ELECTRONIC **WORKSHOPS IN COMPUTING**

Series edited by Professor C.J. van Rijsbergen

Jonathan Furner, School of Information and Media Studies, and David Harper, School of Computer and Mathematical Studies, The Robert Gordon University, Aberdeen, Scotland. (Eds)

Information Retrieval Research

Proceedings of the 19th Annual BCS-IRSG Colloquium on IR Research, Aberdeen, Scotland, 8-9 April 1997

Paper:

A Logical Relational Approach for Information Retrieval Indexing

I. Ounis and T.W.C. Huibers

Published in collaboration with the British Computer Society

BCS

Springer

A Logical Relational Approach for Information Retrieval Indexing

Iadh Ounis

CLIPS-IMAG ,MRIM Team, University of Grenoble

BP 53, 38041 Grenoble Cedex, France

Theo Huibers

Department of Information Systems, University of Nijmegen

Toernooiveld 1, NL-6525 ED Nijmegen, The Netherlands

Abstract

In a relational indexing approach (see e.g. Farradane's work), information is carried by a fixed set of relationship types over an underlying set of terms. The idea is that the essence of the meaning of information is encapsulated in the relationships between terms. The importance of relationships is now widely recognized within many fields such as relational databases and knowledge representation formalisms. These fields have substantially improved our understanding of relationships and the problems involved in trying to formalize them. However, although those relationships can be correctly represented by almost all the well-known formalisms in such fields, they are not exploited as much as the objects by concrete operations. In information retrieval, previous attempts at managing relationships have mainly addressed structural aspects, and exclude the manipulation of index expressions by relational operations. This paper suggests a prime use of the relation properties through a logical framework, in a way that it can improve the effectiveness of the matching operation.

1 Introduction

Computer systems are becoming increasingly complex. This is certainly valid for information retrieval systems. With the explosive growth of the amount of information available via the Internet, the high perceived value of multimedia information (texts, graphics, images, video, etc.) and the emergence of new applications such as digital libraries and hypermedia, there has been a strong need for new techniques and models to access this information and to improve the effectiveness of the retrieval process. More and more approaches (theoretical and practical) are being investigated in order to expand the boundary of information retrieval. For instance the field of logic-based information retrieval [1, 2, 3, 4] is broadening rapidly as a theoretical framework for studying information retrieval. The use of logic can provide all the necessary tools to model the different functions of an information retrieval system, and in addition, seem to be a more accurate model of information. Furthermore, a logic provides the possibility to explain the information retrieval performance or behavior [4].

Despite all novel approaches the fundaments of an information retrieval system remain the same. An information

retrieval system is still viewed as a system that selects documents on the basis of a matching operation between a document representation $\chi(d)$ and a query representation q. If the matching operation deems a document as being sufficiently similar to the query, then the document is assumed *likely to be relevant* and returned to the user. According to the *logical model* of information retrieval, the task of the system can be described as the extraction, form the document base, of those documents d that, given a query q, make the formula $\chi(d) \rightarrow q$ valid, where $\chi(d)$ and q are formulae of the underlying logic, and "$\rightarrow$" denotes the logical consequence decision formalized by the logic in question. Such a logic-based approach originates in the work of Cooper [5] who provided a formal definition of relevance in terms of logical consequences. However, this approach to information retrieval and to relevance, as well as the classical approaches [6, 7], is commonly called the topical approach [8, 9] in which only the document's and the query's representations matter. As mentioned by Saracevic [10], it has been known for a long time that topicality is not the only criterion of a user's relevance judgment. A number of others factors also affect the relevance result such as the user's knowledge, the expected use of information, the application domain and so on. Van Rijsbergen [1] pointed out this problem when he proposed the use of a non-classical logic for information retrieval.

A general part of information retrieval systems is the index language, which is the language used to represent the documents in the collection, and the request of the seeker. The creation of the internal representation of a document is a prominent function of any information retrieval system. It is often referred to as the *indexing* process. The outcome of this process is a set of index expressions that supposedly summarize the information content of a document. The index expressions can be keywords, parse trees, semantic structures and so on. Central to our approach in this paper is the assumption that the output of the internal representation can influence considerably the effectiveness of the information retrieval system, especially when this output consists of some *semantic structures* (conceptual graphs, parse trees, infons, etc.). Indeed, as mentioned by [11], the most intricate or carefully designed retrieval algorithm can not compensate for inappropriately represented documents. It is then evident that the index language must be much more expressive and richly structured than the keywords-based languages usually adopted by the classical information retrieval systems. As a consequence, the index expressions should have a complex internal structure that we must handle with the utmost care: the more complex the index expressions are in their structures and contents, the more the underlying semantics of this complexity has to be made explicit. We claim here that the knowledge we have about the resulting indexing expressions can improve the matching process.

Recently, some authors make use of the knowledge representation formalisms to analysis document's information content and information need, and to evaluate the relevance of a document to a query. Good examples of such formalisms, experienced within the RIME [12] and MIRTL [13] projects, are based on a formalism derived from the notion of Conceptual Dependency and on Terminological Logics, respectively. Usually, these formalisms allow for a number of term-forming operators by means of which one may build *semantic structures* starting from a basic repertory of simple terms and relationships (see figure 1) [1]. Hence, a semantic structure is an expression of the form E_i *Relation* E_j, where E_i and E_j are simple or complex terms, and *Relation* is a relationship holding between the terms. As a consequence, one can explore in addition to the thesaurus of terms (synonymy, related terms, etc.), the relationships properties, like symmetry, transitivity and so on, to derive more knowledge. This can be done through a logical framework following the ideas of Nie [14]. He showed that the logical model suggested by van Rijsbergen [1] seems to be a very promising approach to represent the impact of the system's knowledge on the matching operation.

[1] In this example, *bears-on* and *has-for-value* are relations, while OPACITY, LUNG and TISSULAR are terms.

Hence, various kinds of inference rules can then be determined, and they form the derivation system which drives the plausible inference process between the documents and the queries. This derivation system will be defined over the language of index expressions.

