mathrel{\mid\!\sim}$ *satisfies the following properties:*

$$\Gamma \mathrel{\mid\!\sim} \Gamma \text{ (Reflexivity)} \tag{4}$$

$$\frac{\Gamma \mathrel{\mid\!\sim} \Delta \quad \Gamma \mathrel{\mid\!\sim} \Delta'}{\Gamma \mathrel{\mid\!\sim} \Delta \cup \Delta'} \text{ (Compositionality)} \tag{5}$$

Moreover, there are two connecting properties for $\vdash$ *and* $\mathrel{\mid\!\sim}$ *:*

$$\frac{\Gamma \vdash \Delta}{\Gamma \mathrel{\mid\!\sim} \Delta} \text{ (Inclusion)} \tag{6}$$

$$\frac{\Gamma \mathrel{\mid\!\sim} \Delta \quad \Delta \vdash \Gamma}{\Gamma \mathrel{\mid\!\sim} \Lambda \Leftrightarrow \Delta \mathrel{\mid\!\sim} \Lambda} \text{ (Cumulative monotony)} \tag{7}$$

3.2 Correspondence Theory

Gärdenfors and Makinson (1991) showed how to move from the postulates of belief revision to nonmonotonic logic and vice versa. The translation is obtained by the relation

$$C \in K^*_A \Leftrightarrow A \mathrel{\mid\!\dot{\sim}} C \tag{8}$$

Using this interpretation it is possible to translate each postulate $(k^*1) - (K^*8)$ of belief revision (Gärdenfors 1988) into a condition on $\mathrel{\mid\!\sim}$. The main nonmonotonic systems obtained in literature correspond to subsets of $(k^*1) - (K^*8)$.

3.3 Entrenchment Relation

It has been shown that suitable epistemic entrenchment yields belief revision systems and vice versa (Gärdenfors 1988, Gärdenfors and Makinson 1991, Georgatos 1997). We must assume only three basic properties (Georgatos 1997):

$$\Gamma \leq \Gamma \qquad \text{(Reflexivity)} \tag{9}$$

$$\frac{\Gamma \vdash \Delta, \Delta \leq \Lambda}{\Gamma \leq \Lambda} \text{ (Left Monotonicity)} \tag{10}$$

$$\frac{\Gamma \vdash \Delta, \Delta \vdash \Gamma}{\Lambda \leq \Delta \Leftrightarrow \Lambda \leq \Gamma} \text{ (Logical Equivalence)} \tag{11}$$

The main nonmonotonic systems can be obtained through epistemic entrenchment by adding new properties to the three basic ones above. Readers familiar with the axiomatization of nonmonotonic systems, will find it interesting to also know the rational ordering (i.e., the preference relation $\leq$ is also transitive, linear and contains that of the logical entailment $\vdash$) corresponds exactly (i.e., in both directions) to the nonmonotonic system which satisfies Cut, Dominance, Cautious and Rational Monotonicity, Left Logical Equivalence, Right Weakening, And and Or (Georgatos 1996).

Both results show that the problem of constructing revision functions or nonmonotonic systems can be *equivalently reduced* to establish appropriate ordering on the set of formulas. This positions nonmonotonic reasoning in IR: *if we establish a preference or some other ranking on the set of terms then we always generate a nonmonotonic logical deduction relation* (Amati and Georgatos 1996).

3.4 Logical Foundations of Query Expansion

Suppose that the user formulates her query Q which is still incomplete with respect to her informative need. She can explore different expansions Q' with $Q' \vdash Q$. Observe that, by monotonicity which is in turn equivalent to the transitivity property of $\vdash$, the set Δ' of document representations X' satisfying $X' \vdash Q'$ is contained in the set Δ of document representations $X \vdash Q$. Hence, by expanding the query a more restricted number of documents is retrieved, so long as the expansion entails Q. Monotonicity in this sense leads to a nice property whereby the more the user expands the query, the less the set of retrieved documents are obtained. Monotonicity is the property which guarantees this behaviour. Since this is what usually expect in reformulating their queries, our aim is therefore to preserve monotonicity as much as possible in IR. Monotonicity seems only to be supported by the Boolean model. The inner product matching of the vector space model does not. Suppose that the vector space system retrieves documents with weights greater than a fixed threshold τ. By expanding the query it is easy to observe that, when assigning weights to documents, the additivity property of the matching over terms increases previous match values. Therefore, the set of retrieved documents w.r.t. the expanded query *is more, rather than, less numerous* than that of the initial query.

As stated earlier, a requirement on the monotonicity is that it be conservative. The correspondence between belief revision and nonmonotonic reasoning allow the following rules to be put forward as a means of driving query expansion:

$$\frac{\Gamma \mathrel{|\!\sim} \Delta, \Gamma \mathrel{|\!\sim} \Lambda}{\Gamma \cup \Delta \mathrel{|\!\sim} \Lambda} \text{ (Cautious Monotonicity)} \tag{12}$$

$$\frac{\Gamma \mathrel{|\!\sim} \Delta, \Gamma \not\mathrel{|\!\sim} \Lambda}{\Gamma \cup \Lambda \mathrel{|\!\sim} \Delta} \text{ (Rational Monotonicity)} \tag{13}$$

These rules, or variations of them, have been advocated by several authors (Amati and Georgatos 1996, Bruza and Huibers 1996, Bruza and Van Linder 1998, Huibers 1996).

Query revision makes the situation more complex. The user in general starts with a small set of query terms, and then selects one or more of them to be replaced by more specific ones. At the same time these replacement terms are general enough to preserve as much as possible the selective effect of monotonicity in the retrieval.

3.5 Logical Foundations of Query Revision

One possibility for defining term replacement is by using Mackinson's cumulative monotony property.

$$A \mathrel{|\!\sim} B, B \vdash A \Rightarrow A \mathrel{|\!\sim} C \Leftrightarrow B \mathrel{|\!\sim} C \tag{14}$$

Cumulative monotony can be used to describe the query revision described in the introduction. Consider $A =$ *information retrieval*, $B =$ *text retrieval.* We assume that *text retrieval* $\vdash$ *information retrieval*, because *text retrieval* is a specific form of *information retrieval.* When perusing the documents, the user establishes the relevance entailment

$$\textit{information retrieval} \mathrel{|\!\sim} \textit{text retrieval}$$

Cumulative monotony permits

$$\textit{information retrieval} \mid\!\sim Q \Leftrightarrow \textit{text retrieval} \mid\!\sim Q$$

The significance of this is, for example, that the user can use the query *text retrieval* instead of *information retrieval*. Any further revision Q of the query *information retrieval* can be also considered a revision of *text retrieval* and vice versa.

The user thus can choose a term, replace it by a more specific one, but still as general as possible to preserve the cumulative monotony property. The term *text* is indeed more specific than information (in the sense that the type of *information* can be among many others textual, but also that in general the term *information* is more frequent than the term *text*). On the other hand, if we consider the subset D' of the collection D, $D' \subset D$, in which *information retrieval* is relevant, then it is reasonable to think that the two terms have the same generality and thus the same effect on retrieval with respect to a background knowledge B contained in D' (i.e. the complete and ideal theory B which describes D').

4 Conclusions

The use of belief revision in IR has been explored earlier by Logan et al. (1994). This work focussed on an intermediary founded on the theory of belief revision. The work presented here, however, focuses on query reformulation.

Conditional logic is a completely different way to realize belief revision. It satisfies the Ramsey test and it has been widely explored in IR, not for query expansion, but for the retrieval of documents. In what they do differ? The answer is "always", unless the information to be added is already in the belief set. In fact belief revision coincides with a simple expansion when the information to be added is compatible. Thus $A \in K^*_A = K^+_A$ after revision. Conditional logic applies a more complex revision process in this case. It can happen that $K > A$ or $\neg(K > A)$. When the information A to be added is incompatible or syntactically inconsistent with K then conditional logic gives the answer $\neg K > A$. In fact in all closest possible worlds in which K is true is not possible that A can be also true, so that $\neg(K > A)$ is true. If the set of closest world reduces to one, then we strengthen the conclusion to $K > \neg A$. There is no way to accept A as a new belief. Belief revision instead operates a contraction and then an expansion of the set K of beliefs, namely $K^*_A (\neq K^+_A)$.

This article attempts to draw parallels between query reformulation and the theory of belief change. We argue that query expansion and query revision can be explained in terms of the revision of the user's beliefs in query terms. Query contraction is argued as being underpinned by belief expansion. Although the connection between belief change theory and query reformulation has intuitive appeal, there are problems when adopting the theory as a formalization of the query reformulation process. For example, negation has a different character in query reformulation than it does in belief change theory. More work is needed to adapt Gärdenfors' formalization and to check whether which results (or their counterparts) still hold.

Acknowledgments The work reported in this paper has been funded in part by the Cooperative Research Centres Program through the Department of the Prime Minister and Cabinet of Australia.

References

[1] G. Amati and K. Georgatos. Relevance as deduction: a Logical View of Information Retrieval. In F. Crestani and M. Lalmas, editors, *Proceedings of the Second Workshop on Information Retrieval, Uncertainty and Logic WIRUL'96*, pages 21–26. University of Glasgow, Glasgow, Scotland, 1996. Technical Report TR-1996-29.

[2] P.D. Bruza and S. Dennis. Query re-formulation on the Internet: Empirical Data and the Hyperindex Search Engine. In *Proceedings of the RIAO97 Conference - Computer-Assisted Information Searching on Internet*, pages 488–499. Centre de Hautes Etudes Internationales d'Informatique Documen taires, June 1997.

[3] P.D. Bruza and T.W.C. Huibers. A Study of Aboutness in Information Retrieval. *Artificial Intelligence Review*, 10:1–27, 1996.

[4] P.D. Bruza and B. van Linder. Preferential Models of Query by Navigation. In F. Crestani, M. Lalmas, and C.J. van Rijsbergen, editors, *Information Retrieval: Uncertainty and Logics*, volume 4 of *The Kluwer International series on Information Retrieval.* Kluwer Academic Publishers, 1998.

[5] C.W. Cleverdon. The Significance of the Cranfield Tests on Index Languages. In A. Bookstein, Y. Chiaramella, G. Salton, and V.V. Raghav an, editors, *Proceedings of the 14th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 3–12, Chicago, Illinois, October 1991. ACM Press.

[6] P. Gärdenfors. *Knowlede in Flux.* MIT Press, Cambridge, MA, 1988.

[7] Konstantinos Georgatos. Ordering-based representations of rational inference. In José Júlio Alferes, Luís Moniz Pereira, and Ewa Orlowska, editors, *Proceedings of the European JELIA Workshop (JELIA-96): Logics in Artificial Intelligence*, volume 1126 of *LNAI*, pages 176–191, Berlin, September 30–October3 1996. Springer-Verlag.

[8] Konstantinos Georgatos. Entrenchment relations: A uniform approach to nonmonotonicity. In *Proceedings of the First International Joint Conference on Qualitative and Quantitative Practical Reasoning ECSQARU-FAPR'97*, volume 1244 of *Lecture Notes in Computer Science*, pages 282–297, Berlin, 1997. Springer-Verlag.

[9] T.W.C. Huibers. *An axiomatic theory for information retrieval.* PhD thesis, University of Utrecht, Utrecht, The Netherlands, 1996.

[10] B. Logan, S. Reece, and K. Sparck Jones. Modelling Information Retrieval Agents with Belief Revision. In *Proceedings of the 17th ACM-SIGIR Conference*, pages 91–100. Springer Verlag, 1994.

[11] David Makinson and Peter Gärdenfors. Relations between the logic of theory change and nonmonotonic reasoning. In *The Logic of Theory Change*, pages 185–205, Berlin, 1991. Springer Verlag.

Some uses of Fuzzy logic in multimedia databases querying

Didier Dubois, Henri Prade, Florence Sèdes

Institut de Recherche en Informatique de Toulouse (IRIT) – CNRS
Université Paul Sabatier
118 route de Narbonne
31062 Toulouse Cedex 4 - FRANCE
{dubois, prade, sedes}@irit.fr

Abstract. Fuzzy set methods have been already applied to the representation of flexible queries in databases systems as well as in information retrieval. This methodology seems to be even more promising in multimedia databases which have a complex structure and from which documents have to be retrieved and selected not only from their contents but also from their appearance, as specified by the user. This paper provides a preliminary investigation of two potential applications of fuzzy logic in multimedia databases querying. The first one concerns the detection of modifications in semi-structured documents and relies on a graph matching procedure where subparts of the graph describing the structures have different levels of importance. The second illustrates another aspect of the fuzzy logic methodology allowing for a querying by examples process. Examples and counter-examples are supposed to be provided by the user with their levels of representativity; attributes used in their descriptions can be also weighted according to their level of importance.

1. Introduction

The full use of multimedia information systems raises new issues, especially for the access and the manipulation of information in a more intuitive, less formalised and human-friendlier way. It poses new challenges from data representation to querying and retrieval. Due to the richness of multimedia data content, querying systems must have extended capabilities: providing high-level abstractions in order to model multimedia data and their presentation, querying them on their appearance as well as by their content, querying by example and allowing for flexible queries.

It is well-known that fuzzy logic provides a framework for modeling flexibility and handling vagueness in the interface between human conceptual categories and data. Such capabilities have been already developed in the database field, specially for handling flexible queries [BP92, BP95, BDPP97, BLP98]. There has been only a few preliminary and specialized papers on the use of fuzzy logic in multimedia databases systems [BCR98, CR97, C98]. In this paper, we take the example of two potential applications of different nature and which illustrate two important aspects of multimedia querying: i) retrieving documents having similar structures, and ii) looking for documents having their contents similar to examples (and may be dissimilar to counter-examples).

In the next section, we restate multimedia databases specificities and features, versus "classical" (relational) databases requirements. We discuss what functions, traditionally allocated to DBMSs, still make sense in a multimedia environment, what new ones would be useful, specially investigating the querying step. Then, in Section 3, we summarise the main contributions of the fuzzy approach to database systems querying. In Section 4, we focus on detecting modifications in semi-structured data or document. This is important in databases area, for active databases, data warehousing, view maintenance, and version and configuration management. Most previous works in change management have dealt with flat-file and relational data, but hierarchically structured data must be also handled. We examine how approximate matching techniques make it actually possible to improve such systems capabilities. In Section 5, an approach is proposed for evaluating to what extent a document can be considered as a part of the answer to a query expressed in terms of examples and counter-examples.

2. Multimedia databases

A multimedia DBMS must address the following requirements [AN97]: traditional DBMS capabilities, huge capacity storage management, media composition, integration and presentation, query support and interactivity. It must thus cope with data modelling, object storage, indexing, retrieval and browsing. We can view a multimedia database as a controlled collection of multimedia data items, such as text, images, graphic objects, sketches, video and audio, which can be related by temporal and/or spatial constraints. The multimedia DBMS should accommodate these special requirements by providing high-level abstractions in order to manage the different data types, along with a suitable interface for their presentation.

Audio data, for example, can be automatically indexed by signal analysis or speech recognition followed by keyword-based indexing. Images cannot: shape, colour, texture, identified through pattern recognition, are not sufficient, this is why textual descriptions usually refers to them. Thus, representing multimedia information such as pictures or image sequences raises some problems for their retrieval due to the limitations of textual descriptions usually associated to it, versus the massive information available according to the context, interpretation, user's profile, etc. The lack of standard structure and the variety of potential available information means that it can be difficult to express precise queries.

Thus, characteristic features of multimedia data are their lack of structured information, multiplicity of data types and heterogeneity of formats, spatial and temporal characteristics, and inadequacy of textual descriptions.

Queries in relational systems are exact match queries: the system is able to return exactly those tuples a user is precisely asking for and nothing more. To specify the resulting relation, conditions concerning known attributes of known relations can be formulated.

From the Information Retrieval area, we know how difficult it is to characterise the contents of textual objects [FBY92]. Problems are encountered on the one hand in specifying the contents of the objects, and on the other hand, for describing the objects one is looking for. In multimedia databases, the question is thus how to characterise the 'contents' of image, audio or video data. Indeed, we must face the problem of giving an interpretation to a photo or a song, for which many interpretations exist in general, according to the context, the user's point of view [Ape97].

Multimedia data must be preferably interpreted before they can be queried, in order to generate content descriptions (otherwise it might be more costly to make it on line). Multimedia information can be retrieved using identifiers, attributes, keywords,... Note that indexing is context-dependent. Introducing abstractions allows the user to refer to the data in terms of high level features or metadata, which constitute his model of the application domain.

