An Information Retrieval Driven by Ontology: from Query to Document Expansion

Mustapha Baziz, Mohand Boughanem, Gabriella Pasi, Henri Prade

An Information Retrieval Driven by Ontology from Query to Document Expansion

Mustapha Baziz* & Mohand Boughanem* & Gabriella Pasi & Henri Prade***
* IRIT, 118 Route de Narbonne,
31062 Toulouse cedex 04 – France,
{baziz, boughanem, prade}@irit.fr

** Università degli Studi di Milano Bicocca,
Via Bicocca degli Arcimboldi 8,
20126 Milano, Italy,
gabriella.pasi@itc.cnr.it

Abstract

The paper proposes an approach to information retrieval based on the use of a structure (ontology) both for document (resp. query) indexing and query evaluating. The conceptual structure is hierarchical and it encodes the knowledge of the topical domain of the considered documents. It is formally represented as a tree. In this approach, the query evaluation is based on the comparison of minimal sub-trees containing the two sets of nodes corresponding to the concepts expressed in the document and the query respectively. The comparison is based on the computation of a degree of inclusion of the query tree in the document tree. Experiments undertaken on MuchMore benchmark showed the effectiveness of the approach.

Keywords

Information retrieval, concept-based retrieval, query/document completion driven by ontology.

1. Introduction

Textual Information retrieval (IR) is based on keywords or expressions (generally associated with importance weights) extracted from documents and employed as the index keys for both document and query representations: both are expressed in terms of (weighted) keywords [bruza 89]. However, keywords may have different levels of granularity. For instance, "earth science" is a more general expression than "geology". Thus, some may refer to general topics while others are more specific descriptors. It may also happen that terms which can be used for describing the general topic(s) of a document are not so much present in the document. Still they are useful for classifying the document and referring to its contents.

Recently, an increasing number of approaches to IR have defined and designed IR models which are based on concepts rather than keywords, thus modeling document representations at a higher level of granularity, trying to better the meaning and the context of the keywords rather than the "string words" that occur in the documents. These efforts gave raise to the so called concept-based Information retrieval, which aims at retrieving relevant documents on the basis of their meaning rather than their keywords. The main idea at the basis of conceptual IR is that the meaning of a text depends on conceptual relationships to objects in the world rather than to linguistic relations found in text or dictionaries (Thomopoulos et al., 2003). To this aim, sets of words, phrases, names are related to the concepts they encode (Haav et al., 2001).

In this paper, a concept-based information retrieval is proposed. Based on the existence of a conceptual hierarchical structure which encodes the contents of the domain to which the considered collection of documents belongs, both documents and queries are represented as

weighted trees. The evaluation of a conjunctive query evaluation is then interpreted as computing a degree of inclusion between sub-trees.
The ontology-based description of the contents of the documents takes into account the semantic equivalences between concepts, as well as the basic principle stating that if a document explicitly heavily includes some terms, it also concerns to some extent more general concepts. This latter point is handled at the technical level by a completion procedure which assesses positive weights also to terms which do not appear directly in the documents.
The paper is organized into six main sections. In section 2, a synthetic overview of some approaches to concept-based IR is presented. Section 3 details the proposed concept-based approach to IR in two subsections. The first sub-section (3.1) describes the concept-based representation of documents and queries based on a sub-tree representation of documents and queries whereas the second (3.2) details the process of query evaluation. Section 4 describes a novel method of completion based on a minimal common sub-tree and used both to complete the document and to enlarge the query. Section 5 describes the evaluation of the proposed approach. First, some details concerning the use of the approach in practice are presented (Detecting concepts from text and Pruning abstract nodes from documents/query representations). Then a summary of the experiments are reported and the results are discussed. Finally, some concluding remarks and perspectives are drawn in section 6.