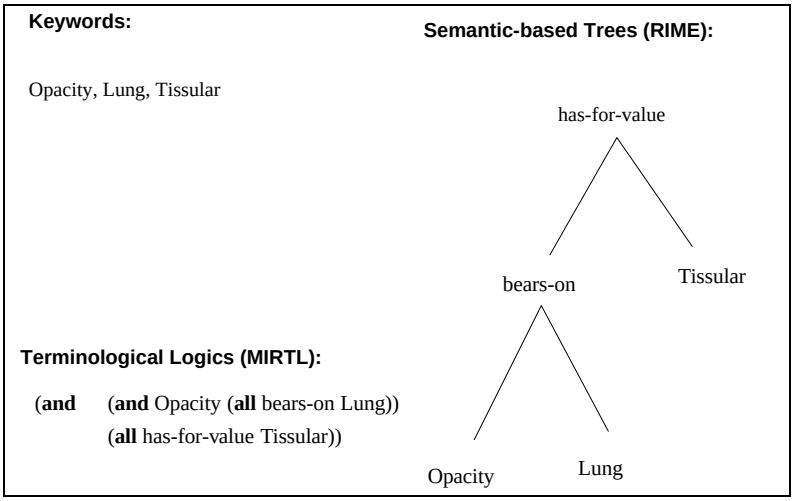


Figure 1: Examples of index expressions for *" A Tissular opacity affecting the Lung"*

The contributions made by this paper are twofold. Firstly, some useful relational operations in information retrieval are given. These operations allow us to handle the relationships that may occur in the documents and the queries representations. Secondly, we can generate more, hopefully precise, derivations and as a consequence more relevant answers to the user. The remainder of the paper is organized as follows. In Section 2 we recall the classical indexing approaches used in information retrieval. In Section 3 we will show how the semantical properties of index expressions can be used to generate more derivations rules. We define rules that take into account the relationships involved in these index expressions. In Section 4 we will show how the relationships will be organized in case of real information retrieval systems, some particular problems concerning the application of such an approach to information retrieval are also mentioned, while further directions of investigations are described in Section 5.

2 Current logical approaches

One can define a relation as a set of tuples that represents a relationship among objects in the universe of discourse. Each tuple is a finite, ordered sequence of objects [15]. Tuples are in the universe of discourse, and can be represented as individual objects. Usually, in the literature the words relation and relationship refer to the same thing. In CycL [15], relations are called relationships and in LOOM [16], relations are called relations. Usually, relations are denoted by predicates. A fact that a particular tuple is an element of a relation is denoted by **relation-name** $(arg_1, arg_2 \ldots arg_n)$, where the arg_i are the objects in the tuple. In the case of binary relations, the fact can be read as "arg_1 is **relation-name** arg_2" or "a **relation-name** of arg_1 is arg_2".

One of the first indexing strategies used in information retrieval was the extraction of *keywords* or *topics* from documents and the valuation of their importance according to their frequency of appearance. It has often been noted

that this approach neglects the important relations between keywords [9]. Sometimes, those relationships are even removed from the indexing process by an initial selection of words that removes the common words in the document (like "is", "on", "of", etc.); This is mainly done out of practical considerations. Such a classical approach can benefit from the availability of efficient algorithms that automatically extract the keywords from the document. As a result the relationships in which the keywords were involved are no longer taken into account. However, these relationships could be used to provide contextual information for driving the matching process. In fact, in the real word, relationships between objects play an important role. In the field of relational databases [17] or information systems engineering (e.g. Merise [18]), the importance of relations has already been recognized.

In the well-known information retrieval CACM test collection, we have noted a high occurrence frequency for some relations. After analysis, we have converted this collection into an homogeneous representation including only the title and the abstract fields. The number $\mathcal{N}$ of words in the collection was then computed. We has developed a tool that allows to search for sentences containing a sequence of words. It is possible to include parameters in these sentences. Hence, by the query "*the #parameter on #parameter*" we mean that we are looking for all sentences in the collection beginning with the word "*the*" followed by any string, followed by the relation "*on*" and ending with any string. Some example of these sentences are shown in the figure 2, with their respective probability of occurrence in the collection, according to the number $\mathcal{N}$.

Form of the relation	*Numb. of Occ*	*Probability*
$Entity_A$ **on** $Entity_B$	2680	0.0128
the $Entity_A$ **on** the $Entity_B$	190	0.00091
$Entity_A$ **in** $Entity_B$	2598	0.0124
the $Entity_A$ **in** the $Entity_B$	403	0.0019
$Entity_A$ **of** $Entity_B$	2097	0.0100
$Entity_A$ **for** $Entity_B$	1771	0.0084
$Entity_A$ **Without** $Entity_B$	82	0.00039
The paper **deals with/describes** $topic$	29	0.00013

Figure 2: Frequency of some usual relations

In information retrieval the representation of the relationships between keywords takes its root in the work of Farradane described in [19, 20]. Farradane introduced the idea that much of the meaning in information objects is denoted in the relationships between terms. For example, "*John drinks a cup*" exhibits a functional dependence relationship type between *John* and *cup*. Hence, whereas in classical information retrieval approaches such a sentence will be indexed by the two keywords *John* and *cup*, Farradane projected the idea that the relationship *drink* must also be represented in the final index of the sentence. In fact, in this special example, the indexing of the sentence by the keywords *John* and *cup* can even be called erroneous, as obviously John drinks the content of the cup (which can be wine, coke, water, etc.) and not the cup itself. The representation of the relationship *drink* will avoid such an ambiguity as we all know that we don't drink *cups* but only the liquid they contain.

A parallel was drawn with the conceptual model from the database world where relationship types between entities play an important role; the index description of a document consisting solely of keywords would be like an

entity relationship model without relationship types. In Farradane's work, information was carried by a fixed set of relationship types over an underlying set of terms. This conception bears a close resemblance to a large class of knowledge formalisms such as terminological logics [21], conceptual graphs [22] and situation theory [3]. A terminological logic is a subset of first order logic with equality that contains only unary relations, representing sets of objects in the domain (referred to as concepts) and binary relations (called roles) linking together the objects of the domain. A conceptual graph is a bipartite, connected, finite and oriented graph of *concepts* and *conceptual relations*. In the graphs, concept nodes represent entities, attributes, states and events, and relation nodes show how the concepts are interconnected [22]. Finally, the situation theory formalism introduces the important notion of infon [23, 3]. An infon is a structure that represents the information that a relation R holds or does not hold between a particular set of objects. The use of those three formalisms in information retrieval was partially motivated by the fact that they all allow to represent more complex index terms and they offer enough capabilities to represent objects and relationships between objects [13, 24, 25, 26, 27]. Recent studies [28, 9] about the impact of structured documents on both indexing and retrieving have shown the need to represent some aggregative relationships that satisfy a set of properties and constraints.