Extensions of conventional concepts of query languages -all the retrieved objects exactly match to the query- require approaches that can deal with the temporal and spatial semantics of multimedia data, or query languages that can incorporate flexibility in the expression of requests. Indeed, queries are usually imprecise, so, relevance feedback and meaning similarity, rather than exact matching, and mechanisms for displaying ranked results are important. This is particularly important in combination with content-based access, where the user's criteria are often approximately and imprecisely formulated. Moreover, multimedia querying should offer support for new requirements such as querying by examples, querying spatial or temporal data, flexible querying using fuzzy predicates.

In this context, the concept of document generally refers to any composite object combining elements characterised by different media. Making the most open form of querying possible is important to allow queries to refer to the document appearance as well as its content. So, the fuzzy set techniques referred to in the next section can be applied in multimedia data querying.

The next section deals with flexible queries to a "classical" database, that can also work on an abstract representation (metadata) of the objects, via indexes for instance.

3. Fuzzy data bases

Research on "Fuzzy databases" (see [BK95, Pet96]) has been developed for about twenty years. These works have been mainly concentrating on flexible querying in classical languages (e.g., [KZ86, BP95]), and in data representation and object-oriented languages [DeC97, C98]. An introduction to these different issues may be found in a recent survey by Bosc and Prade [BP97].

The flexibility of a query reflects the preferences of the end-user. Using a fuzzy set representation, for expressing a flexible selection criterion, the extent to which an object described in the database satisfies a request then becomes a matter of degree. In this case, the end-user provides set of attribute values which are fully acceptable for him as well as set of values which are clearly not acceptable, the remaining values being rank-ordered according to their level of acceptability. Moreover, a query may also allow for some similarity-based tolerance: close values are often perceived as similar, interchangeable. Indeed, if for instance an attribute value v satisfies an elementary requirement, a value "close" to v should still somewhat satisfy the requirement. The introduction of a tolerance relation can make flexible a non-fuzzy query as in the example: "find people less than 40 year old" and where a 41 year old person can be retrieved with an intermediary degree of matching based on the (context-dependent) appraisal of the proximity between 40 and 41.

An advantage of fuzzy set-based modelling, is that it is mainly qualitative in nature. Indeed in many cases, it is enough to use an ordinal scale for the membership degrees (e.g., a finite linearly scale). This also facilitates the elicitation of (context-dependent) membership functions, for which it is enough in practice to identify the elements which totally belong and those which do not belong at all to the fuzzy set.

Fuzzy set membership functions [Zad65] are convenient tools for modelling user's preference profiles and the large panoply of fuzzy set connectives can capture the different user attitudes concerning the way the different criteria present in his/her query compensate or not; see [BP92] for a unified presentation in the fuzzy set framework of the existing proposals for handling flexible queries.

Thus, the interest of fuzzy queries for a user are twofold:

i) A better representation of his/her preferences. For instance, "he/she is looking for an apartment which is not too expensive and not too far from downtown". In such a case, there does not exist a definite threshold for which the price becomes suddenly too high, but rather we have to differentiate between prices which are perfectly acceptable for the user, and other prices, somewhat higher, which are still more or less acceptable (especially if the apartment is close to downtown). Obviously, the meaning of vague predicate expressions like "not too expensive" is context/user dependent, rather than universal.

The large panoply of fuzzy set connectives can capture the different user's attitude concerning the way the different criteria present in his/her query compensate or not. Moreover in a given query, some part of the request may be less important to fulfil; this leads to the need for weighted connectives.

ii) Fuzzy queries, by expressing user's preferences, provide the necessary information in order to rank-order the answers contained in the database according to the degree to which they satisfy the query. It contributes to avoid empty sets of answers when the queries are too restrictive, as well as large sets of answers without any ordering when queries are too permissive.

4. Looking for similar semi-structured documents

In the context of documentary applications, one of the main differences between a multimedia database and an ordinary one from a querying point of view, is that in the first case the request may refer to a document in terms of its appearance and not only in terms of its information contents. The lack of standardised structure of the document(s) to be retrieved calls for the use of flexible queries.

Indeed, an important class of multimedia documents are semi-structured documents, like HTML or XML documents [CS98]. The idea of allowing for flexibility in the exploitation of semi-structured information [Abi97] consists here to refer to the structure in an approximate matching procedure.

For example, a request may refer to the description of a document structure (e.g. title, author(s), probably an abstract, maybe key-words, a body structured in sections and sub-sections, certainly followed by references, among which we preferably may find some author's name...). Thus, a first level of querying is to retrieve an already seen document where the description refers to its structure [DS96] in a flexible way.

In the following, the problem of detecting modifications in semi-structured data or document is addressed. This problem is challenging due to the irregularity, incompleteness and lack of schema that characterise semi-structured data.

Structural matching and discovery in document such as HTML are useful for data warehousing, version management, hypertext authoring, digital libraries and web databases, for instance to know modifications in an HTML document or to find common substructures. Indeed, in addition to offering access to large collection of heterogeneous and semi-structured data (e.g. HTML documents), the Web allows this information to be updated at any time. These rapid and often unpredictable changes create a problem of detecting these modifications. A user may visit documents repeatedly and is interested in knowing how each document has changed since the last visit. This may be achieved by saving a snapshot of the previous HTML pages at the site (something that most browsers are able to do anyway). Assuming we have saved the old version of the document, we want to periodically detect the changes due to their evolutions, by comparing the old and new versions of the document and knowing how much similar they are.

Most previous work in change management has dealt only with text files [Mye86, Kif95], flat-files or relational data (for example, [LGM95] presents algorithms for efficiently comparing sets of records that have keys). A system, called *LaDiff*, has been implemented to detect, mark, and display changes in structured documents. It is based on their hierarchical structure analysis, taking two versions of a Latex document as input and producing as output a Latex document with the changes marked [PGMW95].

Much database research has been conducted about structured documents such as SGML [CACS94, CLS94, MZ98] : given a Document Type Definition (DTD), any document will conform to it. This DTD is, like a database schema, a generic structure for a class of documents. Few systems have been built to support semi-structured data. Indeed, these data are irregular, incomplete and does not necessarily conform to a fixed schema. Semi-structured data may have some structure but, if it exists, it is enclosed into the instance and a priori unknown. It must be elicited, in order to discover patterns. From a given mark-up language, a parser can analyse the document content in order to elicit the structure items. Documents are represented as ordered labelled trees, since hierarchically structured information can be represented as trees, in which the children of each node have a designated order (see *Fig. 1*).

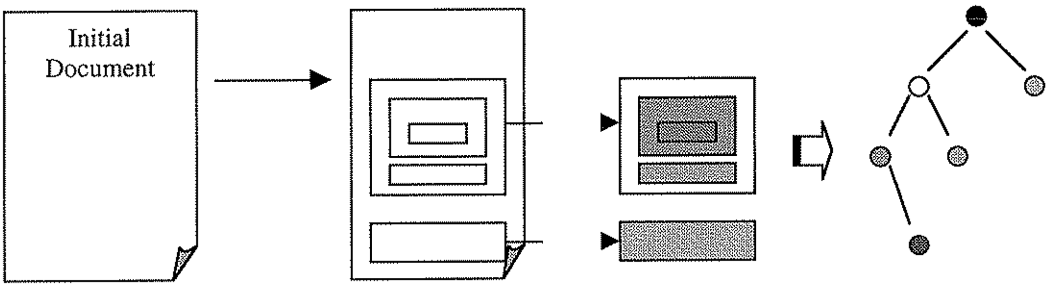


Fig. 1. Extracting the document structure

We must identify changes not just to the "nodes"' in the data, but also to their relationships. For example, if a node (and its children) is moved from one location to another, or if a leaf node has been split up into its children, we would like to detect it (e.g., given a level *l*, the structure has been reorganised, directly attaching the children at level *l+1*, to the father at level *l-1* (see *Fig. 2*)).

Hence, one of our problems is to find an appropriate matching for the trees we are comparing. In some application domains, the problem is easy, such as when data objects contain object identifiers or unique keys. In other domains, such as structured documents, the matching is based on labels and values only, so it is more difficult. Two documents can be compared using tree matching techniques [SZ90, ZS89, WZJS94]. Furthermore, not only do we want to match trees that are identical (with respect to the labels and values of the nodes and their children), but we also want to match trees that are "approximately equal", according to their similarity. Graph structure is not adapted to comparison other than exact identity. In the following, we suggest a process adapted to comparison algorithms [SZ90, ZS89], by successive adaptation, for example deleting a structure level if this one is considered as not much relevant. One of the main characteristics of this model is that it saves the ordering between elements. Without it, the problem of "subtree" relations between two trees cannot be handled [Kil92].

The possible cohabitation of multiple views of the same document handicaps any approach based on strict comparison. An additional semantic knowledge about elements (nodes) or relationships (arcs) is necessary to allow an approximate comparison. This knowledge can be extracted from the element name (mark-up content), from the attributes or their values, or from any external user's specification. Our approach to the problem of detecting modifications in pieces of hierarchically structured information has two main original features:

- *Semi-structured data.* Although some database systems, particularly active database systems, build change detection facilities into the system itself, we focus on the problem of detecting changes given old and new versions of the data. What can be known as a structure of a semi-structured document should be extracted from the encoding of the document itself.
- *Approximate matching.* By approximate matching of graph, we mean that the graphs at least match for their essential parts and if possible, for their less important subpart. Then, the matching becomes a matter of level.

A first version of the elicitation process of the structure (without any grading of the importance of each subpart of the document) has been implemented in the rewriting tool called eXrep [LQC95]. The ordered tree-like structure of the document is built by recognising structure items from the content itself, through automatic identification of marks and their rewriting by means of dictionaries. From this rewriting process, it is possible to identify in the tree eddges which can be considered as more important than others: from the document analysis, a syntactic and semantic marking can allow to infer that some edges have a lower weight than other ones. The ideas at the basis of the weights' computation are that the level of importance of the edge depends on the recognised mark, its content, the text associated with the pending node, and the depth of the tree representing the structure. Indeed, comparison cannot rely on structural similarities while content is ignored. The weighting process first deals with structural comparison and mark-up semantics, second with the content. In practice, one could only distinguish between a rather small number of levels in the importance scale.

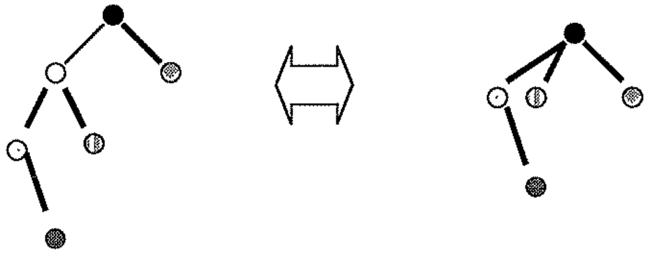


Fig. 2. Comparing two structures

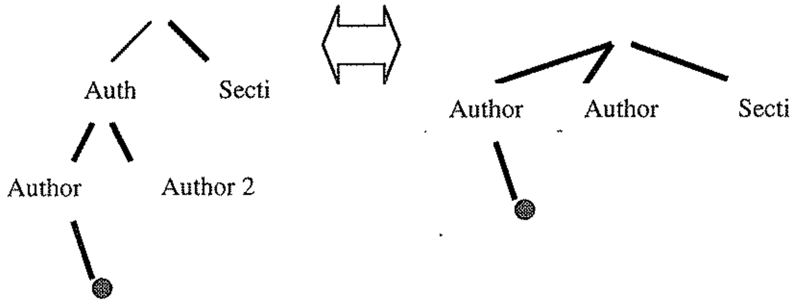


Fig. 3. Example of similar structures

Given two tree structures T^1 and T^2 whose nodes are weighted on a scale

$w_1 = 1 > w_2 > ... > w_n > w_{n+1} = 0$

their similarity can be evaluated in the following way. Let T_i be the tree built from T by only keeping edges whose level is greater or equal to w_i and fusing the nodes belonging to a suppressed edge.

Then

$\text{similarity}(T^1, T^2) = w_{n-i+1}$

where the function i → n-i+1 is the order-reversing map of the scale $\{1,\ldots,n\}$

if $T^1{}_i$ and $T^2{}_i$ are identical but $T^1{}_{i+1}$ and $T^2{}_{i+1}$ are different,

and $\text{similarity}(T^1, T^2) = 0$ if $T^1{}_1$ and $T^2{}_1$ are different.

So

$\text{similarity}(T^1, T^2) = 1$

if T^1 and T^2 are similar at each level, from 1 to n (at each level, similarity is binary, equals to 0 or 1).

What about the content ? When comparing T^1 with an approximation of T^2 where some edge has been deleted, we may have to detect whether the text associated with the suppressed node, if there is one, is pasted elsewhere in the approximation of T^2. More generally we may think in the computation of the similarity of adding requirements on identity or subsumption of the attached texts (in order to have a nonzero similarity).

Thus, given a document, it is possible to retrieve and rank-order approximately similar documents.

5. Querying by examples

The user is not always able to easily express his request even in a flexible way. First, it may be more convenient for him to express what he is looking for from examples. Second, since he may have absolutely no a priori knowledge of the amount of retrievable documents (for a given request), it may be useful if the system is able to guide him about what exists, by incrementally building queries from examples.

Querying based on examples, for eliciting user's preferences, can provide the necessary information for building a query. Thus, the user may say to what extent a few examples of documents are representative of what he is looking for by using some (finite) similarity scale. Then, relevance of current documents should be evaluated in terms of their similarity with respect to these examples and may be counter-examples. Issues are then close to fuzzy case-based reasoning [DEGGLP98].

Examples as well as counter-examples are described in terms of precisely known attribute values. These attributes are supposed to be relevant and sufficient for describing their main features from a user's point of view. Let a_{ij} with i=1,m, and j=1,n be the value of attribute i for example j. Let b_{ik} with k=1,p be the value of attribute i for counter-example k. Let λ_j and ρ_k be the extent to which example j and counter-example k respectively are highly representative (from the user's point of view). Let us now consider a document d described by the attribute values $d_1,..,d_m$, supposed to be precisely known (the attributes are supposed to be the same as the ones used for describing the examples and counter-examples). Clearly, the more *similar* d to *(at least)* a representative example and the more *dissimilar* to *all* counter-examples, the more *possible* d is eligible for the user.

This can be estimated by the following expression:

$$\mu(d) = \min(ex, c\text{-}ex) \qquad \textbf{(1)}$$

where $ex = \max_j \min(\lambda_j, r_j)$

is the extent to which there exists *at least one* highly representative example similar to d provided that r_j be the similarity between example j and d.

Note that if all the examples are fully representative (i.e. $\forall j$, $\lambda_j = 1$) then $ex = \max_j r_j$ as expected; if $\lambda_j = 0$ example j is not considered,

where $c\text{-}ex = \min_k \max(1-\rho_k, s_k)$

is the extent to which *all* highly representative counter-examples are dissimilar to d provided that s_k be the dissimilarity between counter-example k and d.

Note that if all the counter-examples are fully representative (i.e. $\forall k$, $\rho_k = 1$) then

$c\text{-}ex = \min_k s_k$ reduces to a conjunctive aggregation; if $\rho_k = 0$, counter-example k is not considered,

Then, let S_i be a fuzzy similarity relation on the attribute domain of i with membership function μ_{S_i} (S_i is supposed to be reflexive and symmetrical) and w_i be the level of importance of attribute i in a description,

the similarity between d and example j is computed as

$r_j = \min_i \max(1-w_i, \mu_{S_i}(d_i,a_{ij}))$ (example j and d are similar as far as *all* the highly important attribute values which describe them are similar),

the dissimilarity between d and counter-example k is computed as

$s_k = \max_i \min(w_i, 1-\mu_{S_i}(d_i,b_{ik}))$ (counter-example k and d are dissimilar as far as there exists an highly important attribute for which their respective values are much dissimilar).

When w_i=1 $\forall i$, r_j is just the conjunctive aggregation of the similarity degrees and s_k the disjunctive combination of dissimilarity degrees).

Finally, we get

$$\mu(d) = \min[\max_j \min(\lambda_j, \min_i \max(1-w_i, \mu_{S_i}(d_i,a_{ij})), \quad \min_k \max(1-\rho_k, \max_i \min(w_i, 1-\mu_{S_i}(d_i,b_{ik}))] \qquad (2)$$

This evaluation might be improved by requiring the similarity of d with *most* examples (see, e.g., [DPT88] for the introduction of a soft quantifier in a weighted min expression).