2. Concept-Based IR: Related Works

The primary objective of the research in information retrieval is to define models, which allow designing effective IR systems, i.e. systems able to retrieve from the considered archive the documents concerned with the topics relevant to a user and expressed by a user query. The production of effective retrieval results depends on both subjective factors, such as the users' ability to express their information needs in a query, and the characteristics of the IR system. The indexing process plays a crucial role in determining the effectiveness of an IR system: in fact, to provide IR systems with powerful query languages or sophisticated retrieval mechanisms is not sufficient to achieve effective results if the representation of documents oversimplifies their information content. The indexing process has the aim of generating a formal representation of the contents of the information items (documents' surrogates). The most used automatic indexing procedures are based on term extraction and weighting: the documents are represented by means of a collection of index terms with associated weights (the index term weights); an index term weight expresses the degree of significance of the index term as a descriptor of the document information content. The automatic computation of the index term weight is based on the occurrence count of the term in the document and in the whole archive. In this case, a numeric weight is computed for each document d and each term t by means of an indexing function (Salton et al., 1984).
As a consequence, also query languages are usually based on (weighted) keywords, thus allowing the matching mechanism to compare two compatible representations. However, keyword-based retrieval models have several limitations; an important one is that they do not take into account the topical structure and content of documents, thus preventing concept-oriented document representation and query formulation.
Recently, some approaches have been proposed to concept-based Information retrieval. In concept-based IR, sets of words, names, noun phrases are mapped into the concepts they encode (Gonzalo et al., 1998). By these models, a document is represented as a set of concepts: to this aim a crucial component is a conceptual structure for mapping concepts to document representations. Conceptual structures can be general or domain specific. In (Gonzalo et al., 1998) an analysis of conceptual structures and their usage to improve retrieval is presented. Conceptual structures include dictionaries, thesauri and ontologies, and they can be either manually or automatically generated or they may pre-exist (Guarino et al., 1999). WordNet and

EuroWordNet are examples of (thesaurus-based) ontologies widely employed to improve the effectiveness of IR systems.
A conceptual structure can be represented using distinct data structures: trees, semantic networks, conceptual graphs, etc. In (Montes et al., 2000; Thomopoulos et al., 2003), the use of conceptual graphs for representing documents and queries is discussed. The authors propose a method for measuring the similarity of phrases represented as conceptual graphs. In (Gonzalo et al., 1998), the authors propose an indexing method based on the use of WordNet synsets: the vector space model is employed, by using synsets as indexing space instead of word forms. In a similar spirit in (Buitelaar et al., 2004), the authors have advocated the use of possibilistic ontologies, distinguishing between specialization and approximate synonymy relations, in information querying.
In (Chen et al., 1993), a concept based information retrieval approach based on the use of a thesaurus is proposed. In (Kan et al., 2001), an approach to detect the topical structure of a set of documents is presented. Probabilistic latent semantic analysis has been employed as a mean to define conceptual structures; it applies an unsupervised learning technique to define a semantic (topic-based) language model (Azzopardi et al., 2004; Ponte et al, 1998).
In this paper, we do not face the problem of generating a conceptual structure, we take as a preliminary assumption the existence of a conceptual structure (more precisely an ontology) describing the contents of a considered document collection. This structure is a hierarchical tree, as it will be explained in section 4. In the next section, we will describe how concepts are extracted and weighed from documents.

3. Concept-Based IR Driven by Ontology

The principle of the proposed approach aims at representing both documents and queries by means of sub-trees. A document/query is represented by a set of concepts corresponding to nodes in the hierarchical structure of ontology. Each node in the resulted sub-trees corresponds to a disambiguated term from a document/query that matches one concept of ontology. Sub-trees are obtained by considering only the "subsumption" relation represented in our case by a classical is-a relation (hypernymy). The idea behind this representation is to complete the document/query description by possibly adding intermediate nodes in order to complete these representations by concepts that do not appear explicitly in a document and/or a query but that deal somewhat with the same topic.
More details regarding the approach are given in the next sections.

3.1. Concept-Based Representation of Documents and Queries

In the proposed approach, the query evaluation of is mediated by an ontology made by a unique tree-like hierarchy H of concepts, which are supposed to be sufficient for describing the contents of the considered documents with an appropriate level of accuracy. Leaves in H can be thought as simple keywords, i.e. keywords expressing specialized concepts, while other nodes refer to keywords, which are labels of more general concepts. Edges in this hierarchy represent the classical *is-a* link.
Both documents and queries are supposed to be interpreted or expressed in terms of labels of nodes of *H*, possibly in association with weights.
Let d be a document. Each document d is identified by means of a set of pairs $R_d = \{(w_i, \alpha_i), i = 1, k(d)\}$ where w_i is a key word or phrase taken from d that corresponds in a univocal way to a concept c_i (node n_i) from *H* and α_i is its importance weight (index term weight), k(d) is the number of terms in document *d*.
The first step of the approach is to compute the projection H_d of *d* on H, in the following way:

a weighted subset *N(H, d)* of nodes of H, namely $N(H, d) = \{(n_j, \gamma_j), j = 1, m(d)\}$ where for any node n_j, there exists w_i in $d = \{(w_i, \alpha_i), i = 1, k(d)\}$ such that $n_j = w_i$, and n_j is known as the *most appropriate concept* (or node) (Baziz et al., 2005.1) that represents better w_i in the conceptual structure *H*, and then we take $\gamma_j = \alpha_i$. m(d) is the number of nodes in *H* that are equivalent to some terms in R_d. When several equivalent expressions w_i in *H* exist, the longest term is retained as described in section 5.2.1.
The second step of the approach is to build the minimal sub-tree H_d, of *H* which contains *N(H, d),* where the weights associated with the nodes are those obtained at step *i* if the nodes belong to *N(H, d)* and are 0 otherwise.
Let *q* be a query obtained by selecting a collection of labels (concepts) in *H*, with possibly an importance weighting, namely as a set $\{(l_k, \delta_k), k = 1, r(q)\}$. A query q is also modeled by a sub-tree H_q of H. Namely H_q is the minimal weighted sub-tree of *H* containing $\{(l_k, \delta_k), k = 1, r(q)\}$, keeping the weights δ_k, and putting 0 on the other nodes of H_q.
At these two stages the query and the documents are represented by sub-trees with weighted nodes.

3.2. Query Evaluation

Query evaluation is based on the comparison of two weighted subsets of nodes of H, one corresponding to the query and the other to the current document. The relevance of the query with respect to the document is interpreted as an inclusion degree which evaluates to what extent a document includes all the features of the query.
Various implication connectives in the definition of the inclusion degree is discussed.

3.2.1 Comparison Based on the Minimal Common Sub-Tree

The evaluation of a query q with respect to a document d, is performed in terms of a degree of relevance $rel_c(d; q)$ of *d* with respect to *q* computed as degree of inclusion of H_q into H_d, namely

$$rel_c(d; q) = f(\mu_{H_q*}(n) \rightarrow \mu_{H_d*}(n)),\ n \in H_E \qquad (1)$$

where $\mu_{H_d*}(n)$ (resp. $\mu_{H_q*}(n)$) is the weight associated with node n in H_d* (resp. H_q*), and $\rightarrow$ is a multiple-valued implication connective expressing that all the concepts of the query should appear in the description of the document.
The usual way to interpret *f* function is to use a conjunction aggregation *min*. One may think of introducing equivalence connectives in place of implications in (1) for requiring that the topic of the document correspond exactly to the topic of the query. However, note that looking for exact matching may be dangerous: suppose we are looking for documents dealing with topic A (q=A) but there does not exist any document dealing with A without B (d=A, B); in such a case the exact matching strategy will give nothing. However, strict equivalence could be relaxed into approximate similarity by weakening the equivalence connective by means of a similarity relation.
A strict conjunctive evaluation, *f=min,* may be too requiring, and in information retrieval, "best matching" is usually preferred to exact matching. Therefore, a simple function that is known to allow best matching is the sum computed as follows, and that is also used in this paper:

$$Sum : rel_d(d; q) = \sum_{n \in H_E} \mu_{H_q*}(n) \rightarrow \mu_{H_d*}(n) \qquad (2)$$

3.2.2 Choice of an Implication Connective

Several choices can be considered for the implication $\rightarrow$ used in (1), depending on the intended semantics of the weights in the query. A usual way to interpret this implication in IR consists in using a similarity function. Another possible method, borrowed from the fuzzy approach is to consider the Lukasiewicz implication connective. More details concerning the use of more implications connective that have clear semantics in a retrieval context are discussed in (Baziz et al., 2005.2) and (Pasi et al, 1999) (another comparison is also described in (Dubois et al., 1996) in a database context):

Lukasiewicz implication $a \rightarrow b = min\ (1,\ 1 - a + b)$,
namely $a \rightarrow b = 1$ *if* $a \leq b$ *and* $a \rightarrow b = 1 - a$ *if* $b = 0$.

3.2.3 Completing the Document Representation / Expanding the Query

Query expansion is one of the well-known approaches that was proved to be effective in IR. It usually consists in adding to the initial query terms that are related to those expressed by the user to better convey the topic of the information need.

However, document expansion, is in our knowledge rather not at all used in IR.

Our goal in the approach proposed here is precisely to propose a way that is able to complete (expand) the document descriptions in order to add concepts that are closely related to those expressed in the documents. The way we propose to complete the document (resp. the query) is to assign weights to the concepts that were added to the document (resp. the query) minimal sub-tree H_d (resp. H_q) is built. The interest of such process can have several reasons.