However, if the relationships between terms can be represented by almost all the actual knowledge representation formalisms, they are not exploited as much as the objects by the primitive operations given by those knowledge formalisms. Moreover, no behavior can explicitly be associated to relationship types, and neither does relation-based reasoning allow one to explore the implicit knowledge that can be of beneficial use in the matching process. For example, it has been shown in [29], that although the terminological logics are able to represent the relationships needed to index complex and structured documents, it is far from clear how to describe their characteristics. For instance, it is not possible to specify mathematical properties about roles (symmetry, transitivity, ...) without considering all possible derivable facts as a part of the knowledge base. Let us take a simple example: assume that *Aggregate*(os_i, os_j) and *Aggregate*(os_j, os_k) are two assertions (facts) in the terminological knowledge base Σ. Also, assume that the *Aggregate* role is transitive. In such case, there is no possibility in terminological logics to infer *Aggregate*(os_i, os_k) from the knowledge base Σ. Hence, a document indexed by *Aggregate*(os_i, os_j) and *Aggregate*(os_j, os_k) will not be retrieved by a query containing *Aggregate*(os_i, os_k). In order to avoid this problem, we must add explicitly this derivable fact to the knowledge-base Σ. However doing this would be very expensive, especially in cases where the document base contains several thousands of structural objects. In order to avoid practical problems like this, the introduction of some relational operations can help to describe the relationships behavior. Hence, we aim in the following sections to operate directly on the relationship types that can occur in semantical index expressions. We will propose a technique that can infer new information by analyzing the properties of the relations such as symmetry, transitivity, semantical behavior, arrangement with others relationships, and so on. This will be done through a general logical framework for studying relationships and their semantical properties. This framework captures the semantical information of the relations for information retrieval purposes by specifying their properties by way of inference rules.

3 A logical relational approach

Suppose one wants to build a logical system for a given application. First of all, one must specify which language $\mathcal{L}$ is to be used for defining the notion of *well formed formulae* in this system. This will allow us to construct the

axioms and rules capturing the system. In other words, it is not enough to know the behavior of this logical system, as one must also know how it is presented. Here, we are interested in a logical framework for information retrieval. Hence, if one considers the deduction as the retrieval operation, this means that the relevance relationship between document and query can be established in terms of the axioms and rules belonging to the logical system. As a result, the index language we use for the description of documents and queries constitutes the one specifying the syntax of a well formed formula in the above system. Following Farradane's idea, the index expressions must be able to represent the following form : $E_i\ Relation\ E_j$ where E_i and E_j are simple or complex descriptors, and *Relation* is a relation holding between the descriptors. We choose the situation theory [30, 23] as the derivation system's language:

Definition 3.1 (The index expression) Let $\mathcal{O}$ be a set of objects and $\mathcal{R}$ a set of relations. We define recursively the language $\mathcal{L}(\mathcal{O},\mathcal{R})$ of index expressions over $\mathcal{O}$ and $\mathcal{R}$ as the set of all structures $\langle\langle R,\mathrm{o}_1,\ldots,\mathrm{o}_n;\ i\rangle\rangle$ that represents the information that the relation R holds (if i = 1) or does not hold (if i = 0) between the objects $o_1, o_2, \ldots, o_n$.

The objects in this definition include *individuals*, such us 'John', 'Cup, etc.; *types*, high order uniformities, for instance concepts like PERSON, TABLE and finally complex structures corresponding to an index expression. A relation R is a uniform property that links the objects. For instance, it can exhibits a position relationship (in, on, etc.), an association relationship (with, and, or, etc ..), a directed association relationship (to, for, etc...), an action relationship (write, print, create, ...) and so on.

Example 1 *The Expressions* $\langle\langle$and, IR, DB*;1*$\rangle\rangle$, $\langle\langle$on, CAT,TABLE*;1*$\rangle\rangle$ *and* $\langle\langle$attitude-of, DOG, CATS*;1*$\rangle\rangle$ *can be considered as index expressions.*

The semantical content of each document depends greatly on the relationships between objects and the index expressions it contains, and these relationships are the means of describing how the objects are combined. It seems interesting then to capture the behavior and the properties of these relationships through a set of inference rules in order to generate more knowledge about the content of each document. Such relationship properties will be defined by the set of axioms and rules contained in the derivation system.

Definition 3.2 (The derivation system) Given a language $\mathcal{L}(\mathcal{O},\mathcal{R})$ as defined in 3.1, a derivation system $\mathcal{S}$ is a pair of the form $(Ax, Rule)$, with Ax a subset of expressions constructed from $\mathcal{L}(\mathcal{O},\mathcal{R})$ that we call axioms, and $Rule$ a set of rules of the form $R(T_1, \ldots, T_k, T_{k+1})$. Here, $T_1, \ldots, T_k$ are the premises of the rule and T_{k+1} is the conclusion, all of which are elements from the language $\mathcal{L}(\mathcal{O},\mathcal{R})$.

A document d is assumed to be *logically relevant* to a query q if there exists a causal chain of derivations beginning at the document and ending at the query. The derivation system attempts to derive if a document is relevant to a given information need. Following Maron [31], we call the logical relevance relation '*aboutness*' denoted by $\boxdot\!\!\rightsquigarrow$.

The document characterization $\chi(d)$ is an approximate representation of the document's content. Whereas in classical information retrieval systems, a document is characterized by a set of keywords, we will assume here that the representation of the document consists of a set of index expressions built from the language $\mathcal{L}(\mathcal{O},\mathcal{R})$ as described in 3.1.

Definition 3.3 (The document characterization) A document d is characterized by a set of index expressions $\chi(d)$ in the language $\mathcal{L}(\mathcal{O}, \mathcal{R})$. The elements of $\chi(d)$ constitute a *monoid* with an associative binary operation $\odot$ satisfying the following properties:

- Implication property:

$$\begin{array}{lll} \forall e_i, e_j \in \chi(d) & e_i \odot e_j & \boxdot\!\!\rightsquigarrow \quad e_i \\ & e_i \odot e_j & \boxdot\!\!\rightsquigarrow \quad e_j \end{array}$$

- $\odot$ is *commutative*. This property expresses that the semantic content of the document characterization is independent of the order in which we consider the aggregation of its index expressions:

$$\forall e_i, e_j \in \chi(d), e_i \odot e_j \boxdot\!\!\rightsquigarrow e_j \odot e_i$$

- By definition we assume that the aboutness inference $\boxdot\!\!\rightsquigarrow$ is reflexive, this seems to be an inherent property of aboutness in many IR models [3]; i.e., $\forall e \in \mathcal{L}(\mathcal{O}, \mathcal{R}), e \boxdot\!\!\rightsquigarrow e$. Using this property, one may infer that the operator $\odot$ is *idempotent*:

$$\forall e \in \mathcal{L}, e \odot e \boxdot\!\!\rightsquigarrow e$$

- The empty index expression ε is a neutral element for $\odot$:

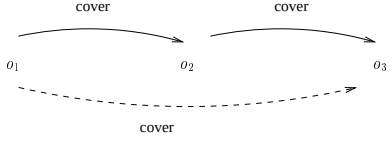
$$\forall e \in \mathcal{L}, e \odot \varepsilon \boxdot\!\!\rightsquigarrow e$$

A query can been seen as a request for information. It can therefore also been represented as an aggregation of expressions from the language $\mathcal{L}(\mathcal{O}, \mathcal{R})$. What remains now is the question of how to derive the query from the document description in a way that will improve the effectiveness of the matching operation.