It is worth pointing out that the above expression can be used in another way that we now briefly outline and that is a topic for further research, apart from the ranking of documents w.r.t. a query. This expression also provides the starting point for generating a description of the documents which are looked for. This description may then be used for interface needs with the user. Indeed, letting d unspecified in the expression, it defines a compound fuzzy set expression which can be logically analysed. Suppose for simplicity that all the weights are equal to 1, we obtain:

$$\mu(.) = \min[\max_j \min_i \mu_{S_i}(\,.\,,a_{ij}), \min_k \max_i 1-\mu_{S_i}(\,.\,,b_{ik})] \qquad (3)$$

Clearly, $\mu_{S_i}(\,.\,,a_{ij})$ defines a fuzzy set of values close to a_{ij}, while $1-\mu_{S_i}(\,.\,,b_{ik})$ defines a fuzzy set of values significantly different from b_{ik}, for each attribute i.

Note that the above expression may incorporate *interactivity* [Zad75] between attributes. For instance we have two examples of documents on a given topic, one which is 'short' without illustrations, and another which is 'long' but having many illustrations. Such a set of examples suggest that acceptable documents may be 'long' only if they have 'many illustrations' in them. In this case, there is an interaction between the length of the document and the number of illustrations. This interaction will be embodied in the above type of expressions and will make its reading less simple.

It should be pointed out that the set of examples and counter-examples provided by the user, enlarged by the similarity relations, does not usually cover all the cases which can be encountered. Namely, a document in the database may be just dissimilar to all the counter-examples without being similar to any examples. This suggests that if all the stored documents are in this situation, we may only use the c-ex part of the evaluation in order to avoid an empty answer to a request. However, the set of answers provided on the basis of c-ex only may be too large if it is not properly focused on the context of interest. For instance, looking for documents on a given topic having some length and number of illustrations characteristics, c-ex should only apply to these latter characteristics, in order not to retrieve all the documents on a different topic !

Another method for not "forgetting" the documents which may be in between the examples and the counter-examples, would be rather to allow for requests on the form: "all the documents satisfying some properties are relevant except the ones similar to counter-examples", or: "only the documents somewhat similar to at least one of the examples are relevant". Mixed requests specifying that the documents similar to example(s) are welcome, these similar to counter-examples should be excluded, while the remaining ones (in the context of interest) are provided to the user only if the search is enlarged (then it may help the user providing further examples and counter-examples).

6. Conclusion

The intended purpose of this paper was to provide a preliminary investigation of potential applications of fuzzy logic techniques in multimedia databases. Emphasis has been put on two querying issues: detecting modifications in semi-structured documents, and querying by examples. The different uses of fuzzy logic techniques in relation with these two querying problems have been discussed. The application of these tools to semi-structured documents could be developed to hypermedia. Our work could be adapted to the detection of changes in documents that can be represented as graphs but not necessarily as trees. Concerning the querying by examples, instead of counter examples, we may think of hybrid queries made of examples and of classical (fuzzy) restrictions expressing what attribute values are undesirable.

7. References

Abi97, Abiteboul S., Semi-structured information. In Proc. of Intl Conf. On Database Theory ICDT'97, International Conference on Database Theory, Invited talk, 1997.

ACM95, Abiteboul S., Cluet S., and Milo T., A database interface for file updates. In *Proceedings of the ACM SIGMOD International Conference on Management of Data,* 1995.

AN97, Adjeroh D. A., Nwosu K. C., Multimedia Database Management Requirements and Issues, IEEE Multimedia, Vol. 4, n 3, pp. 24-33, 1997.

Ape97, Apers P. et al, Multimedia Database in Perspective, Springer-Verlag, 1997.

BCR98, Bosc P., Connan F., Rocacher D., Flexible querying in multimedia databases with an object query language. Proc. 7th IEEE Int. Conf. on Fuzzy Systems, Anchorage, May 5-10, 1308-1313, 1998.

BDPP97, Bosc P., Dubois D., Pivert O., Prade H. (1997) Flexible queries in relational databases —The example of the division operator— Theoretical Computer Science, 171, 281-302.

BK95, Bosc P., Kacprzyk J. (Eds.) Fuzziness in Database Management Systems. Physica-Verlag, Heidelberg, 1995.

BLP98, Bosc P., Liétard L., Prade H. (1998) An Ordinal Approach to the Processing of Fuzzy Queries with Flexible Quantifiers. in: Applications of Uncertainty Formalisms (A. Hunter, S. Parsons Eds.). Springer Verlag LNCS 1455, pp. 58-75, 1998.

BP92, Bosc P., Pivert O., Some approaches for relational databases flexible querying. J. of Intelligent Information Systems, 1, 323-354, 1992.

BP95, Bosc P., Pivert O., SQLf: A relational database language for fuzzy querying. IEEE Trans. on Fuzzy Systems, 3(1), 1-17, 1995.

BP97, Bosc P., Prade H., An introduction to the fuzzy set and possibility theory-based treatment of soft queries and uncertain or imprecise databases. In: Uncertainty Management in Information Systems: From Needs to Solutions (A. Motro, Ph. Smets, eds.), Kluwer Academic Publ., Chapter 10, 285-324, 1997.

CACS94, Christophides V., Abiteboul S., Cluet S., Scholl M., From structured documents to novel query facilities. 1994 ACM SIGMOD Intl Conf. on Management of Data, pp. 313-324, Minneapolis, May 1994.

CGM97, Chawathe S., Garcia-Molina H., Meaningful Change Detection in Structured Data, *Proceedings of the ACM SIGMOD International Conference on Management of Data,* 1997.

CLS94, C. Chrisment, P.Y. Lambolez, F. Sèdes, Hyperdocument Management and Exchange in an OO System, Indo-French Workshop on OO Sytems, pp. 313-333, Goa, India, 1994.

C98, Connan F., Interrogation flexible dans un environnement objet. Rencontres Francophones sur la Logique Floue et ses Applications, Lannion, pp. 253-259, 1998.

CR97, Connan F., Rocacher D., Gradual and flexible Allen relations for querying video data. Proc. 5th Eur. Congr. on Intelligent Techn. and Soft Computing (EUFIT'97), Aachen, Germany, Sept. 8-11, 1132-1136, 1997.

CS98, Chrisment C., Sèdes F., Bases d'objets documentaires. Tutoriel, INFORSID'98, 1998.

CRGMW95, Chawathe S., Rajaraman A., Garcia-Molina H. and Widom J., Change detection in hierarchically structured information. *Proceedings of the ACM SIGMOD International Conference on Management of Data,* 1996.

DeC97, De Caluwe R. (ed.) Fuzzy and Uncertain Object-Oriented Databases. Concepts and Models. World Scientific, Singapore, 1997.

DS96, Djennane S., Sedes F., Audio facilities for hypermedia consultation. 2nd Intl Workshop on Natural Language and Databases, NLDB'96, Amsterdam, IOS Press, 91-101, 1996.

DEGGLP98, Dubois D., Esteva F., Garcia P., Godo L., Lopez de Mantaras R. and Prade H., Fuzzy set modelling in case-based reasoning. Int. J. of Intelligent Systems, 13, 301-374, 1998.

DPT88, Dubois D., Prade H., Testemale C., Weighted Fuzzy Pattern Matching, Fuzzy Sets and Systems, Vol. 28 n° 3, 313-331, 1988.

FBY92, Frakes W. B., Baeza-Yates R., Information Retrieval: Data Structures and Algorithms, Prentice-Hall, 1992.

Gro97, Grosky W.I., Managing Multimedia Information in Database Systems, Communications of the ACM, Vol. 40, n 12, pp. 73-80, 1997.

Kif95, Kifer M., EDIFF- a comprehensive interface to diff for Emacs 19. Available through anonymous ftp at `ftp.cs.sunysb.edu`, 1995.

Kil92, Kilpelaïnen P., Tree matching problems with application to structured text databases. Tech. Report, Dept. of Computer Science, Univ. of Helsinky, Finland, 1992.

KZ86, Kacprzyk J., Ziolkowski A., Data base queries with fuzzy linguistic quantifiers. IEEE Trans. on Systems, Man and Cybernetics, 16(3), 474-478, 1986.

LGM95, Labio W. and Garcia-Molina H., Efficient algorithms to compare snapshots. Manuscript, available by anonymous ftp from `db.stanford.edu` in `pub/labio/1995/`, 1995.

LQC95, P.Y. Lambolez, J.P. Queille, C. Chrisment, EXREP : a generic rewriting tool for textual information extraction. Revue ISI, "Ingénierie des Systèmes d'Information", vol. 3, n° 4, pp. 471-485, 1995, Hermès.

MZ98, Milo T., Zohar S., Using Schema Matching to Simplify Heterogeneous Data Translation, VLDB'98, pp. 122-133, New York, August 1998.

Mye86, Myers E., A difference algorithm and its variations. *Algorithmica*, 1(2):251--266, 1986.

Pet96, Petry F.E., Fuzzy Databases: Principles and Applications. Kluwer Acad. Pub., Dord, 1996.

PGMW95, Papakonstantinou Y., Garcia-Molina H., and Widom J., Object exchange across heterogeneous information sources. In *Proceedings of the 11th International Conference on Data Engineering*, pp. 251-260, Taipei, Taiwan, March 1995.

SZ90, Shasha D. and Zhang K., Fast algorithms for the unit cost editing distance between trees. *Journal of Algorithms*, 11(4):581-621, 1990.

WSCRZP97, Wang J., Shasha D., Chang G., Relih V., Zhang K. and Patel G., Structural Matching and Discovery in Document Databases, *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp. 560-563, 1997.

WZJS94, Wang J. T. L., Zhang K., Jeong K. and Sasha D., A system for approximate tree matching. IEEE Transactions on Knowledge and Data Engineering, 6(4) :559-571, August 1994.

Zad65, Zadeh L.A., Fuzzy sets. Information and Control, 8, 338-353, 1965.

Zad75, Zadeh L.A., The concept of a linguistic variable and its application to approximate reasoning, Information Sciences, Part 1: 8: 199-249, Part 2: 8: 301-357, Part 3: 9: 43-80. Reprinted in [SP], 219-366. 1975.

ZS89, Zhang K. and Shasha D., Simple fast algorithms for the editing distance between trees and related problems. *SIAM Journal of Computing*, 18(6):1245-1262, 1989.

MEASURING THE EFFECTS OF *AND, AND NOT* and *OR* OPERATORS IN DOCUMENT RETRIEVAL SYSTEMS USING DIRECTED LINE SEGMENTS

M. H. Heine

School of Information Studies, University of Northumbria
Newcastle upon Tyne, England. NE1 8ST
michael.heine@unn.ac.uk

Abstract. The expression by cognition of its own 'natural logic' in boolean terms provides the essential controlling input to document retrieval systems. Evaluation of the effectiveness of such inputs is facilitated by the 'signal detection model' of searcher/database interaction, which historically has focused on the evaluation of 'Recall/Precision' value-pairs without any explicit emphasis on the role of boolean logic as determinant of such pairs. This paper suggests that directed-line segments (and their associated vectors) provide useful characterisations of boolean effects, whether within the historical Recall/Precision outcome space, or an alternative outcome space defined by Precision and the Number of informing documents retrieved. Bundles of DLSs are illustrated for simple *AND, OR* and *AND NOT* relationships between pairs of attributes, using empirical data that attempts to simulate real life document retrieval. In particular, the ratio of the lengths of mean DLSs offers a quantification of the sensitivity of a database to the use of the *AND* , *OR* and *AND NOT* operators. The vector sum of the mean vectors associated with these operators, when conditioned by the null search expression, offers a further construct that might be said to define the 'logical compactness' of the interaction between sender and database. It is also suggested that sequences of DLSs might usefully portray the effect of a given database searcher using a sequence of search expressions, as in those document retrieval systems that support feedback from the user.

1. Introduction

We define 'information' here as a cognitive process, rather than 'a property of objects', and regard the mechanical systems we call 'information systems' as facilitating this process through the bringing of extra-cognitive (i.e. 'public' or 'social') recorded knowledge ('documents') within the purview of cognition. Document retrieval is controlled by the human searcher, who inputs to the system a search expression made up of one or more string literals representing knowledge (usually keywords, but sometimes classification codes), and none or more of the boolean operators *AND*, *OR* or *AND NOT* connecting these variables. (Usually parentheses are also permitted.). The system interprets the literals as binary variables (i.e., asks whether each document or document surrogate in the store is characterised by that literal or not?) and then evaluates the search expression to *True* or *False*. Documents that evaluate the search expression to *True* are said to be 'retrieved' and can be displayed to the searcher. Usually, the searcher chooses operators according to his or her experience and intuition, possibly aided by a human or machine intermediary, just as the string literals are so chosen. However, the 'character' of the system's response to those choices is not forseeable by the searcher, nor may the searcher be aware of the quantitative differences in effect of choosing different logic operators. In this paper, we attempt (1) to illustrate the effects of *AND* and *OR* in *visual* terms, and (2) to provide a simple quantitative measure of the differences in effect of these two operators. The use of the *AND NOT* operator is also discussed, but only briefly in view of the relative rarity of its use.

2. The signal-detection paradigm

Our approach represents an extension of the signal detection paradigm or 'model' of document retrieval first proposed by J. A. Swets [1], and re-expressed in a discrete form based on elementary logical conjuncts by the author [2]. The model, so revised, can be described fairly quickly in its *classical* form as follows:

1. A set of potentially informing *documents* is characterised by a set of *attributes* (above: 'string literals'.). These attributes are chosen from a pre-defined inventory of same (usually termed a 'thesaurus') by an 'indexer'. Searching is essentially a process which requires the searcher to guess or 'perceive' the attributes that have been chosen by the indexer to characterise the documents that will inform him.
2. Thus a searcher of a set of potentially informing documents:
    a) First chooses a set of attributes which he or she believes will characterise informing documents
    b) Secondly, chooses a boolean expression that will retrieve a particular combination of documents, namely those that evaluate the expression to *True*, based on those attributes. By 'based on' we mean that an elementary logical variable (ELVs) is associated with each attribute, and that these ELVs along with boolean operators and parentheses also chosen by the searcher, form the search expression of interest.
3. The above process defines (at least in principle—see later) two probability distributions over the elementary logical conjuncts (ELCs) of the set of attributes chosen by the searcher. One is for the set of documents that *would* inform the searcher were they to be retrieved—the set of so-called 'relevant documents, the

other is for the remaining documents. Since ELCs partition any set of documents, the probabilities making up the distribution add to 1 in each case.

To illustrate the model, suppose a searcher identifies just two attributes, say, t_1 and t_2. (The 't' notation is suggested by our disposition to think of document attributes as *terms*.) Then we can define elementary logical variables $\mathtt{t}_1$ and $\mathtt{t}_2$, associated with these literals, such that $\mathtt{t}_i$ evaluates to *True* when a document has been characterised by the indexer using t_i, and otherwise to *False*. The possible ELCs made up from these two ELVs are shown in the following table:

$\mathtt{t}_1$ && $\mathtt{t}_2$
$\mathtt{t}_1$ && $\neg\mathtt{t}_2$
$\neg\mathtt{t}_1$ && $\mathtt{t}_2$
$\neg\mathtt{t}_1$ && $\neg\mathtt{t}_2$

—using '¬' for '*NOT*', and '&&' for '*AND*'. (Later, we use '||' for '*OR*'.) Each ELC can then be associated with two frequencies: the number of potentially informing documents that evaluate that ELC to *True*, and the number of 'other' (i.e. not-potentially informing) documents that evaluate it to *True*. For example, for a set of 3000 documents, 100 of which are flagged by some prior procedure as potentially informing, we might have frequencies as shown below:

ELCs	Number of informing documents evaluating the ELC to *True*, n_i	Number of non-informing documents evaluating the ELC to *True*, m_i
$\mathtt{t}_1$ && $\mathtt{t}_2$	55	15
$\mathtt{t}_1$ && $\neg\mathtt{t}_2$	10	6
$\neg\mathtt{t}_1$ && $\mathtt{t}_2$	20	12
$\neg\mathtt{t}_1$ && $\neg\mathtt{t}_2$	15	2867
$\mathtt{t}_i$ || $\neg\mathtt{t}_i$	100	2900

The above frequencies are easily converted to probabilities, as follows:

ELCs	Probability that an informing document evaluates the ELC to *True*, r_i	Probability that a non-informing document evaluates the ELC to *True*, f_i
$\mathtt{t}_1$ && $\mathtt{t}_2$	0.55	0.00517
$\mathtt{t}_1$ && $\neg\mathtt{t}_2$	0.10	0.00207
$\neg\mathtt{t}_1$ && $\mathtt{t}_2$	0.20	0.00414
$\neg\mathtt{t}_1$ && $\neg\mathtt{t}_2$	0.15	0.98862

The notation r_i and f_i is suggested by the familiar Recall and Fallout concepts of (so-called) information retrieval (IR) studies. The probability that a document that evaluates an ELC to *True* is also an *informing* document, i.e. the 'Precision' of that ELC, again borrowing IR terminology, can easily be found using Bayes' Theorem. This is simply the value of $n_i/(n_i+m_i)$ for each row, or we can (more elegantly?) calculate it using $gr_i/(gr_i+(1\text{-}g)f_i)$, where g is the probability that an document is informing. In the above case, g has the value 100/3000. The following table records this probability for each ELC, in addition to r_i and f_i.