Regarding document *d*, if a node has a non-zero weight in H_d, we may think that a node which is an ancestor of the node in *H* is also somewhat relevant for the description of the document (even if its own weight in H_d, is zero or small). Then, we may think of "completing" H_d by computing updated weights in the following way. Let α_i^s and α_i^{s+1} denote weights at level s and s + 1 in the hierarchy (the root is at level 0). The idea to propagate the non-zero weights from the leaves to the root. More precisely it consists in recursively updating the weights of the nodes starting from the leaves by having the revised weights computed as:

$$\alpha_{i,rev}{}^s = max(\alpha_i^s,\ (max_i\ \alpha_{i,rev}{}^{s+1}) * disc(s)\ ),$$

where disc(s) is a discounting factor possibly depending on level s. Indeed, if a document includes many instances of the word 'cat', it clearly deals with 'pets' (the "father" of 'cat'), but to a smaller extent if the word 'pets' (or its synonyms) do not appear as much in the document. In order to control the number of nodes to be added to document/query descriptions, only the common ancestors of couples or triples of co-occurring words in a same document might be considered in the completion procedure.

Regarding the queries, the completion procedure of the weights may be motivated by a potential expansion of the query to less specific terms. Here, the use of a discounting factor will reflect the fact that documents dealing directly with the terms initially chosen should be preferred to more general documents. A similar idea has been used in (Thomopoulos et al., 2003) when dealing with fuzzy conceptual graphs for handling possibilistic information and fuzzy queries (however with a different interpretation for the weights in the fuzzy conceptual graphs leading to a different evaluation procedure).

The completion (expansion) process can be done according two approaches. The first one, called topical completion consists of weighting and/or reweighing the concepts (nodes) of H_d (resp. H_q). Thus, the document (query) is completed by concepts that are related directly to those appearing only in the document (resp. query). This approach is query independent it is done a priori.

The second approach called query based completion is contextual completion. It consists in adding and/or re-weighting concepts to the documents (resp. Query) considering the sub-tree H_E representing both H_d and H_q. More precisely, instead of considering H_d and H_q, the documents and the query are represented in the minimal non-weighted sub-tree which contains both H_d and H_q. Let H_d* and H_q* be the extensions of H_d and H_q on H_E putting zero weights on the nodes of $H_E - H_d$ and $H_E - H_q$ respectively. The completion process is done in the same manner than the previous one by assigning recursively weight to the added nodes. This approach is query dependent; it is processed during the query evaluation.

Remark. One might also think of expanding the query by introducing children of nodes present in q. In fact, this makes the query more demanding (at least if we keep unchanged the levels of importance for the labels present in the original conjunctive query).

4. Experiments

The aim of the experiments is to evaluate the effectiveness of the concept-based approach proposed here compared to classical IR approach. Especially two main contributions described in the paper are evaluated:

- How good is the concept-based approach compared to the classical one?
- How good is the completion of documents and/or queries compared to the classical?

The classical approach used in these experiments is based on a vector space model which is implemented in the Mercure system (Boughanem et al., 1998).
Below, we detail the experiments settings.

4.1. Document Collection

The test collection we used in these experiments is issued from the MuchMore project[1] (Boughanem et al., 1998). This collection contains 7823 documents (medical papers abstracts) obtained from the Springer Link web site, 25 topics from which the queries are extracted and a relevance judgment file which determines for each topic its set of relelant documents. These assesments were established by domain experts from Carnegie Mellon University, LT Institute.

4.2. Ontology

WordNet (Miller, 1995) is used as general purpose ontology. In WordNet, concepts are organized into taxonomies where each node is a set of synonyms (called synset) representing a single sense. Several semantic relationships between nodes are defined, denoting generalization, specialization, composition links, etc. We used only the concept hierarchy identified by the *ISA* relation. Using WordNet as ontology needs specific setting. Indeed the first stage is concepts node identifying and the second is the document (query) completion. These two operations are detailed below.

	AVERAGE NUMBER OF TERMS	
	Documents	Queries
ALL TERMS (Classical)	56,01	4,05
WORDNET TERMS ONLY	48,89	3,1
%	87%	77%

Table 1: Average number of terms per document and query covered by WordNet.