Given the definitions above, we offer to operate directly on relations, that means we propose to exploit their semantical properties and the information they bear implicitly. Hence, in addition to the semantic networks that establish the relationships that objects have to other objects (i.e., the thesaurus), usually used in classical information retrieval approaches, for each relation we give its properties, its behavior and its arrangement with others relations. All these information will be captured through a set of derivation rules.

3.1 The Mathematical Properties

General mathematical properties can be used to augment relations implicitly contained in the knowledge's system. Such a relations can be used to improve the matching process between the document and the query. For instance suppose that in an image, we have three objects o_1, o_2 and o_3 and that they are connected by the "*cover*" spatial relationships, then we can add a new "*cover*" relationship between o_1 and o_3 such as::



Indeed, in the raster mode, we say that an object o_1 *covers* another object o_2 if, and only if, all the pixels of the object o_1 are included in the pixels of the object o_2, i.e. pixel(o_1) $\subset$ pixel(o_2). This relation is transitive as mentioned by [32].

It should be noted that these mathematical properties are more easy to formalize when they are associated to binary relations. However, although it is theoretically possible to split a relation into binary relations [22], we choose here to handle both binary and N-ary relations. Indeed, it is perhaps the case that due to some practical considerations, one may need some N-ary relations. For instance in [29], it has been shown that in order to cope with the complexity of structured multimedia documents, some tertiary relations are needed. To represent such relations, we need two primitive binary relations which are more difficult to manage: we must check if all the properties of the N-ary relation are faithfully represented. In fact one can easily see that the only use of binary relations requires the creation of a large number of relations which makes dealing with them hard and rather complex their treatments. Using the N-ary relations, the number of relations will decrease and a better computational behavior of the underlying operational system will be expected. On the other hard, one can remark that only the N-ary relations having an interest in information retrieval will be considered in the framework. As their semantical properties are known, it is not hard to include them such as derivation rules.

The mathematical properties of relations can be defined as follows:

Sequentiality

$$\frac{\langle\langle R, e_1, e_2, \ldots, e_k; 1\rangle\rangle \odot \langle\langle R, e_k, e_{k+1} \ldots, e_{k+m}; 1\rangle\rangle}{\langle\langle R, \ldots e^{N-1} \ldots, e_{k+m}; 1\rangle\rangle}$$

In case *R* is an N-ary relation, e^{N-1} denotes a combination of $N-1$ expressions chosen from the set e_1, …, e_{k+m-1}. We can have $N-1$ possible derivations, according to which combination of expressions we take.

For example in case of tertiary relations, the following derivations are valid if the relation R satisfy the above *Sequentiality* property:

Example

$$\frac{\langle\langle R, b, a, c; 1\rangle\rangle \odot \langle\langle R, c, b, d; 1\rangle\rangle}{\langle\langle R, b, a, d; 1\rangle\rangle}$$

or

$$\frac{\langle\langle R, b, a, c; 1\rangle\rangle \odot \langle\langle R, c, b, d; 1\rangle\rangle}{\langle\langle R, c, a, d; 1\rangle\rangle}$$

For instance, the relation "*Between*" that is a tertiary relation satisfy the two above derivation rules. This property is a general case of the transitivity relation.

The *permutation* rule expresses the claim that the order of linking some objects in the relation does not have any influence on the derivation process.

Permutation

$$\frac{\langle\langle R, e_1, e_2, \ldots, e_i, \ldots, e_j, \ldots, e_k; 1\rangle\rangle}{\langle\langle R, e_1, e_2, \ldots, e_j, \ldots, e_i, \ldots, e_k; 1\rangle\rangle}$$

Consider for instance the relation *Between*. If we assume that $\langle\langle$*Between*, $X, Y, Z; 1\rangle\rangle$ means "*X is between Y and Z*", then the document "*Reading is between London and Bristol*" is about the query "*Reading is between Bristol and London*", as the *Between* relation is symmetric. We have the following instantiation of the "*Permutation*" rule:

Example

$$\frac{\langle\langle \textit{Between}, \text{Reading}, \text{London}, \text{Bristol}; 1\rangle\rangle}{\langle\langle \textit{Between}, \text{Reading}, \text{Bristol}, \text{London}; 1\rangle\rangle}$$

In case of binary relations, this property corresponds to the classical notion of *Symmetry*. For instance, the relation Married-To is Symmetric. The symmetry property is primarily intended to increase the number of the aboutness theorems [3].

Relations could have the behavior of a function. In the work of Chiaramella and al. [28], it has been shown that in order to allow a solid formal basis to express the retrieval of structured documents, some specific functions must be introduced. They can satisfy one of the following logical constraints:

- there is no $i_1, \ldots, i_n, e_1, e_2$ such that $R(i_1, \ldots, i_n, e_1) \wedge R(i_1, \ldots, i_n, e_2)$
- there is no $e, i_1, \ldots, i_n, j_1, \ldots, j_m$ such that $R(e, i_1, \ldots, i_n) \wedge R(e, j_1, \ldots, j_m)$

In our logical framework, the first constraint can be expressed as follows:

Left Exclusivity

$$\frac{\langle\langle R, i_1, \ldots, i_n, e_1; 1\rangle\rangle}{\langle\langle R, i_1, \ldots, i_n, e_2; 0\rangle\rangle}$$

For example we can see that, in a lattice of types, the greatest lower bound (GLB) operation on types that search for their greatest common sub-type satisfies the *Left Exclusivity* constraint.