ELCs	Probability that an informing document evaluates the ELC to *True*, r_i	Probability that a non-informing document evaluates the EL:C to *True*, f_I	Probability that a document that evaluates the ELC to *True* is also informing, p_I
t_1 && t_2	0.55	0.00517	0.7857
t_1 && $\neg t_2$	0.10	0.00207	0.6250
$\neg t_1$ && t_2	0.20	0.00414	0.6250
$\neg t_1$ && $\neg t_2$	0.15	0.98862	0.0052

From such a table we can calculate the effects on the search of:

0. Imposing *an order*—strong or weak—on the ELCs (such as might be determined by a 'weighting formula', or else determined by arbitration), and then disjoining ELCs successively 'across that order'. It is on this basis that the claim was made in [2] that document retrieval based on 'weighting' is *precisely* equivalent to boolean searching. This follows from the observation that assigning *numeric* values to ELCs and then using ELCs to retrieve documents only when the numeric value exceeds some arbitrary criterion value, is tantamount to disjoining the ELCs concerned and retrieving only those documents that evaluate the disjunction to *True*. The only (temporary) awkwardness here is where the document weighting expression includes *document-specific* properties in addition to ELC-specific ones. But such properties can be accommodated within more generally defined, i.e. 'richer', ELCs. The limited returns attendant on using such weighting expressions, incidentally, simply reflects the fact that most of the richer ELCs so defined will attract zero documents.) An order imposed on the ELCs determines a *sequence* of values of evaluative variables, e.g. a sequence of paired (*p*,*r*) values, as disjuntcion takes its course through the ordering.

0. Choosing an arbitrary boolean expression. It is well known from discussions centred on 'reduction to disjunctive normal form', that any boolean expression can be expressed as an *OR*ing of ELCs made up from its constituent ELVs. For example, the simple boolean expression t_1 is precisely equivalent to:

$$(t_1 \;\&\&\; t_2) \;||\; (t_1 \;\&\&\; \neg t_2)$$

The argument that allows us to say that allows us to add together the document frequencies pertaining to the ELCs concerned (or equivalently to add the r_i and f_i probabilities—although not the p_i probabilities!). For example, a document search using the single ELV t_1 will retrieve 65 informing and 21 non-informing documents. Equivalently, it identifies 0.65 of the informing documents and 0.00724 of the non-informing documents. The probability that an document that evaluates t_1 to *True* happens also to be an informing document is 0.7558 (i.e., 65/(65+21)). Searching using the expression $t_1 \,||\, t_2$, on the other hand, would retrieve more (0.85) of the informing documents (and 0.1138 of the non-informing documents) but it does so with a reduced Precision (0.7203). The phenomenon that the *r* value *increases* with *OR*ings while at the same time the value *decreases* is a very commonly observed empirical fact in the IR literature, although not inevitable, i.e. not a tautological truth. [3,4]

3. Defining the set of *all possible* boolean expressions, in order to portray the *overall* sensitivity of the set of documents for searcher using the attributes t_1 and t_2, and a given *a priori* partitioning into informing and non-informing documents. This can show the distribution of all such expressions over (say) the $P \times R$ outcome space, where $p_i \in P[0,1]$, and $r_i \in R[0,1]$, or over the $R \times F$ outcome space, $f_i \in F[0,1]$.

To summarise, in the signal-detection model of document retrieval, the individual is regarded as choosing attributes of recorded-knowledge that he or she perceives will be attached to the documents that will inform him or herself. The scope for interaction between the individual and the set of documents is then described using two probability distributions, defined over ELCs based on those chosen attributes. One of these distributions is for those documents that *could* contribute to his information process (i.e. would contribute were they to be retrieved), the other is for the remaining documents. The problem for the individual is then to strike a balance between identifying ('retrieving') as many as possible of those documents that could inform him, with an acceptable level of delivery of non-informing documents retrieved at the same time. The individual's decision takes the form of defining a boolean search expression that optimises this balance, the effect of that decision being characterised by the performance variables of Precision, P (describing the level of 'noise' in the documents retrieved) and Recall, R, (describing the 'adequacy' of the retrieval of potentially informing documents.)

'Uncertainty' in the classical model thus appears as a macro-level effect, one which appears to aggregate *four separately-fallible processes*:

- the initial generation of a knowledge attribute set (e.g. the creation of a controlled indexing vocabulary, serving the purpose of knowledge representation)
- the assignment of attributes from that set to individual knowledge documents (e.g. the assignment of terms from a controlled vocabulary to documents, to represent the knowledge in them)
- the choice of attributes by the individual to serve his or immediate information needs (e.g. the selection of terms from a vocabulary, these being terms that the individual perceives will have been used to represent the knowledge he or she is seeking and which itself is perceived to exist.
- the choice of a boolean syntax based on the above attributes, i.e. suitable operators and parentheses.

It may be helpful to note, in the last regard, that the 'natural logic' of (say) 'and', does not necessarily agree with the meaning of the boolean operator *AND*. Das-Gupta [5] offers interesting examples that convince one of this, comparing for example the two natural language statements "I am interested in computers and education" and "I am interested in compilers and interpreters."

2.1 Criticism of the signal-detection paradigm

We note briefly here that despite the *mathematical* simplicity and apparent generality of the signal-detection paradigm, and its probabilistic model, its pre-mathematical foundations can be criticised. It seems a very strong assumption indeed that the recorded knowledge that informs a person who inspects ('reads', etc) a document is uninfluenced by that person's disposition (skills, interests), experience and memory, and awareness of contexts (social, personal, workplace) at the time of the search. In other words, it is questionable as to whether 'information' is something that can always be definable a priori, i.e. *prior to a search being undertaken.* If one is referring to a simple 'fact', e.g. whether each and every document in a database does, or does not, have a picture of a 5-spot ladybird in it, then that is reasonable enough. The metaphor of 'retrieval' in the sense of 'trawling for those documents that do have such a picture, is then an apt one. This indeed was the view of 'information' that dominated the design of the Cranfield experiments [6]. The concept of 'information' was then barely distinguishable from the notion of recorded knowledge. But if one accepts a more process-oriented view of information—of an entity that is defined by the human mind only at a point of use of recorded knowledge, then what is retrieved is *a set of documents* (or recorded knowledge), not information. The experimental procedure must then recognise that one can only observe those documents *within the set of retrieved documents* that inform the searcher. The values of r_i blithly defined in the model are *unknowable* in principle. The concept of information or relevance to hand is then one that cannot be separated from the cognition that 'owns' the need to be informed. It does not have a prior existence, but is rather created 'on the fly', at least in some cases. Informing, in other words, can plausibly be seen as an intelligent action of cognition representing an *interpretation* of recorded knowledge, rather than a more mechanical and passive process of 'assimilation'. A fuller argument is offered in [7], which labels the two paradigms of 'information retrieval' as 'CRAN' and 'SAR'[1]. We use this convention below, for brevity. Anecdotal (subjective) evidence suggests that both viewpoints may have their separate validities in everyday searching.

For the main purpose of this paper, as described in the next section, we will use data derived from an experiment undertaken under the CRAN paradigm. This is simply because of the absence (to the author's knowledge) of published data available from a SAR experiment. However, the two performance measures we will use are not the standard CRAN ones of P and R, but rather P and N, where N is the *number* of informing documents retrieved. N is obtainable under CRAN from R using N=R.n(A), where A is the set of a priori–informing documents.) Doing this allows us common ground with future 'SAR data', which of course recognises N as a count of an informing subset of the set of retrieved documents. In other words, the data we will describe, although obtained via a CRAN experiment, *could* have come from an SAR experiment, difficult though it is to design a convincing SAR experiment.[2]

[1] CRAN reflects the thinking behind the seminal Cranfield Experiments; SAR is for 'searcher adaptation and response'.

[2] In [7], the author suggests the use of a three-valued performance measure <P,R,N> in future experiments on informing processes, to respect the separate validities of the two paradigms.

3. Visual representations of n_i and p_i effects

The main suggestion of this paper is that the effects of the basic boolean operators as these are usually defined, i.e. as *AND*, *OR* and *AND NOT*, can usefully be visualised as bundles of directed line-segments on one or other outcome space. In view of the discussion in the above section, the two outcome spaces of central interest for this purpose are:

- The space determined by the pair of performance variables (r_i,p_i), central to the CRAN paradigm
- The space determined by the pair of performance variables (n_i,p_i), valid for either the CRAN or the SAR paradigm

We choose the second of these to illustrate these effects, in view of its greater generality.

Before proceeding we note that when a boolean expression retrieves no documents, we arbitrarily define the p_i value to be 0, and that we associate values $p_i = 0 = n_i$ with the empty or null boolean expression.

3.1 Definitions of directed-line segments and bundles

Basic definition of directed-line segments

Suppose that an initial, non-empty, boolean expression $\mathtt{E_i}$ generates a point (n_i,p_i). Then a successor expression to $\mathtt{E_i}$, say $\mathtt{E_j}$, which generates a point (n_j,p_j), shows a displacement represented by the directed-line segment:

$$((n_i,p_i),(n_j,p_j))$$

The 'quality' or 'effectiveness' of $\mathtt{E_j}$ relative to its predecessor expression $\mathtt{E_i}$, is shown by this line and its three properties of location, magnitude and direction. (We could, of course, refer to 'the vector associated to this directed-line segment', but then we would lose both 'location' in the new effect, and its reference to the predecessor search expression.)

This device applies to any logical expression relative to its predecessor, including any initial expression $\mathtt{E_1}$ which could be said to have had the null expression as predecessor, i.e. the effectiveness of $\mathtt{E_1}$ will be the directed-line segment:

$$((0,0),(n_1,p_1))$$

Single-attribute and binary-attribute directed-line segments

Despite the above general definition, our interests are much more modest. We immediately restrict ourselves to predecessor expressions derived from the use of single-attribute variables, such as $\mathtt{t_1}$ or $\mathtt{t_2}$ in Section 2. Suppose that $\mathtt{E_1}$ has the form $\mathtt{t_1}$, for example. Then the effect of introducing a new boolean expression which *AND*s $\mathtt{t_1}$ with another variable based on a new attribute such as t_2, t_3, t_4, t_5, etc, i.e. the effect of defining the set of boolean expressions:

t_1 && t_2
t_1 && t_3
t_1 && t_4
t_1 && t_5

—will be to produce a bundle of lines emanating from (n_1,p_1), with end points (n_2,p_2) etc. Similarly, bundles will be defined using *OR* or *AND NOT* as the operator.

Statistics of binary-attribute bundles

Suppose that an intelligent system, using some algorithm A_k, prompting the searcher for a new binary expression E_j, based on the single-valued expression E_i, suggests the use of new attributes t_i, $i \in I$. Then the directed-line segment bundle generated by searching on each of the new variables now *AND*ed to t_1 (or *OR*ed, or *NOT*ed) may be said to have a *mean* directed-line segment, pivoting on (n_1,p_1), defined by:

$$((n_1,p_1),(\text{mean}(n_i),\text{mean}(p_i)))$$

As indicated above, our choice of n_i and p_i and the effectiveness variables of interest is arbitrary. An experimental regime using CRAN would see the key variables of interest as r_i and p_i.

Implicit logic structures in information system use

The above simple devices provide a novel, quantitative tool for illustrating the *different capacities* of an information system *and a given user* to respond to searching using different logical operators. Suppose we are interested in the relative performance of the information system in regard to the use of the *AND* and *OR* operators, for a given predecessor search. More specifically, suppose a medical searcher, within some testing environment (e.g. involving Medline), has searched for informing medical papers using the MeSH term:

NATRIURESIS

as well as *AND*ings and *OR*ings of this term with each of the four terms:

DESOXYCORTICOSTERONE
ALDOSTERONE
VASOPRESSIN
GLOMERULAR FILTRATION RATE

The *AND*ings will then determine one mean directed-line segment, L_{AND} say, and the *OR*ings will produce another, L_{OR} say. The directed line $L_{OR} - L_{AND}$, could then be said to show the general effect of his moving his logic preference from *OR* to *AND*, for (1) this database, (2) that starting term, and (3) the relevance-attribution behaviour that he showed during the study.

Further studies would then elucidate the sensitivity of results of this type to choice of starting term, and experimental subject.

The searcher may also be curious as to the effect of the *AND NOT* operator when used to limit the size of the set of retrieved papers. His or her intuition may suggest that this will reduce both N and P, and yet wonder why, if that is the case, '*NOT*' is used at all. Such questions are answered by an interesting mixture of tautological facts and empirical facts. The example below is suggestive in this regard.

4. Illustration of the above effects

4.1 Background to the example

Data was collected, now somewhat historic but suitable for illustrative purposes, as described in [8], under the CRAN paradigm, i.e. on the numbers of informing and non-informing documents that matched each ELC derived from an algorithmically-derived set of search terms, for a particular partitioning of the database. In greater detail:

1. The document database MEDLINE was partitioned into 'informing' and 'non-informing' documents using the following operational definitions: (1) an 'informing document' was one that the author of a particular medical-review paper *had cited in that paper* and which also was included in MEDLINE, and (2) 'non-informing documents' were all the other documents in the MEDLINE database, provided they were published within the time-scope of relevance specified by the author of the review paper.[3]
2. Search terms contributing to a set of usable ELVs were also obtained in a standardised manner, using two different algorithms. One used cluster analysis, with specified parameters, on the terms attached to informing documents, choosing the most shallowly clustered terms as search terms; the other ranked all terms attached to informing documents according to their relative frequency (i.e. 'frequency in informing documents' as against 'frequency in non-informing documents') and chose those terms that ranked high. In each method, *five* search terms were identified as candidate search terms.
3. Searching was restricted to *assigned-terms*, i.e. terms assigned by National Library of Medicine indexing staff using the MeSH vocabulary. The study arbitrarily ignored the appearances of search terms in title and abstract fields, so that conceivably the performance data obtained were 'pessimistic'.
4. For each ELC based on a set of five search terms, frequencies of informing and non-informing documents were identified. The latter frequency was estimated from a contemporary sample of 512444 MEDLINE records, and 'scaled up' according to the actual size of the database taking the author's specification of 'time scope' into account. This scaling-up process thus introduced both a sampling error, and a systematic error arising from the rounding of 'floating

[3] The latter information was obtained by correspondence with the author: e.g. if the author of a review paper reviewed all the known literature in his or her field published between 1965 and 1970, then the set of *non*-relevant documents of interest would be 'all records in MEDLINE that had been published from 1965 to 1970, discounted by the (relatively few) records included in the review paper itself and in MEDLINE.' The latter were of course the set of *relevant* documents of interest.

point' to 'integer' values.

This methodology was believed to generate data that was much more credible than (say) the original Cranfield data, even though it accepted the CRAN paradigm, insofar as (1) third-party 'relevance judges' were not involved, (2) the information need that led to the operational definitions of 'informing document' was a 'real world' need, and not an artificiality of experimentation isolated from 'actual' information-seeking behaviour, and (3) search terms were defined in an objective, standardised manner, not through subjective arbitration within an experiment.