[1] http://muchmore.dfki.de

As the collection deals with the medical domain, the question of the suitability of WordNet for this kind of collections could be asked. Statistics carried out over the collection show that the vocabulary of the documents of the collection is almost covered by WordNet. Table 1 gives the cover rate. It can be seen that about 87% (respectively 77%) of terms used in documents (respectively queries) appear in WordNet.

4.3. Detecting Concepts from Text

The first stage of the proposed approach aims at identifying terms from a document to be connected to concepts (nodes) of ontology. This stage includes tokenizing a text into sentences; parsing each sentence; extracting from the parsing results all terms (singles and compounds) that belong to at least one entry of ontology and then selecting for each term, the appropriate entry (concept). Terms could be proper nouns like *"henry_kenneth_alfred_russell" or* noun phrases like *"academy_of_motion_picture_arts_and_sciences"*. In order to face the problem of morphological variation of terms, we first question the ontology using these words just as they are, and then we use their base forms. Moreover, the selected concept is associated with the longest multiword. Remind that in word combination, the order must be respected (left to right) otherwise we could be confronted to the syntactic variation problem (*"science library"* is different from *"library science"*).

It may arise that a given term possibly corresponds to several entries (concepts) in ontology (polysemy problem). In this case, the appropriate concept is selected according to a contextual disambiguation algorithm described in (Baziz et al., 2005.1) by carrying out similarity measures between all candidate concepts.

The extracted concepts are then weighted as following. The weight of a concept (node) n_i in a document d_j is:

$$Weight(n_i, d_j) = tf/max\_tf\_c$$

Where *tf* is the frequency of a concept n_i in a document d_j and *max_tf_c* is the maximal frequency of all the nodes overall the collection.

This weighting schema can be seen as very simplistic way to weight concept. Other methods for concept weighting are proposed in the literature, they use in general statistical and/or syntactical analysis (Baziz et al., 2005.1) (Croft et al., 1991), (Huang et al., 2001). Roughly, they add single words frequencies, multiply them or multiply the number of concept occurrences by the number of single words belonging to the concept.

It should be noticed that the documents (query) are represented only by the concepts that belong to the document (query) content. However, our approach may benefit from the specificity of WordNet by using all the Synsets related to a given concept of a document. This is another way to complete (expand) the document (query) by using synonyms. This case is also evaluated in this work.

4.4. Pruning Abstract Nodes

In practice, when completing a document or a query, not all intermediate nodes are added to the document (query) representation. Indeed, nodes located in the high level of the hierarchy are removed as they represent abstract concepts. This additional stage, which consists in a pruning method, uses two pieces of information in order to decide whether a node could be added to a document (query) representation. The first one is the position in the hierarchy (*depth*) of the head node of the sub-tree containing original nodes and the second one is the *length* (number of nodes) of the current sub-tree branch containing the candidate intermediate nodes to be added.

So, for a branch B_i of a given sub-tree, the number *Nb* of extra nodes to be added is given by the following formula:

$$Nb(B_i) = min\ [(length\ (B_i)\text{-}1 + depth)/2,\ length(B_i)\text{-}1] \quad (4)$$

A high value of *depth* means that a node is located near leaves and allows adding specific intermediate nodes, while a low value of *depth* permits to prune abstract nodes, as they are located in the immediate vicinity of the root.

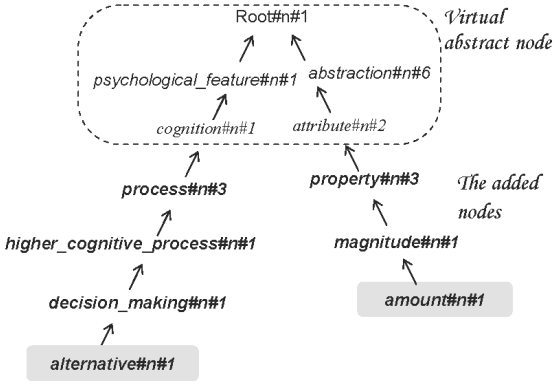

Figure 3.A sub-tree containing the nodes labeled alternative#n#1 and amount#n#1.

The remaining nodes are effectively more abstract and could be pruned from the document representation. They form with the Root node what we call a *virtual abstract node.*

4.5. Evaluation Methodology

In order to evaluate our approach, two sets of experiments were carried out. The first set is based on classical indexing and the second is based on the approach proposed in this paper. In the classical approach, the documents were first indexed using a classical term indexing. It consists in selecting single words occurring in the documents, and then stemming these words using Porter algorithm (Porter, 1998) and at the end removing stop-words according to a standard list (Salton et al., 1984). A weight is then assigned to each term following a kind of BM25 TF.IDF formula (Bordogna et al., 1995). The same process is applied to queries. A vector-based model (Bordogna et al., 1995) is then used to retrieve documents. We used this run as a baseline.