Instead of applying the *Exclusivity* property at the left side of the tuples, we can have the following right variant formalizing the possible second constraint:

Right Exclusivity

$$\frac{\langle\langle R, e, i_1, \ldots, i_n; 1\rangle\rangle}{\langle\langle R, e, j_1, \ldots, j_m; 0\rangle\rangle}$$

For instance, assume that a structured document is composed of a set of structural objects. For instance, it may consist of a mixture of text, image, graphic and video. Assume that $Desc_{str} : \mathcal{D} \times 2^{SO}$ is a function that for a given structural document d gives the set of all its component objects, where SO is the set of all structural objects in the document base. We have the following postulate:

Example

$$\frac{\langle\langle Desc_{str}, d, i_1, \ldots, i_n; 1\rangle\rangle}{\langle\langle Desc_{str}, d, j_1, \ldots, j_m; 0\rangle\rangle}$$

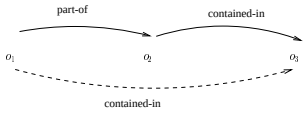


Figure 3: Arrangement of two relations

3.2 Arrangement with others relations

Assume that we have two relationships between three objects o_1, o_2 and o_3 as represented by the full arrows in the figure 3. One can then add the relationship between o_1 and o_3 according to the semantical properties of the relations *Part-of* and *Contained-in.*

Some special relation relationships are then to be considered. The next property rule specifies that the tuples of two relations R_1 and R_2 may be juxtaposed to form a new tuple of a relation R_3.

Juxtaposition

$$\frac{\langle\langle R1, e_i, \ldots, e_k; 1\rangle\rangle \odot \langle\langle R2, e_k, \ldots e_n; 1\rangle\rangle}{\langle\langle R3, e_i, \ldots, e_k, \ldots e_n; 1\rangle\rangle}$$

In the case of binary relations, this rule can be instantiated by the well-known composition relation:

Composition

$$\frac{\langle\langle R1, x, y; 1\rangle\rangle \odot \langle\langle R2, y, z; 1\rangle\rangle}{\langle\langle R3, x, z; 1\rangle\rangle}$$

For instance assume that a document deals with *the effects of fundamentalism in society*. Given that this document is indexed by $\langle\langle$*Incites-To*, FUNDAMENTALISM, HATRED; $1\rangle\rangle$ and $\langle\langle$*Causes*, HATRED, VIOLENCE; $1\rangle\rangle$, then this document will be retrieved by a query looking for all documents about *fundamentalism as a factor of violence.* This is due to the knowledge we have about how the relations *Incites-To* and *Causes* can be arranged:

Example

$$\frac{\langle\langle \textit{Incites-To}, \text{FUNDAMENTALISM}, \text{HATRED}; 1\rangle\rangle \odot \langle\langle \textit{Causes}, \text{HATRED}, \text{VIOLENCE}; 1\rangle\rangle}{\langle\langle \textit{Is-Factor-Of}, \text{FUNDAMENTALISM}, \text{VIOLENCE}; 1\rangle\rangle}$$

Some relations could in certain cases[2] satisfy the *Strict Composition* rule instead of the *Composition* rule. This rule states the following:

Strict Composition

$$\frac{\langle\langle R1, x, y; 1\rangle\rangle \odot \langle\langle R2, y, z; 1\rangle\rangle}{\langle\langle R2, x, z; 1\rangle\rangle}$$

We can have then the following example:

Example

$$\frac{\langle\langle \textit{Part-Of}, o_1, o_2; 1\rangle\rangle \odot \langle\langle \textit{Contained-In}, o_2, o_3; 1\rangle\rangle}{\langle\langle \textit{Contained-In}, o_1, o_3; 1\rangle\rangle}$$

[2] See the example of the relations *Part-Of* and *Contained-In* in the beginning of this section.

3.3 Link with others relations

The definition of links between relations enables one to deduce new information about the indexing expressions, using relation-based reasoning. Hence, relations can be handled as well as concepts.

Like the synonymy in case of keywords or concepts (considered as unary relations), one may express that two relations are the same. The following "*Alias*" rule is a way to specify that two relations have the same extension and then that they are logically equivalent. Like for the terms (keywords, concepts, etc.), in information retrieval, this rule may be useful to avoid the omission of some relevant answers when the name of the relation mentioned by the user is not the same as the one used in the indexing process.

Alias

$$\frac{\langle\langle R1, e_1, \ldots e_n ; 1\rangle\rangle}{\langle\langle R2, e_1, \ldots, e_n ; 1\rangle\rangle}$$

A link between relationships that can be interesting for information retrieval is introduced by the following *Inversion* rule.

Inversion

$$\frac{\langle\langle R1, e_1, e_2, \ldots, e_n ; 1\rangle\rangle}{\langle\langle R2, e_n, \ldots, e_2, e_1 ; 1\rangle\rangle}$$

This rule signifies the fact that the relations R_1 and R_2 are equivalent when their arguments are swapped is mentioned. Relationships of this kind are often used in some knowledge representation formalisms, especially in the case of binary relations. For instance in many terminological logics, a special operator **inv** is introduced. This operator applied to a given role (binary relation) produces an inversion of its arguments. This derivation rule allows for instance to derive that an image in which *A tree is on the left of a house* is relevant to a query looking for all images representing *A house on the right of a tree*:

Example

$$\frac{\langle\langle \textit{On-Left}, \textsc{Tree}, \textsc{House} ; 1\rangle\rangle}{\langle\langle \textit{On-Right}, \textsc{House}, \textsc{Tree} ; 1\rangle\rangle}$$

A relation R_1 can be a sub-relation of another relation R_2. Intuitively, if these relations are viewed as sets of tuples, R_1 is a subset of R_2. In other words, every tuple of R_1 is also a tuple of R_2, i.e., if R_1 holds for some objects $o_1, o_2, \ldots, o_n$, then the relation R_2 holds for the same arguments. The following *Subrelation* rule expresses this fact. Note that a relation and its sub-relation must have the same arity N.

Subrelation

$$\frac{\langle\langle R1, e_1, e_2, \ldots, e_n ; 1\rangle\rangle}{\langle\langle R2, e_1, e_2, \ldots e_n ; 1\rangle\rangle}$$

For example, if a document is about a situation where "*John is the child of Mary*", one can conclude that this document is also about a query looking for all situations where: "*John and Mary are relative*".