4.2 An example of a database partitioning

In the example below, we consider just one partitioning of MEDLINE[4], for search terms derived using the relative frequency method. This method was found to give superior results to the clustering method. The search terms derived from the items cited by a sample review paper, and included in MEDLINE, were, in order of eligibility (most-eligible first): IRON ISOTOPES, ERYTHROPOIETIN/UR, ERYTHROPOIETIN/AN, ERYTHROPOIETIN/BI, and POLYCYTHEMIA/ET. The strings '/xx' appended to the latter terms are 'MeSH qualifiers'.[5]

An inventory of ELC frequency-pairs for these five terms (numbered as above, e.g. with $\mathbf{t}_2$ standing for term 'ERYTHROPOIETIN/UR') was then found (see Appendix 1.)[6]

In the sections below, we show how boolean operations between pairs of terms can be expressed on the *N-P* outcome space, for the data given in Table 1 In each case, we use the fact that the ELCs partition the sets concerned in arriving at the *N*- and *P*-values concerned.

The effect of single-term searching

Directed line segments (DLSs) for each of the search terms of the example are then

[4] The medical review paper concerned was: Fried, W. Erythropoietin and the kidney. *Nephron*, 15(3/5), 1975, 327-349. (This review paper attempted to be comprehensive against all relevant items published up to and including February 1974.)

[5] In one variant of the method of identifying search terms algorithmically, these qualifiers were recognised (so distinguishing terms) in another variant, these qualifiers were not recognised, i.e. search terms distinguished solely by these qualifiers were *OR*ed together into a confounded term.

[6] We note in passing that the classical 'signal detection model' of information receipt, characterised by two (usually) continuous distributions, is replaced here by two probability distributions defined over an (initially) *unordered set* of (discrete) constructs, the ELCs.[2] Ordering (strongly or weakly) these ELCs gives a modified signal-detection model. 'Document weighting' formulas define such orderings. Deviation from the classical signal-detection model is also evident in the very large 'spike' of probability attaching to the all-negated ELC, which classical 'continuous' variate models (usually based on twin Normal density functions) completely ignore.

the lines that run from the coordinate (N=0,P=0) to the coordinates shown in the following table, and in Figure 1.

Search term	*N*	*P*
t_1 IRON ISOTOPES	44	0.9167
t_2 ERYTHROPOIETIN/UR	16	0.1616
t_3 ERYTHROPOIETIN/AN	12	0.1600
t_4 ERYTHROPOIETIN/BI	35	0.1351
t_5 POLYCYTHEMIA/ET	18	0.0641

The mean DLS for this 'bundle' of DLSs (DLSB) is that obtained by taking the means of the initial N- and P-values (trivially (0,0)) and the above N-values and P-values. The mean effectiveness of these five search terms is thus the DLS:

((0,0.0000), (25,0.2875)).

We suggest that DLSs (and mean DLSs for DLSBs) should always be seen as conditioned by a predecessor search expression, to pave the way for using DLSs to portray sequences of search expressions in studies of 'heuristic', interactive document retrieval. The above mean DLS is conditioned by the null (i.e. empty) search expression.

The effect of *AND*ing between pairs of search terms

If we choose to follow the search expression t_1 with its *AND*ed variants, i.e. t_1 && t_2, etc, and similarly for *AND*ings that follow the other single-term searches, we define a DLSB that 'pivots' on each of the end-points of the DLSs in the above table. For example, following t_4 with its *AND*ed forms with the other terms gives the end-points shown in the table below. (The DLSs here are conditioned by t_4, i.e. each commence with the point (35,0.1351).)

Search expression	*N*	*P*
t_4 && t_1	17	1.0000
t_4 && t_2	6	0.6000
t_4 && t_3	4	1.0000
t_4 && t_5	8	0.5000

The mean DLS for this bundle of DLSs, conditioned on t_4, is thus:

((35,0.1351), (8.75,0.7750))

In contrast, had we defined the bundle of DLSs of interest to have been conditioned on the null search expression, the mean DLS would have been:

((0,0.0000), (8.75,0.7750))

Our other four search terms generate similar mean DLSs. These are, for all five DLSBs conditioned on the null search expression:

Search expression set defined by all four *AND*ings with t_x, where t_x is	Mean DLSB, conditioned on the null search expression
t_1	((0,0.0000), (8.25,1.0000))
t_2	((0,0.0000), (4.25,0.7750))
t_3	((0,0.0000), (3.00,0.8750))
t_4	((0,0.0000), (8.75,0.7750))
t_5	((0,0.0000), (5.75.0.6250))

The DLSB for *AND*ings with t_4, conditioned on t_4, is illustrated in Figure 2.. The mean DLSBs conditioned on t_x are obtained by substituting for the (0,0.000) coordinate as appropriate.

The effect of *OR*ing between pairs of search terms

The procedure of *OR*ing the single terms with each other generates DLSBs for each term. For example, for *OR*ings with t4, the end-points of the members of the DLSB are as shown in the following table:

Search expression	*N*	*P*
t_4 || t_1	62	0.2138
t_4 || t_2	45	0.1293
t_4 || t_3	43	0.1303
t_4 || t_5	45	0.0859

The mean DLS for this bundle of DLSs, conditioned on t_4, is thus:

((35,0.1351), (48.75,0.1398))

In contrast, had we defined the bundle of DLSs of interest to have been conditioned on the null search expression, the mean DLS would have been:

((0,0.0000), (48.75,0.1398))

Our other four search terms generate similar mean DLSs. These are, for all five DLSBs conditioned on the null search expression:

Search expression set defined by all four *OR*ings with t_x, where t_x is	Mean DLSB, conditioned on the null search expression
t_1	((0,0.0000), (56.00,0.3023))
t_2	((0,0.0000), (39.00,0.1873))
t_3	((0,0.0000), (37.25,0.2007))
t_4	((0,0.0000), (48.75,0.1398))
t_5	((0,0.0000), (39.00,0.1030))

The DLSB for *OR*ings with t_4, conditioned on t_4, is illustrated in Figure 3. The mean DLSBs conditioned on t_x are obtained by substituting for the (0,0.000) coordinate as appropriate.

The effect of *AND NOT* between pairs of search terms

The use of the *AND NOT* operator is interesting insofar as it clearly (obviously?) produces inferior search performance when the negated search term is a 'good' one. This immediately sugegsts that a 'downward point' DLS provides an operational definition of this term—assuming it is a useful one to have. For the search term t_4, for example, the DLSB defined by *AND NOT* linkages between it and t_1, t_2, t_3 **and** t_5 are as shown in the tables below:

Search expression	*N*	*P*
t_4 && $\neg t_1$	18	0.0744
t_4 && $\neg t_2$	29	0.1165
t_4 && $\neg t_3$	31	0.1216
t_4 && $\neg t_5$	27	0.1111

In visual terms, this DLSB, conditioned on t_4, is shown in Figure 4. Clearly, all of the search terms are 'good'.

Obviously one cannot generalise from this effect, and suggest that the use of the *AND NOT* operator should always be avoided, since the distributions of n_i and m_i frequencies for other selections of search terms may lead to improved *N* and *P*-values when one or other term is subjected to *AND NOT*. (Such distributions might be expected when, for example, someone was searching for literature on 'the healthy human body' and considered using the search term INJURY. Presumably using *AND NOT* INJURY will not degrade the search and may even improve it.

The fuller characterisation of *AND / OR / AND NOT* differences

The introduction of DLSs allows us to characterise a database in terms of its *differential responsiveness* to choice of operator. We note also that since the choice of search terms that are used to define the DLSB depends on our specifying either an algorithm for this purpose, or an experimental protocol (one where the searcher is asked to specify the most eligible search terms), measurements of differences in the effects of *AND* and *OR* logic are so dependent. The measurement of such differences is also dependent *on the choice of predecessor search expression.*

The following constructs suggest themselves for this purpose:

1. The DLS from the end-point of the mean of the DLSB under one operator (e.g. *AND*), to the end-point of the mean DLSB under another operator (e.g. *OR*).
2. The *length* of the above DLS.
3. The *ratio of the lengths* of the mean DLSs for each of the two DLSBs.

The author suggests that '3' should be the preferred construct when the effectiveness measures are (as here) *N* and *P*, rather than Recall and Precision. Such a choice prevents the differential responsiveness of the database to be dominated by a frequency variable, which naturally extends over a much larger range than the Precision. (The former maps into $(0,K)$, where *K* could be of the order of 10^2 or 10^3, whereas, of course *P* maps into $[0,1]$.

Choosing our operators to be AND and OR, for illustration, the relevant data are as shown in the following table:

Length of mean DLS for the DLSB defined by 't_1 && t_x' conditioned on t_1	35.75
Length of mean DLS for the DLSB defined by 't_1 || t_x' conditioned on t_1	12.02
Ratio of lengths of two mean DLSs conditioned on t_1	2.98
Length of mean DLS for the DLSB defined by 't_2 && t_x' conditioned on t_2	11.77
Length of mean DLS for the DLSB defined by 't_2 || t_x' conditioned on t_2	23.00
Ratio of lengths of two mean DLSs conditioned on t_2	0.51
Length of mean DLS for the DLSB defined by 't_3 && t_x' conditioned on t_3	9.03
Length of mean DLS for the DLSB defined by 't_3 || t_x' conditioned on t_3	25.25
Ratio of lengths of two mean DLSs conditioned on t_3	0.36
Length of mean DLS for the DLSB defined by 't_4 && t_x' conditioned on t_4	26.26
Length of mean DLS for the DLSB defined by 't_4 || t_x' conditioned on t_4	13.75
Ratio of lengths of two mean DLSs conditioned on t_4	1.91
Length of mean DLS for the DLSB defined by 't_5 && t_x' conditioned on t_5	12.26
Length of mean DLS for the DLSB defined by 't_5 || t_x' conditioned on t_5	21.00
Ratio of lengths of two mean DLSs conditioned on t_5	0.58
MEAN VALUE OF LENGTH RATIOS:	1.27

At a rough, intuitive level, we might interpret the above to a searcher of MEDLINE, as follows: that on this evidence "*AND*ing and *OR*ing of term pairs have roughly the same *quantitative* effect, as DLSs on the *N-P* outcome space." We would also of course, at the same time, remind the searcher of the different *directions* of the DLSs, as shown by Figures 1 and 2.

One interesting, but provisional hypothesis, follows when we recall that the terms t_1 to t_5 are ranked in order of decreasing relative frequency in the informing set and the non-informing set. This is that the differences in the aggregate effects of *AND* and *OR* do not appear to depend significantly on the relative frequencies of search terms. A fuller statistical analysis of all 60 combinations of database partitioning and search-term set would be needed to investigate this conjecture more throroughly, but as mentioned above, it seems preferable to generate fresh experimental data from a more fully thought-through experimental regime.

Further constructs and investigations

The data to hand also allow us to evaluate an *AND/AND NOT* metric, and also an *OR/AND NOT* metric, analogous to the above *AND/OR* metric, and so obtain a more complete characterisation of MEDLINE/searcher interaction. The vector sum of the three mean vectors associated to the mean DLSs of each search term, when conditioned by the null search expression, offers a further construct that might be said to define the 'logical compactness' of that interaction, for each term. See Figure 5.

Investigation of metrics that involve the *AND NOT* operator may give helpful insights into the value to the searcher of specifying search terms that are (allegedly) useful as 'negative discriminators'. Anecdotal evidence suggests that searchers tend to specify search terms that are 'relevant' and deserving of *AND*ing or *OR*ing operations, rather than 'non-relevant' and deserving of negation. The large number of potentially useful negatively discriminating terms will presumably limit their value in retrieval, but adjudication by experiment rather than conjecture seems essential.

What appears to be equally necessary is a re-working of the approach in this paper, using term-document and term-query matrices

Comment by logic theoreticians would be helpful on the effect of the commutative property of *AND* and *OR* on the mean values of DLSs derived from different DLSBs (i.e. for DLSBs conditioned by different terms), and on the effect (if any) on 'logical compactness' of choosing sets of logical operators other than *AND*, *OR* and *AND NOT*.

It would also appear to be useful to attempt to link the approach given here with (1) those contributions to the 'IR' literature that focus on the 'logic of interaction' and 'logical imaging' studies, e.g. [9-10], (2) an analysis of the recent contribution on vector bundles by Lukes and Mishchenko [11], for its potential contributions in this area, and (3) an analysis of classical probability texts for their approaches to characterising random vectors and hence application to the current field.

However, the most valuable next steps, in the author's opinion, would be a formal analysis of the relationships between (1) the inequalities that attach to the (discrete) Swetsian frequencies, i.e. the two ELC frequency distributions as illustrated in Appendix 1, and (2) the extent and direction of the DLS metrics we have described.

References

1. Swets, J.A. Effectiveness of information retrieval methods. *American Documentation,* **20**, 1969, 72-89.
2. Heine, M. H. Information retrieval from classical databases from a signal-detection standpoint. *Information Technology: Research and Development,* **3**, 1984, 95-112.
3. Heine, M. H. The inverse relationship of precision and recall in terms of the Swets model. *Journal of Documentation,* 29, 1973, 81-84.
4. Bookstein, A. The anomalous behaviour of precision in the Swets model and its resolution. *Journal of Documentation,* 30, 1974, 374-380.
5. Das-Gupta, P. Boolean interpretation of conjunctions for document retrieval, *Journal of the American Society for Information Science,* 38, 1987, 245-254.
6. Cleverdon, C. W., Mills, J., Keen, M. *Factors determining the performance of indexing systems.* Cranfield, England, Aslib Cranfield Project, 1966. 2 vols.
7. Heine, M. H. Time to dump '*P* and *R*'? Mira '99: Final Mira Conference on Information Retrieval Evaluation [*Proceedings*], Glasgow, April 1999, pp.61-74; revised version to be published on the British Computer Society Electronic Workshops in Computing (eWIC) Web site, 1999: www.ewic.org.uk/ewic/.

8. Heine, M. H. An investigation of the relative influences of database informativeness, query size and query term specificity on the effectiveness of Medline searching, *Journal of Information Science,* **21**, 1995, 173-185

9. Crestani, F., van Rijsbergen, C.J. Probability kinematics in information retrieval. In: *SIGIR '95: Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Seattle, 9-13 July 1995, edited by Edward A. Fox, Peter Ingwersen and Raya Fidel.* ACM Press, 1995. pp291-299.

10. van Rijsbergen, C. J. Information, logic and uncertainty in information science. In: *Proceedings CoLIS 2: Second International Conference on Conceptions of Library and Information Science: Integration in Perspective October 13-16, 1996, edited by Peter Ingwersen and Niels Ole Pors.* Royal School,of Librarianship, Copenhagen, 1996. pp1-10.