Example

$$\frac{\langle\langle \textit{Is-Child-Of}, \text{John}, \text{Mary} ; 1\rangle\rangle}{\langle\langle \textit{Is-Relative-Of}, \text{John}, \text{Mary} ; 1\rangle\rangle}$$

The next rule denotes the *Simultaneity* property. Let for instance n and m be the respective arities of R_1 and R_2. Let $i_1, i_2, \ldots, i_{n-1}$ and $j_1, j_2, \ldots, j_{m-1}$ be a set of indexing expressions. The *Simultaneity* property states the fact that each indexing expression e taking part in the relation R_1 by $R_1(e, i_1, i_2, \ldots, i_{n-1})$, necessarily takes part in the relation R_2 such that $R_2(e, j_1, j_2, \ldots, j_{m-1})$ and vice versa. In our framework, this property can be expressed as follows:

Simultaneity

$$\frac{\langle\langle R1, e, i_1, \ldots, i_{n-1}; 1\rangle\rangle}{\langle\langle R2, e, j_1, \ldots, j_{m-1}; 1\rangle\rangle}$$

For instance, if we assume that $\langle\langle$*Parents*, $x, y, z; 1\rangle\rangle$ means "*the parents of x are y and z, with y being the mother and with z being the father*", then a document about $\langle\langle$*Parents*, $x, y, z; 1\rangle\rangle$ is also about $\langle\langle$*Have-As-Father*, x, z $; 1\rangle\rangle$.

Another rule claims that for some pairs of relations R_1 and R_2 having the same arity N, there is no set of objects $o_1, o_2, \ldots, o_n$ that can be linked together at the same time by R_1 and R_2. In terms of our framework, we write:

Preclusion

$$\frac{\langle\langle R1, e_1, e_2, \ldots, e_n; 1\rangle\rangle}{\langle\langle R2, e_1, e_2, \ldots, e_n; 0\rangle\rangle}$$

In case of binary relations, if a document is about an expression such as $\langle\langle With,$ TRAIN, RESERVATION; $1\rangle\rangle$, then this document is not about a query looking for *Trains without Reservation*, i.e.$\langle\langle Without,$ TRAIN, RESERVATION; $1\rangle\rangle$. Here, the relations "*With*" and "*Without*" preclude each other. This property is interesting for information retrieval as it may be used to determine the non-aboutness [3].

4 A theoretical study of relations

In the previous section, we have presented a general logical mechanism of relation-based reasoning. It is based on a selection of rules that possess some important properties, according to real cases we have encountered in today's information retrieval systems [32, 29]. Detailed investigations of what an interesting relation for information retrieval consists of are needed in order to develop an operational system. Indeed, it remains clear that only a specific set of relations are useful or need to be characterized in case of information retrieval. For instance, as mentioned by Palmer [33], the verbs do not convey useful properties. On the other side, we already know that one needs a specific class of aggregative relations in order to handle the structured multimedia documents [28]. For example, one can represent the information that a book b_1 is composed of two chapters c_1 and c_2 by the relations *Aggregate*(b_1, c_1) and *Aggregate*(b_1, c_2), with *Aggregate* an aggregative relation. Another class of useful relations in information retrieval is exposed in the work of Mechkour described in [32]. This work proposes a model for images retrieval and describes a class of spatial relationships that may occur in an image. Twelve spatial relations were proposed in this model, like *far, near, touch, in, etc.*.

Our logical mechanism can be integrated to all the knowledge formalisms that don't allow for relation-based reasoning. Hence, this will resolve their limitations in case where the query refer to some common properties about the relations. The result is an hybrid model that combines the original characteristics of the chosen formalism and our relation-based framework, in a way that it can handle relations, as well as concepts. For instance, assume that a

document d is represented by the full nodes in the figure 4. Hence, this document will be retrieved by the query q, only if the properties of the relations *Aggregate* and *Cover* are applied. As introduced by Sowa [22], the conceptual graph formalism don't allow for such kind of reasoning.

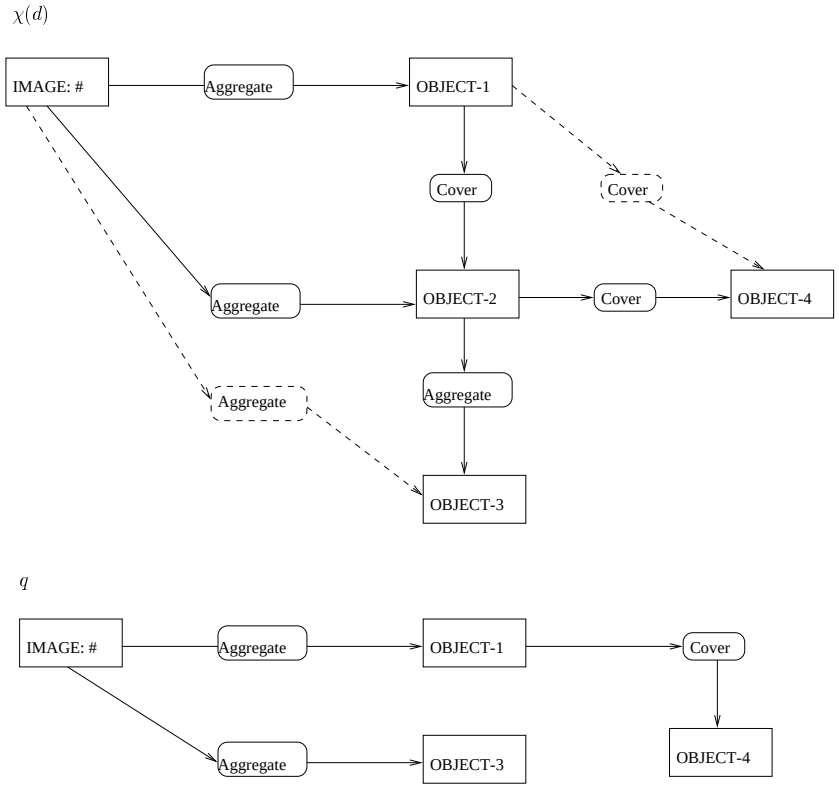


Figure 4: A relation-based reasoning example

Some specific classes of relations can be defined. These classes will serve as a basis to express the properties of relations, their relation relationships and their links with others relations, in order to make explicit their behavior. Indeed, in addition to the fact that it is more easy to determine those properties at the level of the same class, it is usually the case that only relations of a common class can be agenced or linked together. A study of the different kind of relationships used in the information retrieval literature shows that mainly prepositions and verbs are used as relations in the indexing expressions. For example, in Palmer's work [33], an index expression consists of a noun phrase, called a *conceptual group*, where the relationships are essentially French prepositions. Bruza's approach adopts the same principle with English prepositions [34].