11. Luke, G., Mishchenko, A.S. *Vector bundles and their applications.* Dordrecht, Kluwer, 1998.

Appendix 1

Arbit-rary label	**ELCs**	Number of informing documents evaluating the ELC to True, n_i	Number of non-informing documents evaluating the ELC to True, m_i
1	t_1 && t_2 && t_3 && t_4 && t_5	0	0
2	t_1 && t_2 && t_3 && t_4 && $\neg t\ _5$	0	0
3	t_1 && t_2 && t_3 && $\neg t_4$ && t_5	0	0
4	t_1 && t_2 && t_3 && $\neg t_4$ && $\neg t_5$	0	0
5	t_1 && t_2 && $\neg t_3$ && t_4 && t_5	1	0
6	t_1 && t_2 && $\neg t_3$ && t_4 && $\neg t_5$	1	0
7	t_1 && t_2 && $\neg t_3$ && $\neg t_4$ && t_5	0	0
8	t_1 && t_2 && $\neg t_3$ && $\neg t_4$ && $\neg t_5$	4	0
9	t_1 && $\neg t_2$ && t_3 && t_4 && t_5	1	0
10	t_1 && $\neg t_2$ && t_3 && t_4 && $\neg t_5$	1	0
11	t_1 && $\neg t_2$ && t_3 && $\neg t_4$ && t_5	0	0
12	t_1 && $\neg t_2$ && t_3 && $\neg t_4$ && $\neg t_5$	1	0
13	t_1 && $\neg t_2$ && $\neg t_3$ && t_4 && t_5	1	0
14	t_1 && $\neg t_2$ && $\neg t_3$ && t_4 && $\neg t_5$	12	0
15	t_1 && $\neg t_2$ && $\neg t_3$ && $\neg t_4$ && t_5	4	0
16	t_1 && $\neg t_2$ && $\neg t_3$ && $\neg t_4$ && $\neg t_5$	18	4
17	$\neg t_1$ && t_2 && t_3 && t_4 && t_5	0	0
18	$\neg t_1$ && t_2 && t_3 && t_4 && $\neg t_5$	0	0
19	$\neg t_1$ && t_2 && t_3 && $\neg t_4$ && t_5	0	0
20	$\neg t_1$ && t_2 && t_3 && $\neg t_4$ && $\neg t_5$	1	0
21	$\neg t_1$ && t_2 && $\neg t_3$ && t_4 && t_5	2	0
22	$\neg t_1$ && t_2 && $\neg t_3$ && t_4 && $\neg t_5$	2	4
23	$\neg t_1$ && t_2 && $\neg t_3$ && $\neg t_4$ && t_5	1	4
24	$\neg t_1$ && t_2 && $\neg t_3$ && $\neg t_4$ && $\neg t_5$	4	75
25	$\neg t_1$ && $\neg t_2$ && t_3 && t_4 && t_5	1	0
26	$\neg t_1$ && $\neg t_2$ && t_3 && t_4 && $\neg t_5$	1	0
27	$\neg t_1$ && $\neg t_2$ && t_3 && $\neg t_4$ && t_5	2	4
28	$\neg t_1$ && $\neg t_2$ && t_3 && $\neg t_4$ && $\neg t_5$	4	59
29	$\neg t_1$ && $\neg t_2$ && $\neg t_3$ && t_4 && t_5	2	8
30	$\neg t_1$ && $\neg t_2$ && $\neg t_3$ && t_4 && $\neg t_5$	10	212
31	$\neg t_1$ && $\neg t_2$ && $\neg t_3$ && $\neg t_4$ && t_5	3	247
32	$\neg t_1$ && $\neg t_2$ && $\neg t_3$ && $\neg t_4$ && $\neg t_5$	30	2009599

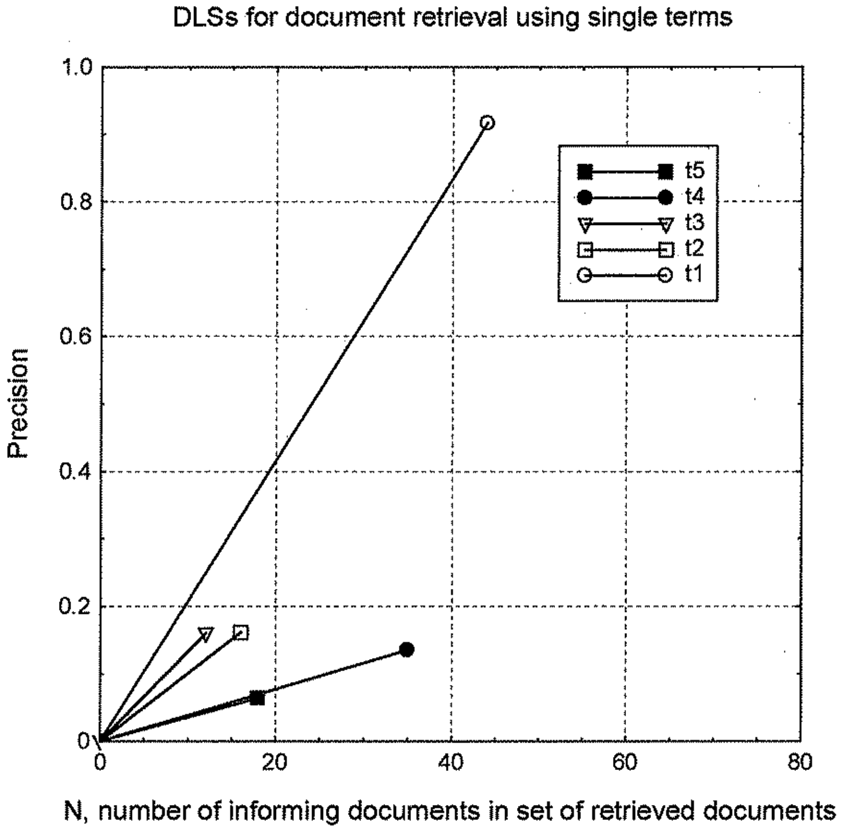


Fig. 1. DLSs for single term searches. The directions of the lines are outwards from the origin.

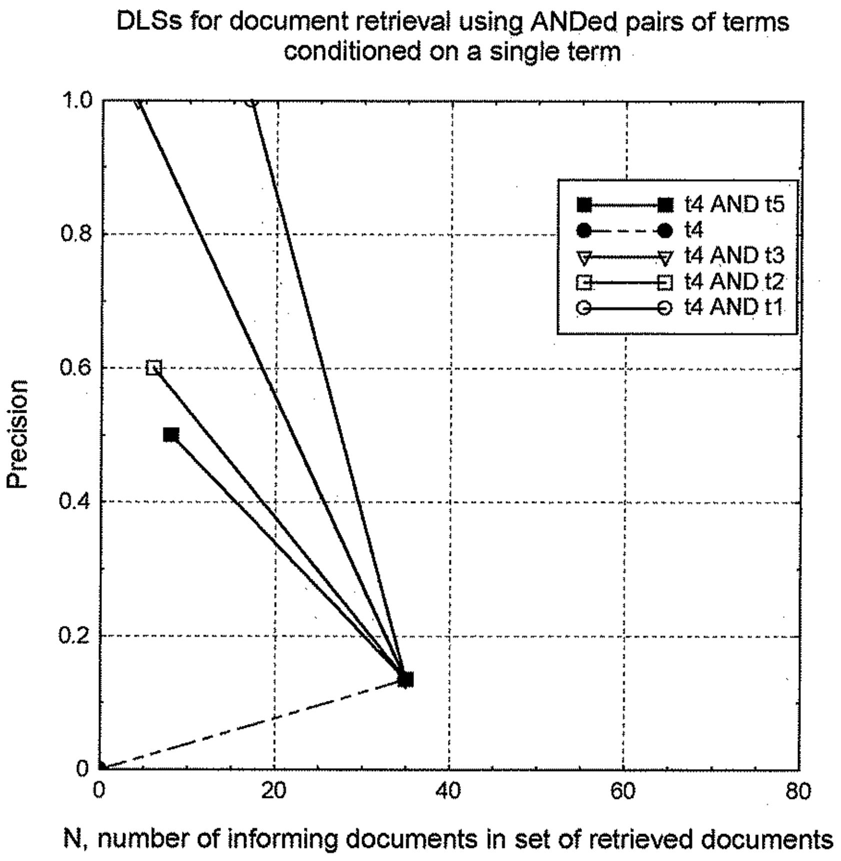


Fig. 2. DLSs for *AND*ed-pair searches conditioned by a prior single term search (**t**$_4$). The lines commence at the point (35,0.1351).

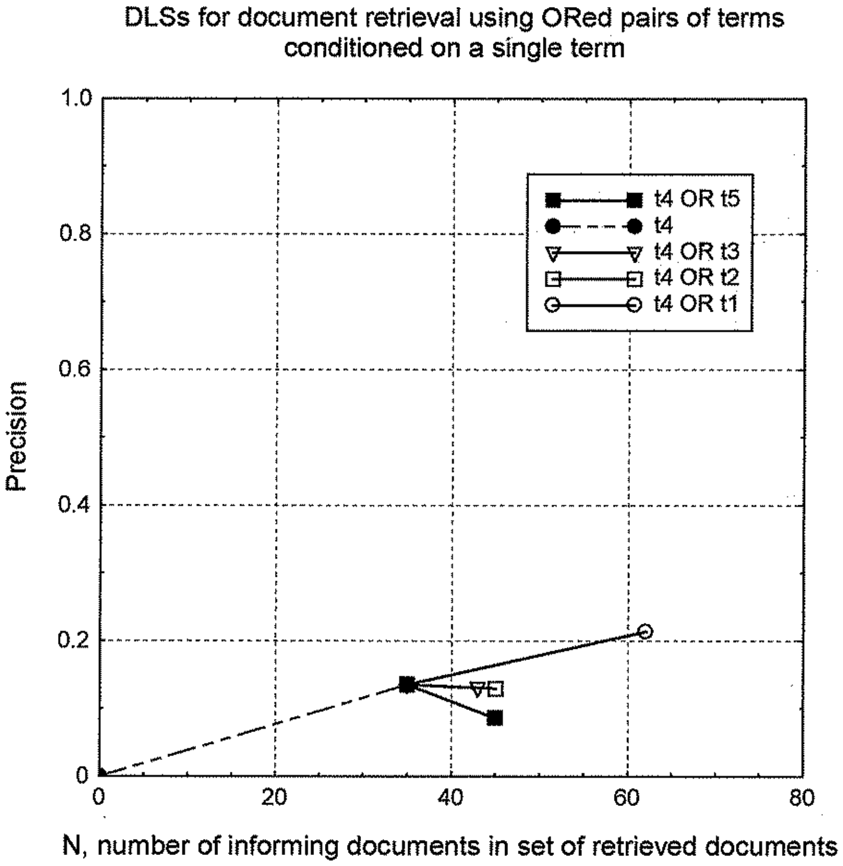


Fig. 3. DLSs for *OR*ed pair searches conditioned by a prior single term search (**t**$_4$). The lines commence at the point (35,0.1351).

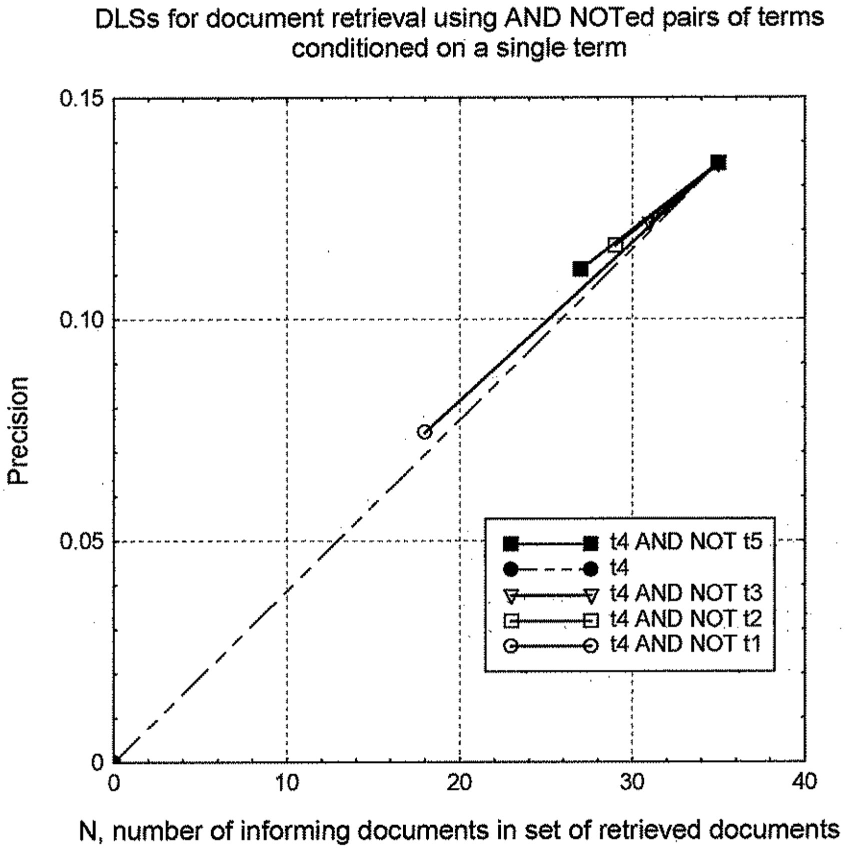


Fig. 4. DLSs for *AND NOT* paired searches conditioned by a prior single term search (**t_4**). The lines commence at the point (35,0.1351), i.e. are towards the origin in each case. The terms are all 'good'.

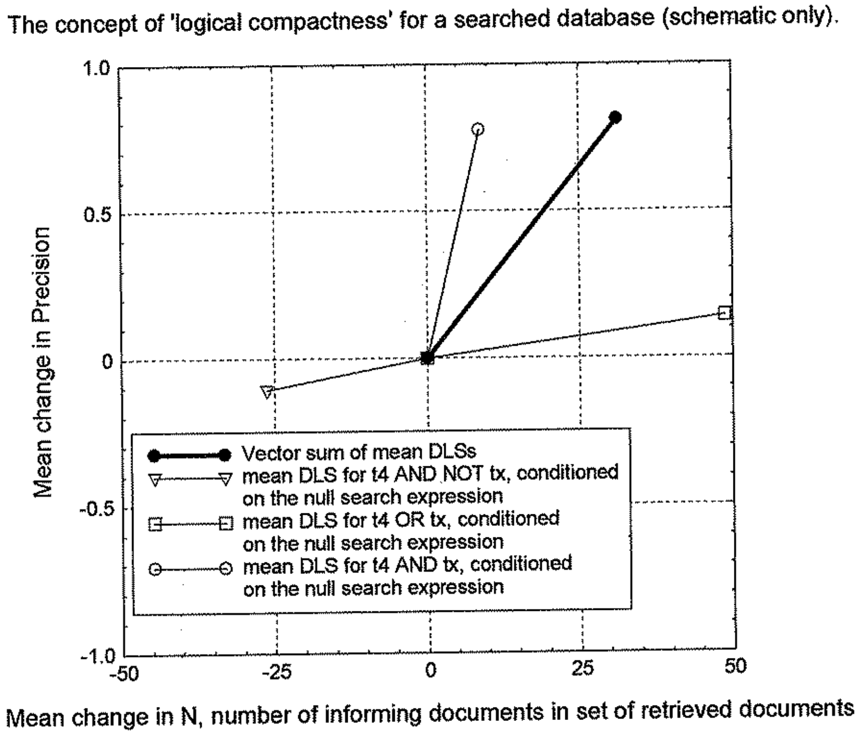


Fig. 5. Illustration of the concept of 'logical compactness' for a searched data-base. This is the sum of the vectors associated to the three mean DLSs formed by *AND*ing, *OR*ing and *AND NOT*ing all pairs of search terms when conditioned on the null search expression. The data shown are appropriate to $\mathbf{t}_4$ when logically connected to $\mathbf{t}_1$, $\mathbf{t}_2$, $\mathbf{t}_3$ and $\mathbf{t}_5$. The lines commence at (0,0) in each case.

Logic as a Tool in a Term Matching Information Retrieval System

Justin Picard

Université de Neuchâtel
Institut Interfacultaire d'Informatique
Pierre-à-Mazel 7, 2000 Neuchâtel
Switzerland
Justin.Picard@seco.unine.ch

Abstract. Information retrieval can be seen both as an inference process under uncertainty involving complex relationships between information items, and as a task of proper assessment of uncertainty. Probabilistic argumentation systems are a technique for reasoning under uncertainty which emphasize both aspects, by clearly distinguishing the qualitative and quantitative aspects of uncertainty. This paper presents the use of probabilistic argumentation systems for (1) taking into account hypertext links in order to improve an initial ranking of documents and (2) considering statistical similarities between query terms to improve their weighting. These two applications can be easily integrated in a retrieval system based on term matching.

1 Introduction

Current information retrieval (IR) systems are usually based on a simple representation of information in which documents and queries are represented by sets of keywords with associated weights. Retrieval is done by ranking documents in decreasing degree of match of their representation to the query representation. Beside its simplicity and computational efficiency, the popularity of this approach to IR can be attributed to the great amount of work spent in developing appropriate weighting schemes, often based on the statistical properties of text [19]. Recent empirical and theoretical works [8, 17] seem to show that lots of progresses can still be done in finding a probabilistic weighting scheme which fits best data. Besides, no approach based on a more sophisticated representation of information can claim in a significant way to yield better retrieval effectiveness on large test collections.

However, one basic assumption of probabilistic models is that terms and documents must be independent (although attempts have been made to get rid of this constraint, see [22]). But investigations have demonstrated the potential usefulness of incorporating dependencies or relationships between terms [18] and between documents [20]. When attempting to incorporate these new features in the matching process, the limits of the traditional term matching approach appear more clearly, and one is often to the use of ad-hoc schemes. For this reason and others, it has been argued by many authors that the use of an appropriate logic is the way to model the information retrieval process [13]. If the expressiveness of logic makes it a very attractive framework for knowledge-based IR [6], multimedia IR [7,14], and for representing structured documents [12], implementations of large-scale information retrieval systems based on logic have been relatively rare until now, to the exception of [4].