As a consequence, a classification of this particular kind of relations can have an interest for our approach. In the table 1, we give a possible organization of the prepositions. This table was constructed out of several English student books about prepositions. Only the prepositions that we judge interesting in information retrieval are considered. Our judgment was mostly motivated by their high occurrence frequency in some collections like CACM or Cranfield. According to their current meaning in English and the way they are used in the sentences, we give four general classes of prepositions: *Time, Place, Direction, Conjunctions*. Each general classes may have some subclasses (See

the *Subrelation rule*). For instance the preposition *In-the-middle-of* belongs to the *AREA* class of relations, which is a subclass of the *PLACE* class.

The relation-based mechanism will rely on such a classification to determine for each relation its properties and behavior. This will be done within each specified class on the relations it contains. Depending on the information contained in the corpus and the document's nature (textual, multimedia, structured, etc.), other classes may be added and analyzed, such as the class of aggregative relations in case of structured documents.

It should be noted that it is difficult to use prepositions correctly. Most of them have several different functions; for instance the dictionary lists eighteen main uses of the preposition *at* [35], though probably only few of them are of interest in information retrieval. In order to differentiate those kind of relations and to specify exactly what kind of the relation *at* we use[3], one may specify a *signature* for each relation. The signature of an n-ary relation R is a sequence of n objects type $\langle t_1, t_2, \ldots, t_n \rangle$ that specifies the types of the objects that can be linked by this relation. Here a hierarchy (lattice) of objects is needed. To be valid each argument (object) of the tuple must be a specialization (a sub-class) of the corresponding type in the signature, according to the hierarchy of objects. For example the relation *Loc*[4] linking an Object to a place may have $\langle \top, \text{PLACE} \rangle$ as a signature, where $\top$ denotes the universal object denoting all the individuals of the domain. We write $t_1 \leq t_2$ to mention the fact that the type t_1 is a specialization of the type t_2. For example, the sentence " *Vehicles arrive at the Station*" can be represented by $\langle\langle$*Loc*,VEHICLE, STATION; $1\rangle\rangle$, where VEHICLE $\leq \top$ and STATION $\leq$ PLACE.

Using the signature, one is able to express the similarity between the sentences of the type "the adventures of Alice" and "Alice's adventure". This can not be deduced by Palmer's [33] and Bruza's [34] approaches. Hence, for each relation, one may specify the following parts:

- The label of the corresponding relation, such as *Loc* for *Locality* relation.
- The class of the relation, such as *EXACT* for relation *loc*, considered as a special case of the *at* preposition in the figure 4.
- The signature of the relation, like $\langle \top, \text{PLACE} \rangle$ for the relation *Loc*.
- An informal comment about the use of the relation, such as "A locality is a relation linking an Object to a Place".
- The Mathematical properties of the relation, for example the *Loc* relation may satisfy the *Left Exclusivity rule*.
- Its arrangement with others relations
- Their links with others relations, for instance: the *Loc* relation is a Subrelation of the relation *Place* (see figure 4), here we use the *Subrelation rule*.

One can remark that the framework we have presented does not prevent an applied rule to have its consequence contradicted by the subsequent application of another rule. One can then distinguish two contexts of reasoning. On one hand, the *non monotonic reasoning* process keeps the last added information in case of conflict and removes all the previous information that contradicts with the new result. On the other side, in case of *monotonic reasoning*, only information in concordance with the available information can be added. In case of conflict, the new

[3] This will lead to consider eighteen different names for the relation *at* if they are all judged to be interesting in information retrieval.

[4] A special kind of the relation *at*.

Word	Time	Place	Direction	Conjunctions
About	Point-Period			
Above		Place		
Across			Direction	
After	Point-Period			
A long way from		Place		
Along			Direction	
Among		Area		
And				Similar
As	Time			Relation
At	Point	Exact		
Back to			Direction	
Before	Point-Period			
Behind		Place		
Below		Place		
Beside		Place		
Between		Area		
But				Opposite
By	Point-Point	Area		
Down			Direction	
For	Period			
From	Point-Period			
In	Period	Area		
In front of		Place		
In the middle of		Area		
Into			Direction	
Near		Area		
Next to		Place		
Not far from		Place		
Of		Place	Direction	
Off			Direction	
Opposite		Area		
On	Point	Area		
Onto			Direction	
Or				Alternatives
Out of			Direction	
Over			Direction	
Past			Direction	
Round			Direction	
So				Relation
Since	Point-Period			
Through			Direction	
To			Direction	
When	Question			
Where		Question		
While	Question			
With				Relation
Under		Place	Direction	
Until	Point-Period			
Up			Direction	

Table 1: Prepositions

information is not added but rejected. Depending on the context reasoning approach and the established arrangement between relations, this problem can be resolved. For instance, as the relation $\langle\langle$*Father-Of*$, a, b; 1\rangle\rangle$ precludes the relation $\langle\langle$*Child-Of*$, a, b; 1\rangle\rangle$, it will be impossible in case of monotonic reasoning to add an information like $\langle\langle$*Child-Of*$,$ John$,$ Jack$; 1\rangle\rangle$ if $\langle\langle$*Father-Of*$,$ John$,$ Jack$; 1\rangle\rangle$ already exists in the system's knowledge about the document. Some other questions can be discussed such as the influence of the application order of the rules on the matching process, or whether all the derivable information from the indexing expression should be added to the system's knowledge about the document. For instance, an automatic application of the *inversion* or *Alias* rules may produce redundant information.

5 Conclusion

In information retrieval some indexing processes combine information items by bringing them into a relationship, in order to describe the information content of a document more precisely. In this paper we have proposed a logical framework for studying such relationships and their impact on the matching process. The framework captures some relationship features and properties. Within this framework, rules have been outlined that enables the generation of new implicit information contained in the document. The work presented here does not refer to a particular indexing language: our relational indexing approach does not mention in which way the index terms are achieved.

The indexing terms could be any element of any complex language, such as conceptual graphs, terminological logics and so on. The construction of the framework is work in progress. It constitutes a first step in the understanding of relations and their influence on the matching process in information retrieval. Our interest is focused on the theoretical aspects, refining the accuracy of the framework and evaluating the effectiveness of the relation approach on both recall an precision. Although the expressive power of our framework has been demonstrated, we still have paid no attention to uncertainty aspects. This is an important focus for future work. For example, the valuation of the fact that the index expression $\langle\langle$*Talking-To*$,$ John$,$ Mary$, ; 1\rangle\rangle$ is about $\langle\langle$*Listening-To*$,$ Mary$,$ John$, ; 1\rangle\rangle$ is very often true but not always. It depends on the context, for instance whether they are in the classroom or in a meeting. From an implementation point of view, we are investigating this framework within a conceptual graph environment [32].