It is probable that logic-based representations of information will progressively replace keywords, at least for precise applications. But we believe that logic can already be integrated in large-scale IR systems based on term matching, as a tool for solving specific problems which cannot be formalized within more conventional approaches. In this paper we propose to apply a logical approach to treat the problems of integrating (1) relationships between documents and (2) relationships between query terms inside a term matching IR system. The different steps of the retrieval process are done as usual, and logic can be seen as a tool which modifies the output of certain components of the retrieval system. The "logical components" can be integrated to any retrieval system working on soft term matching, e.g. the vector-space or probabilistic models.

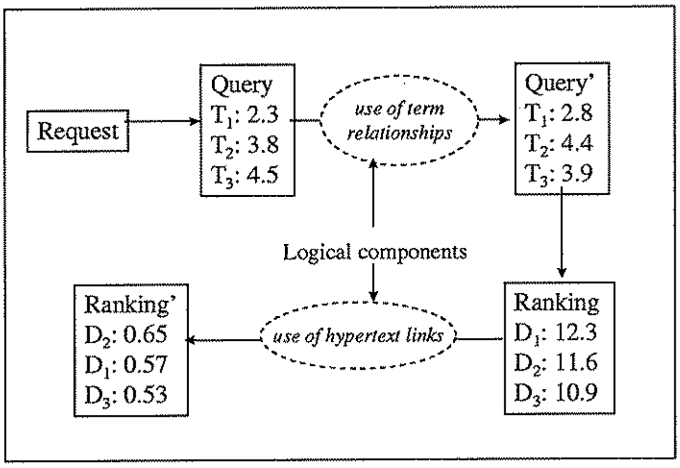


Fig. 1. Positions of logical components in the retrieval system

As shown on Figure 1, the initial weights of the query terms are modified after consideration of their relationships. The idea is to raise the weights of the terms which are expected to be representative of the information need, and to lower the weight of those which are not. Then the matching process is done as usual and a ranking of documents is handed out. This ranking is modified after consideration of the hypertext links between documents.

It might be useful to explain the reason behind the use of "relationships" between information items. We assume that documents and terms are binary variables: a document can be either relevant or non-relevant, and a query term can be either "content" or "noisy". A content term is representative of the information need, while a noisy is not. This last assumption will be justified later. There is a knowledge base of relationships which, in the case of documents, has been defined manually (e.g. hypertext or citation links). In the case of terms, it is created artificially by computing corpus similarities between terms. In any case, we assume and verify experimentally that these relationships have an influence on the state of documents or terms.

The modeling is done with probabilistic argumentation systems (PAS) [11]. PAS is a technique for reasoning under uncertainty which treats inference as a process of finding and assessing symbolic arguments in favor or against a certain hypothesis. PAS represent uncertainty in a clear and easily understandable way: the qualitative part is handled with propositional logic and the quantitative part is treated with probability theory. We will see that PAS offer a natural way to model relationships between terms and between documents, and allows complex inferences. Though a logical approach is a priori taken, we will insist throughout this paper on the importance of assessing precisely the uncertainty.

Next section presents shortly PAS, in the perspective of their use for handling relationships between variables. Section 3 develops the model of hypertext information retrieval. Section 4 treats the problem of determining the useful query terms from the use of relationships between terms.

2 Probabilistic Argumentation Systems

2.1 An introduction to PAS

To represent and handle uncertainty, PAS are based on a combination of propositional logic and probability theory. Apparently, propositional logic cannot include uncertainty in facts such as 'P_1 is true' (denoted P_1), or rules such as 'P_1 entails P_2' (denoted $P_1 \rightarrow P_2$). But there is a simple way to incorporate uncertainty in such statements by using a specific type of propositional symbol called **assumption** to represent uncertainty. With assumptions, facts and rules become true under the specific condition that certain assumptions are true/false. For example, with the use of assumptions the preceding statements would become 'if assumption a_1 is true then P_1 is true', and 'if assumption a_{12} is true then P_1 entails P_2', denoted respectively: $a_1 \rightarrow P_1$ and $a_{12} \rightarrow (P_1 \rightarrow P_2)$. An assumption is a then a proposition which represents the situation

A triple (P, A, S), where $P = \{P_1, ..., P_N\}$ is the set of propositions representing the N variables of interest, $A = \{a_1, ..., a_M\}$ the set of propositions (called assumptions) used for representing the uncertainty, and $\Sigma = \{\xi_1, ..., \xi_R\}$ a set of facts and rules on literals from A and P, is called a **(propositional)**

argumentation system [9]. A propositional argumentation system handles the qualitative part of the inference process.[1]

When knowledge is modeled in a propositional argumentation system, arguments can be inferred from the knowledge base supporting or discounting certain hypotheses. A **hypothesis** h is any logical formula with symbols in $A \cup P$. An **argument** in favor of h is a conjunction a of literals of assumptions for which h can be logically deduced from the given knowledge Σ. In other words, a is an argument for h if: $a, \Sigma \models h$. Then, the hypothesis is said to be supported by a. The **quasi-support** of h is defined as the set (or the disjunction) of all minimal supporting arguments for h. $QS(h, \Sigma)$ denotes the quasi-support if it is considered as a set, and $qs(h, \Sigma)$ denotes the corresponding disjunction. The term "quasi" expresses the fact that some of the supporting arguments for h may be in contradiction with the given knowledge, i.e. $a, \Sigma \models \perp$. $QS(\perp, \Sigma)$ is the set of all minimal arguments which are in contradiction with Σ.

So far, hypotheses are only judged qualitatively. A quantitative judgement of the situation is possible if probabilities are assigned to the assumptions, e.g. $p(a_1) = x_1, p(a_2) = x_2$, etc. Suppose that the assumption are mutually independent, e.g. $p(a_1 \wedge a_2) = p(a_1) \cdot p(a_2)$. Let X be the set of probabilities assigned, then (A, P, Σ, X) is called a **probabilistic argumentation system.** If h is the hypothesis of interest, then the conditional probability:

$$\begin{aligned} dsp(h, \Sigma) &= P(qs(h, \Sigma) | \neg qs(\perp, \Sigma)) \\ &= \frac{P(qs(h, \Sigma)) - P(qs(\perp, \Sigma))}{1 - P(qs(\perp, \Sigma))} \end{aligned} \qquad (1)$$

is called **degree of support** for h. It is a value between 0 and 1 that represents the support or the belief that h is true in the light of the given knowledge. Clearly, $dsp(h, \Sigma) = 1$ means that h is certainly true, while $dsp(h, \Sigma) = 0$ means that h is certainly false.

Computing degrees of support involves 2 major steps:

1. determine $qs(h, \Sigma)$ and $qs(\perp, \Sigma)$;
2. evaluate $P(qs(h, \Sigma))$ and $P(qs(\perp, \Sigma))$, and apply Equation (1).

The details of step 1 are not discussed in this paper, but the reader is referred to [1, 9]. Step 2 can be solved by computing disjoint forms of $qs(h, \Sigma)$ and $qs(\perp, \Sigma)$. Then, the probability of a disjoint form is simply the sum of the probabilities of the disjoint terms [11].

2.2 Handling relationships between binary variables

In the applications presented in this paper, PAS serve to handle the situation where one has some a priori probability on the state of each of a set of binary

[1] In this text, capital letters are used for propositions, and minor letters for assumptions.

variables represented by a set of propositions $\{P_1, ..., P_N\}$, and would like to improve this a priori knowledge by taking into account some relationships between the variables. This a priori knowledge is symbolically modeled by:

$$a_1 \to P_1, ..., a_N \to P_N \tag{2}$$

with $p(a_1), ..., p(a_N)$ being the a priori probabilities of P_1 to P_N.

The type of relationships which is considered here is limited to single variables. A positive influence of P_i on P_j (P_i is positive evidence for P_j) is modeled by:

$$l_{ij} \to (P_i \to P_j) \Leftrightarrow P_i \wedge l_{ij} \to P_j \tag{3}$$

Here, l_{ij} is a proposition (assumption) representing the situations where $P_i \to P_j$, and $p(l_{ij})$ is the probability that these situations occur. In this text, we prefer using the equivalent latter notation $P_i \wedge l_{ij} \to P_j$. The same way, P_i can be negative evidence for P_j:

$$P_i \wedge a_{ij} \to \neg P_j \tag{4}$$

In the IR applications treated here, Equations (2), (3) and (4) are sufficient for modeling the knowledge. The fundamental problem of assessing the probabilities of assumptions will be discussed separately for each application. Because of the simplicity of the rules, we use only a simple path-finding algorithm for finding the support of each P_i. For putting the support in disjoint form, we use an extension of Heidtmann's algorithm [10] which can be applied to any disjunctive normal form [2].

2.3 An example

The following example is the simplest case where there is positive and negative evidence towards a variable, and is pretty representative of the applications of PAS seen in this paper. Suppose we have a set of variables of interest represented by $P = \{P_1, P_2, P_3\}$, and some a priori probability on their state, i.e.: $p(P_1) = 0.1$, $p(P_2) = 0.2$, $p(P_3) = 0.3$. This is represented by:

$$\{a_1 \to P_1, a_2 \to P_2, a_3 \to P_3\} \tag{5}$$

with $p(a_1) = 0.1, p(a_2) = 0.2$, and $p(a_3) = 0.3$. We incorporate the knowledge that P_1 is positive evidence for P_2 with probability 0.5, and P_3 is negative evidence for P_2 with probability 0.4:

$$P_1 \wedge l_{12} \to P_2, P_3 \wedge l_{32} \to \neg P_2 \tag{6}$$

with $p(l_{12}) = 0.5$ and $p(l_{32}) = 0.4$. Figure 2 shows a graphical representation of this PAS.

It might be interesting to see how these relationships modify our initial knowledge on P_2. From the knowledge base, there are two arguments in favor of P_2: a_2

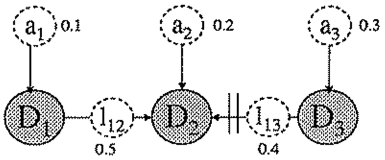


Fig. 2. Graphical representation of the PAS

and $a_1 \wedge l_{12}$. There is also an argument against P_2: $a_3 \wedge l_{32}$. The quasi-supports of P_2 and $\neg P_2$ are shown below in disjunctive form. $qs(P_2, \Sigma)$ is also shown in its equivalent disjoint form.

$$qs(P_2, \Sigma) = a_2 \vee (a_1 \wedge l_{12}) = a_2 \vee (a_1 \wedge l_{12} \wedge \neg a_2) \tag{7}$$

$$qs(\neg P_2, \Sigma) = a_3 \wedge l_{32} \tag{8}$$

There is a contradiction in the knowledge base if arguments for P_2 and $\neg P_2$ are true at the same time. Then the quasi-support of the contradiction is, in disjoint form:

$$qs(\perp, \Sigma) = (a_3 \wedge l_{32}) \wedge (a_2 \vee (a_1 \wedge l_{12} \wedge \neg a_2)) \tag{9}$$

We are interested in having a numerical degree of support of P_2. We must first compute the probabilities of the quasi-supports of P_2 and of the contradiction:

$$P(qs(P_2, \Sigma)) = 0.2 + 0.1 \cdot 0.5 \cdot (1 - 0.2) = 0.24 \tag{10}$$

$$P(qs(\perp, \Sigma)) = 0.3 \cdot 0.4 \cdot (0.2 + 0.1 \cdot 0.5 \cdot (1 - 0.2)) = 0.288 \tag{11}$$

The degree of support is then found by application of Equation (1):

$$dsp(P_2, \Sigma) = \frac{0.24 - 0.288}{1 - 0.288} = 0.2175 \tag{12}$$

3 Using Hypertext Links for Improving the Ranking of Documents

3.1 The general idea

Many investigations have been done in information retrieval on the use of relationships between documents to automatically improve retrieval, e.g. [3, 21]. These relationships may exist naturally in the collection (e.g. citation links in a collection of scientific articles, hypertext links) or they can be artificially created between similar documents, but this latter information is less interesting because it is correlated to the indexing process. In the past, the interest in this source of evidence was limited by the scarcity of test collections including "natural" document relationships. But nowadays, with the growing importance of the Web and in general, of hypertext documents, there is an increasing number of people interested in mining this data for extracting knowledge. The growing interest of the IR community is illustrated by the arrival this year of a special web track in the *Text REtrieval Conference.* This track is specially interested in investigating to what degree links can be used to enhance retrieval.

The intuitive idea behind the use of hypertext links[2] for retrieval can be formulated this way:

- A link between two documents is evidence that (part of) their content is similar or related;
- Two similar or related documents are more likely to be found relevant to the same request;
- Consequently, if two documents are linked and one is relevant, this is evidence for the relevance of the other.

A technique often used in IR to take into account this information is spreading activation [21]. Simply stated, in this technique the score of a document handed out by the retrieval system is updated by the weighted score of its direct neighbors. This step can be repeated several times to take into account the effect of indirect neighbors, but this often leads to reduction of retrieval effectiveness [21]. In [15], we have proposed a formal model based on PAS for spreading hypertext information, which allows to spread evidence to indirect neighbors in a theoretically sound way. In the next sections, we present the qualitative and quantitative parts of the model, and show some experimental results on the CACM collection.

3.2 The qualitative part of the model

In this application of PAS, document relationships are used in the purpose of improving an initial ranking of documents. The interesting variables are the state of each document represented by the set of propositions $P = \{D_1, ...D_N\}$, where D_i means 'document D_i is relevant'. As is often done in IR, relevance is assumed to be binary.

The initial ranking and scores of documents can be seen as uncertain information on their state. As shown in Section 2.2, this a priori knowledge on the state of each document is represented by corresponding assumptions a_1 to a_N:

$$a_1 \rightarrow D_1, ..., a_N \rightarrow D_N \qquad (13)$$

This knowledge is modified when additional knowledge from the links between documents is incorporated. For each link from D_i to D_j, two corresponding

[2] From now on, we will use indifferently 'hypertext links' and 'document relationships'

uncertain rules are created:

$$D_i \wedge l_{ij} \to D_j \tag{14}$$
$$D_j \wedge l_{ji} \to D_i \tag{15}$$

The assumption l_{ji} represents the situation (or the set of necessary conditions) where the rule $D_i \to D_j$ applies. In that situation, there is a connection between an existing link from D_i to D_j and the relevance of the two documents. The same way, assumption l_{ji} represents the case where the link is followed backward.

In general, the probabilities of l_{ij} and l_{ji} are different. For each document, the support is found by a simple path-finding algorithm, as said before. Unless the hypertext structure of the collection is modified, the quasi-support and thus the degree of support of each document does not change, and can then be precomputed and stored. For a precise query, the only operation is to replace the probability of the assumptions by their numeric values, in the degree of support of each document. The computations on the CACM collection (3204 queries, 50 queries, 2721 links) took a few seconds, and on larger collections computational problems can be avoided by establishing a cutoff on the maximum inference chains allowed. Anyway, as the length of inference increases, the weight of the corresponding argument decreases exponentially, and our experiments show that arguments which include more than three link assumptions have an insignificant effect on retrieval effectiveness.

3.3 The quantitative part of the model

We have modeled the symbolic part of the knowledge, and have a scheme for making reliable inferences. We may now attack the fundamental problem of assessing the evidence contained in the hypertext links. This raises two questions:

- What is the meaning of the probabilities of the assumptions?
- How do we compute these probabilities?

Evidence handed out by the retrieval system. $a_i \to D_i$ represents symbolically the uncertain evidence on the relevance of D_i provided by the initial score s_i and rank r_i of D_i. If the document relationships were not considered, the degree of support of D_i would be: $dsp(D_i) = p(a_i)$. It is then natural to assess $P(a_i)$ by $p(D_i = relevant|score = s_i, rank = r_i)$. This probability can be assessed by fitting a logistic regression on a sample of training queries. In our experiments on the CACM collection, we used only the rank as explicating variable, because this feature provides more stable probabilities. The score may provide an interesting additional information, but the features are very highly correlated, since the rank is derived from the score. We do not insist too much on details, but it must be said that the correct estimation of these probabilities is crucial because the final ranking of documents is quite sensitive to these.

Fixed link probabilities. For a more detailed discussion of this subject, the reader is referred to [15]. To compute these probabilities, the general idea is that if there is a link between D_i and D_j, $p(l_{ij})$ can be estimated by $\widehat{p}(D_j|D_i, \emptyset)$, where $\emptyset$ means that there is no other known evidence for D_j. That is, we are estimating the probability that D_j is relevant, knowing that (1) D_i is relevant, that (2) there is a link from D_i to D_j and that (3) there is no link from another relevant document to D_j. This probability can be assessed on a set of training queries, by considering the documents which are linked to one and only one relevant document. If documents which are linked to more than one relevant document are considered, the estimate is biased upward because these documents have more than one source of evidence. This may appear to be only a supposition, but this behaviour is indeed observed: for the 'citing' link, the probability estimates are respectively $\frac{174}{658} \simeq 0.26$, $\frac{77}{143} \simeq 0.53$ and $\frac{28}{42} \simeq 0.67$, for documents linked to one, two and three relevant documents. It is interesting to observe that evidence seem to combine similarly to the way they are combined with PAS. Indeed, there would be no point in using a model which would not fit the behaviour of the data.

Variable link probabilities. Fixed probabilities are not very satisfying, because clearly some links are in general more appropriate than others, and links are more or less appropriate depending on the context (the context is the user's information need, represented by the query). We are presently adapting a technique developed by Frei and Stieger [5] which deals with the semantic of the links, to the computation of individual link probabilities. The computation of this probability should take into account two factors: (1) the general appropriateness of the link, and (2) its adequacy to the present query. Intuitively, the general appropriateness of a link between two documents D_1 and D_2 depends on their similarity $sim(D_1, D_2)$, which can be measured from their indexed representation. And its adequacy for a precise query depends on the similarity between the query and the link between D_1 and D_2, $sim((D_1, D_2), q)$.

In our experiments, links were represented by the ten most representative keywords of each of the documents. For each link from a relevant document to another document, the two cosine similarities are computed. As explained before, we consider only documents linked to one and only one relevant document. On Figure 3, the probability $p(D_j|D_i, \emptyset)$ is computed for four sets of binned similarities. Clearly, the probability of a link assumption varies with the static and dynamic value of a link. We are presently working on converting similarities to individual probability estimates. It seems that a linear model would fit quite correctly the data, but static and dynamic similarities are positively correlated, and this may lead to an overestimation of link probabilities if we are not aware of this.

3.4 Experiments on the CACM collection

We are presently implementing our model on the Trec Web Track (2.1 Gbytes), and in the next months we will be able to show experimental results on this col-

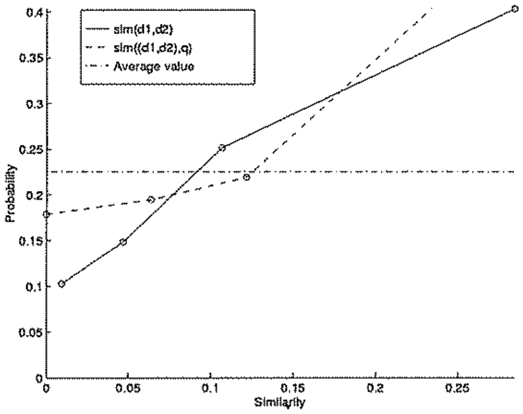


Fig. 3. $P(D_j|D_i,\emptyset)$ vs $sim(D_1,D_2)$ and $sim((D_1,D_2),q)$. ”Cited” links

lection. For now we may present some results obtained on the CACM collection, which were already shown in [15] . The set of preliminary operations, illustrated on Figure 4, is as follows:

- Convert each bibliographic link from D_i to D_j to Equation (14).
- For each document D_i, compute the support $sp(D_i, \Sigma)$.
- Put the support of each document in disjoint form using Heidtmann's algorithm [10]. Convert this disjoint form to a directly usable formula for computing the degree of support $dsp(D_i, \Sigma)$.
- Compute the probability that a retrieved document is relevant given its rank, by fitting a logistic regression.
- Compute the probability of each type of link (fixed link probability) or of each link (variable link probability) using the technique described in Section 3.3.

At this point, all the logical operations are done. When a query is processed, the only operation (if the link probabilities are fixed) is to compute the degree of support of each document $dsp(D_i, \Sigma)$ by assigning values to the probabilities of the a priori assumptions $p(a_i)$. Documents are then re-ranked by decreasing degree of support.

Table 1. Preliminary results on the CACM collection. SA: spreading activation

Type of link	SA	PAS	PAS vs baseline	PAS vs SA
Citing	26.66 (0.3)	26.68	+5.57%	0.00%
Cited	26.13 (0.4)	27.25	+7.83%	+4.11%

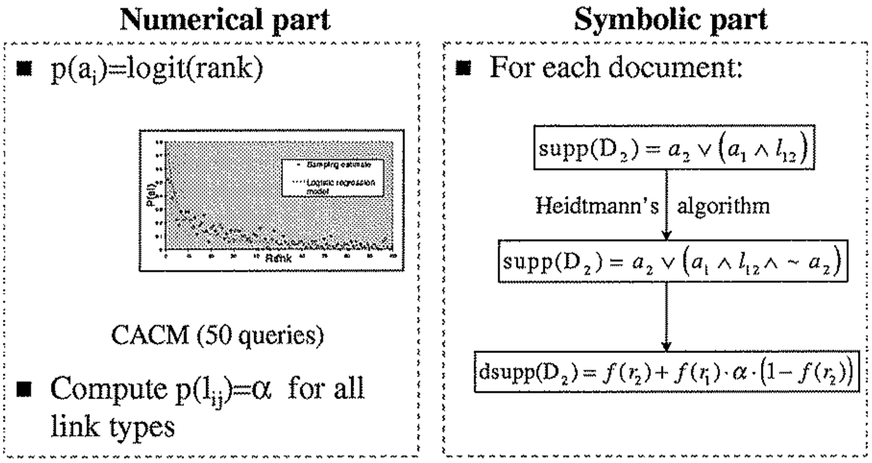

Numerical part
■ $p(a_i)$=logit(rank)
Sampling estimate
Logistic regression model
P(ai)
Rank
CACM (50 queries)
■ Compute $p(l_{ij})=\alpha$ for all link types
Symbolic part
■ For each document:
$\mathrm{supp}(D_2) = a_2 \vee (a_1 \wedge l_{12})$
Heidtmann's algorithm
$\mathrm{supp}(D_2) = a_2 \vee (a_1 \wedge l_{12} \wedge \sim a_2)$
$\mathrm{dsupp}(D_2) = f(r_2) + f(r_1) \cdot \alpha \cdot (1 - f(r_2))$

Fig. 4. Set of preliminary operations.

The basic retrieval process was done using a classical retrieval system based on the cosine similarity measure. The baseline retrieval effectiveness at 11 recall point was 25.27%, with the TRECeval software. Comparisons were made between PAS and baseline, and PAS and spreading activation. For spreading activation, different values of parameters were tried, and the retrieval effectiveness shown here is best one obtained. Citing and cited links were treated separately, and the longest inferences allowed were of length 3. The results are shown on Table 1.

4 Determining Good Descriptors of the User's Information Need

4.1 The general idea

IR systems based on term matching often extract the terms describing the information need from a natural language request. The statistical characteristics of each of these terms, such as their number of occurrence in the query and their document frequency in the collection, serve to weight their relative importance in the matching formula. Terms are then described only by abstract features, and their real information content relatively to the information need is often not considered. But obviously, some terms describe much better the information need than others, and are more likely to be found in relevant documents. For example, take a request such as:

```
Document will provide totals or specific data on changes to the
proven reserve figures for any oil or natural gas producer.
```

The only terms found in relevant documents are:`gas, oil, reserve`. This means that all the other terms are harmful to retrieval because they retrieve only non-relevant documents, which decreases precision. Grossly, query terms can be divided in "content" terms which are useful for retrieval, and "noisy" terms which are harmful to retrieval. A direct way to test if a term is useful to retrieval is to compare the retrieval effectiveness of the query with and without the term.

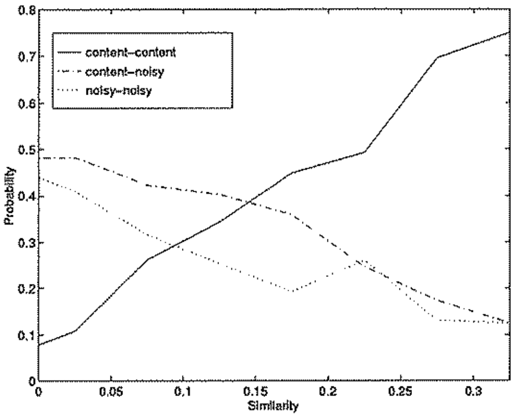


Fig. 5. Probability of each class of term pairs vs Cosine similarity

But is it possible to predict reasonably well which are the "content" terms of a query? Recently, we have stated and verified the hypothesis that content terms of a query tend to be more similar to each other than to other query terms [16]. This can be intuitively justified, because content terms often concern similar topics (e.g. 'gas' and 'oil'), and should then be found in similar contexts in the corpus. Consequently, a similarity measure should be higher on average for content terms. Figure 5 shows the probability to find each of the classes of term pairs (content-content, content-noisy and noisy-noisy) for different values of the (cosine) first order co-occurrence similarity (a similar pattern is observed for second order co-occurrence). Content terms are determined by a χ_2 test of independence between document relevance and occurrence of the query term, at the 95% confidence level. The similarities are binned in intervals of 0.05. As similarity increases the probability of having two content terms increases, and clearly this is an indicator of the usefulness of query terms. For more details about this experiment, the reader is referred to [16].

4.2 Predicting content terms: modeling with PAS

Even if similarities can be evidence for the "state" of the query terms, this pattern is not so easy to use. For n query terms, there are $\frac{n(n-1)}{2}$ similarity values to consider. All query terms are "related" by their similarity value, but the state of each one is unknown. PAS may help us here: we are in the case where each variable has an a priori probability of being in a certain state (45% of the query terms are content terms in the CACM), and we wish to exploit relationships between terms to improve this knowledge. A priori knowledge is modeled by:

$$a_1 \rightarrow T_1, ..., a_N \rightarrow T_N \tag{16}$$

where T_i ($\neg T_i$) means "T_i is a content (noisy) term".

then similarity values are converted to .relationships. A high similarity between T_i and T_j is converted to:

$$T_i \wedge l_{ij} \rightarrow T_j \tag{17}$$

It is to be noted that there is not necessary a symmetric rule $T_j \wedge l_{ji} \rightarrow T_i$, or at least the probabilities $p(l_{ij})$ and $p(l_{ji}$ may not be equal. Conversely, a small similarity will be converted to a negative evidence:

$$T_i \wedge a_{ij} \rightarrow \neg T_j \tag{18}$$

The major problem is to decide when there is a relationship, and how to assign probabilities to relationships. On the CACM collection, the a priori probability of having the class content-content is 0.23, so any significant divergence from this value can be converted to a positive or negative evidence. For example, for a similarity of 0.25 the predicted probability of having class content-content is about 0.55 (see Fig. 5). This can be modeled by Equation (17) with probability $p(l_{ij}) = 0.55 - 0.23 = 0.32$.

4.3 Future research

Using ABEL, a simulation tool for PAS [3], we are presently testing queries of different sizes, with different positive and negative links. Of course, lots of experiments are necessary to understand how query terms "interact" when knowledge is modeled in a PAS, and how some improved knowledge on the state of the query terms can be converted to a proper modification of their weights. Additional sources of evidence such as relationships coming from a manual thesaurus would be worth being considered also.

But the challenge is worth it, because if it is possible, a correct prediction of the 'content' terms can be used in IR to improve query term weighting, to determine which terms can be used to expand the query with similar terms, and ultimately to have a better understanding of the user's information need. Our research objective is to fully test the ideas presented in this section on a subset of the TREC collection (300 Mbytes) by the end of the year.

[3] ABEL can be downloaded at:http://www2-iiuf.unifr.ch/tcs/ABEL/

5 Conclusion

Even though the symbolic part of uncertain knowledge is naturally modeled with PAS, the step of numerical assessment of probabilities is often a very difficult problem. Our experience in modeling the inference processes inherent to IR has led us to the conclusion that if the uncertainty is incorrectly assessed, combined or propagated, a logical formalism will very probably be unable to improve retrieval effectiveness. But at present time, logic is one of the most promising formalism for helping us learn what IR is all about. The two applications presented here highlight that in IR, the numerical and symbolic aspects of uncertainty are profoundly interlaced. It is our opinion that a purely symbolic or numerical approach would not bring the same insight in these problems. The theoretical foundations of PAS, which rely on the theory of evidence, make them a reliable technique for approaching problems in which the quantitative and qualitative aspects of uncertainty are of equal importance.

References

1. B. Anrig, R. Bissig, R. Haenni, J. Kohlas, and N. Lehmann. Probabilistic argumentation systems: Introduction to assumption-based modeling with ABEL. Technical Report 99-1, Institute of Informatics, University of Fribourg, 1999.
2. R. Bertschy and P.A. Monney. A generalization of the algorithm of heidtmann to non-monotone formulas. *Journal of Computational and Applied Mathematics*, 76:55–76, 1996.
3. W.B. Croft and H.R. Turtle. Retrieval strategies for hypertext. *Information Processing & Management*, 29(3):313–324, 1993.
4. M. Sanderson F. Crestani, I. Ruthven and C.J. van Rijsbergen. The troubles with using a logical model of ir on a large collection of documents. *TREC-4*, pages 509–526, 1996.
5. H.P. Frei and D.Stieger. The use of semantic links in hypertext information retrieval. *Information Processing & Management*, 31(1):1–13, 1995.
6. N. Fuhr. Probabilistic datalog- a logic for powerful retrieval models. In *Proc. of the Int. ACM-SIGIR Conf.*, pages 282–290, 1995.
7. N. Fuhr, N. Govert, and T. Rolleke. DOLORES: A system for logic-based retrieval of multimedia objects. In *Proc. of the Int. ACM-SIGIR Conf.*, pages 257–265, 1998.
8. W.R. Greiff. A theory of term weighting based on exploratory data analysis. In *Proc. of the Int. ACM-SIGIR Conf.*, pages 11–19, 1998.
9. R. Haenni. Modeling uncertainty with propositional assumption-based systems. In S. Parson and A. Hunter, editors, *Applications of Uncertainty Formalisms*, Lecture Notes in Artifical Intelligence 1455, pages 446–470. Springer-Verlag, 1998.
10. K.D. Heidtmann. Smaller sums of disjoint products by subproduct inversion. *IEEE Transactions on Reliability*, 38(3):305–311, 1989.
11. J. Kohlas and R. Haenni. Assumption-based reasoning and probabilistic argumentation systems. In J. Kohlas and S. Moral, editors, *Defeasible Reasoning and Uncertainty Management Systems: Algorithms*. Oxford University Press, 1996.
12. M. Lalmas. Dempster-shafer's theory of evidence applied to structured documents. In *Proc. of the Int. ACM-SIGIR Conf.*, pages 110–118, 1997.

13. M. Lalmas. Logical models in information retrieval: Introduction and overview. *Information Processing & Management*, 34(1):19–33, 1998.
14. I. Ounis and M. Pasca. RELIEF: Combing expressiveness and rapidity into a single system. In *Proc. of the Int. ACM-SIGIR Conf.*, pages 266–274, 1998.
15. J. Picard. Modeling and combining evidence provided by document relationships using probabilistic argumentation systems. In *Proceedings of the International ACM-SIGIR Conference on Research and Development in Information Retrieval*, pages 182–189, Melbourne, Australia, 1998.
16. J. Picard. Finding content-bearing terms using term similarities. In *Ninth Conference of the European Chapter of the Association for Computational Linguistics*, Bergen, Norway, 1999. Accepted for publication, student session.
17. J.M. Ponte and W.B. Croft. A language modeling approach to information retrieval. In *Proc. of the Int. ACM-SIGIR Conf.*, pages 275–281, 1998.
18. Y. Qiu and H.P. Frei. Concept based query expansion. In *Proc. of the Int. ACM-SIGIR Conf.*, pages 160–169, 1993.
19. S.E. Robertson and K. Sparck-Jones. Relevance weighting of search terms. *Journal of the American Society for Information Science*, 27(3):129–146, 1976.
20. J. Savoy. A learning scheme for information retrieval in hypertext. *Information Processing & Management*, 30(4):513–533, 1994.
21. J. Savoy. Ranking schemes in hybrid Boolean systems: A new approach. *Journal of the American Society for Information Science*, 48(3):235–253, 1997.
22. C.J. van Rijsbergen. A theoretical basis for the use of co-occurrence data in information retrieval. *Journal of Documentation*, 33(2):106–119, 1977.