References

[1] C. J. van Rijsbergen. A new theoretical framework for information retrieval. In *ACM Conference on Research and development in Information Retrieval, Pisa*, pages 194–200, 1986.

[2] Y. Chiaramella and J.P. Chevallet. About retrieval models and logic. *The Computer Journal*, 35(3), 1992.

[3] T.W.C. Huibers. *An Axiomatic Theory for Information Retrieval*. PhD thesis, Department of Computer Science, Utrecht university, The Netherlands, November 1996.

[4] M. Lalmas. *Theories of Information and Uncertainty for the modelling of Information Retrieval: an application of Situation Theory and Dempster-Shafer's Theory of Evidence.* PhD thesis, Department of Computing Science, University of Glasgow, Scotland, April 1996.

[5] W.S. Cooper. A definition of relevance for information retrieval. *Information Storage and Retrieval*, 7:19–37, 1971.

[6] C.J. van Rijsbergen. *Information Retrieval*. Butterworths, London, 1979.

[7] G. Salton and M.J. McGill. *Introduction to modern Information Retrieval*. Mcgraw Hill Book Company, New York, 1980.

[8] T.J. Froehlich. Special issue on 'relevance'. *Journal of the American Society for Information Science (JASIS)*, 45, 1994.

[9] F. Paradis. *Un modèle d'indexation pour les documents textuels structurés*. PhD thesis, Université Joseph Fourier, Novembre 1996.

[10] T. Saracevic. The concept of 'relevance' in information science: a historical review. In *Introduction to information Science*, pages 111–151. T. Saracevic (ed.), 1970.

[11] D.C. Blair. *Language and representation in information retrieval*. Elsevier science publishers, second edition, 1990.

[12] C. Berrut. Indexing medical reports: The rime approach. *Information Processing and Management*, 26(3):93–109, 1990.

[13] C. Meghini, F. Sebastiani, U. Straccia, and C. Thanos. A model of information retrieval based on a terminological logic. In R. Korfhage, E. Rassmussen, and P. Willit, editors, *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Pittsburgh, PA*, pages 298–307. ACM, ACM Press, June 1993.

[14] J. Nie. *Un Modèle Logique général pour les systèmes de recherche d'informations. Application au prototype RIME*. Phd thesis, Université Joseph Fourier, Grenoble, 1990.

[15] T.R. Gruber. Ontolingua: A Mechanism to Support Portable Ontologies. Reference manual, Knowledge Systems Laboratory, June 1992.

[16] R.M. MacGregor. The evolving technology of classification-based knowledge representation systems. In *Principles of Semantic Networks: Explorations in the Representation of knowledge*. J.F. Sowa editor, 1991.

[17] C. Boksenbaum, B. Carboneill, O. Haemmerlé, and T. Libourel. Conceptual graphs for relational databases. In G. Mineau, W. Moulin, and J.F. Sowa, editors, *Proceedings of the third International Conference on Conceptual Structures, ICCS'93*, volume 699 of *Lecture Notes in Artificial Intelligence*, pages 142–161, Quebec City, Canada, August 1993. Springer-Verlag.

[18] D. Nanci, B. Espinasse, B. Cohen, and H. Heckenroth. *Ingénierie des systèmes d'information avec Merise*. Performance. Sybex, ISBN: 2-7361-0747-7, 1992.

[19] J. Farradane. Relational indexing part i. *Journal of Information Science*, 1(5):267–276, 1980.

[20] J. Farradane. Relational indexing part ii. *Journal of Information Science*, 1(6):313–324, 1980.

[21] B. Nebel. Reasoning and revision in hybrid representation systems. volume 422 of *Lecture Notes in Artificial Intelligence*. Springer-Verlag, 1990.

[22] J.F. Sowa. *Conceptual Structures : Information Processing in Mind and Machine*. Addison-Wesley Publishing Company, 1984.

[23] K. Devlin. *Logic and Information*. Cambridge University Press, 1991.

[24] J.P. Chevallet. *Un Modèle Logique de Recherche d'Informations appliqué au formalisme des Graphes Conceptuels. Le prototype ELEN et son expérimentation sur un corpus de composants logiciels*. Phd thesis, Université Joseph Fourier, Grenoble, 1992.

[25] T.W.C. Huibers, I. Ounis, and J.P. Chevallet. Conceptual graphs aboutness. In P.W. Eklund, G. Ellis, and G. Mann, editors, *Proceedings of the 4th International Conference on Conceptual Structures, ICCS'96*, volume 1115 of *Lecture Notes in Artificial Intelligence*, pages 130–144, Sydney, August 1996. Springer-Verlag, Berlin.

[26] T.W.C. Huibers and P.D. Bruza. Situations, a general framework for studying information retrieval. *Information Retrieval: New systems and current research*, 2, 1994.

[27] M. Lalmas and C.J. van Rijsbergen. A model of an information retrieval system based on situation theory and dempster-shafer theory of evidence. In *Incompleteness and Uncertainty in Information Systems*, pages 102–116. Concordia University, 1993.

[28] Y. Chiaramella, F. Fourel, and P. Mulhem. Modelling multimedia structured documents. Chapter of deliverable D4, FERMI BRA 8134, 1996.

[29] I. Ounis and J.P. Chevallet. Applying MIRLOG to a model of multimedia documents. Chapter of deliverable D8, FERMI BRA 8134, 1996.

[30] J. Barwise. *The situation in logic*. Number 17 in Lecture Notes. Center for Study of Language and Information, 1988.

[31] M.E. Marron. On indexing, retrieval and the meaning of about. *Journal of the American Society for Information Science*, 28:38–43, 1977.

[32] M. Mechkour. *EMIR2. Un Modèle étendu de représentation et de correspondance d'images pour la recherche d'informations. Application à un corpus d'image historiques.* PhD thesis, Université Joseph Fourier, Grenoble, 1995.

[33] P. Palmer. *Etude d'un analyseur de surface de la langue naturelle: application à l'indexation automatique de textes*. PhD thesis, Université Joseph Fourier, Grenoble, 1990.

[34] P.D. Bruza. *Stratified Information Disclosure, a Synthesis between Hypermedia and Information Retrieval*. PhD thesis, Katholieke Universiteit Nijmegen, March 1993.

[35] M. Swan. *Practical English Usage*. Oxford University Press, Honk Kong, 1994.