In the concept-based approach, the documents and the queries are indexed using a concept based detection described in 6. The result of this stage is that each document (query) is represented by a set of weighted concepts (ontology nodes). Once nodes representing the documents (queries) are identified, the corresponding sub-trees, (H_d) and (H_q) are built by using the pruning method. In these experiments the number *Nb* of extra nodes to be added is set to 3. Once the document and the query sub-trees are built, the query evaluation is carried out. It is based on *Lukasiewicz* implication using two aggregation functions namely, *min* and *sum*. More over, in order to evaluate the impact of the completion approach we only evaluated the topical approach, the query independent approach. The contextual approach is not scalable.

The experimental method follows the TREC protocol (Vorhees et al., 1997). For each query, the first 1000 retrieved documents are returned by the search engine and precisions are computed at different points P5, P10, P15 and P30 representing the mean precision values for the 20 used queries at the top 5, 10, 15 and 30 selected documents and MAP, the Mean Average Precision over the 20 queries.

4.6. Results and Discussion

4.6.1 Impact of the aggregation function

Table 1 evaluates the impact of the aggregation functions (min and sum), when no completion is used.

As it can be expected the aggregation function has a great impact. Indeed one can notice the strict conjunctive is clearly less interesting than the sum method at all considered precision levels. In fact, this function which performs an exact matching is mostly adapted to Data Retrieval (a document is retrieved if and only if it contains all query terms), whilst best matching is used in information retrieval (Rijsbergen., 1979). This is confirmed in these experiments, when best matching operators are considered, namely sum. However, one notices that the classical approach is slightly better than the concept-based on MAP level but the concept approach outperforms the classical at top level precisions (P5, 10 and 15).

Run	**Precisions**						
	P5	**P10**	**%**	**P15**	**P30**	**MAP**	**%**
Baseline(classical)	0.700	0.630	-	0.576	0.438	0.414	-
No comp min	0.420	0.310	-51%	0.233	0.190	0.144	-65%
No comp sum	0.760	0.660	5%	0.613	0.438	0.409	-3%

Table 1. Comparison of classical vs. concept-based approach

In fact, this has several reasons. The first one comes from the way the documents are indexed. Indeed only concepts belonging to the documents (queries) and the ontology are used as index terms. Nevertheless, it is usually admitted in IR (Baziz et al., 2005.1) that using only concepts from ontology as index terms are not sufficient to cover all the items of the documents (queries). So, our index lacks of exhaustivity. This explains the slight decreasing of the MAP. But, as the concepts are more specific than single terms, the index becomes more specific. This increases the precision at top documents. This is an interesting result. Indeed even though our weighting schema is quite rudimentary, our approach outperforms the classical one at top retrieved documents (with 5% improvement at P5 and more than 8% at P10).

4.6.2 Impact of the completion

Table 2 illustrates the results of the completion method when applied to documents and/or queries. Sum is used as aggregation function.

Run	**Precisions**						
	P5	**P10**	**%**	**P15**	**P30**	**MAP**	**%**
Baseline	0.700	0.630	-	0.576	0.438	0.414	-
▪**Docs** Comp : no ▪**Queries** Comp: no	0,760	0,660	5%	0.613	0.438	0.409	-3%
▪**Docs** Comp: no ▪**Queries** Comp: yes	0,700	0,650	4%	0,597	0,467	0,390	-6%
▪**Docs** Comp: yes ▪**Queries** Comp: no	0,770	0,690	**10%**	0,620	0,470	0,405	-2%
▪**Docs** Comp: yes ▪**Queries** Comp: yes	0,740	0,690	**10%**	0,653	0,510	0,440	**6%**

Table 2. Impact of documents and/or queries completion on retrieval accuracy

The first important result that can be drawn form Table 2 concerns the difference of the impact of completion when applied to queries or to documents. It seems that completing only the query is not beneficial in our method. The explanation is that queries contain often too few terms and the sub-tree built from the query may be less or better covered by the documents ones. When only documents are completed, the results seem differently. For instance, precision at top 10 brings 10% benefits compared to the baseline, +6% compared to the case where only query sub-trees are completed and +5% compared to the case where no completion is used. However MAP is still decreased (-2% compared to the baseline). So completing documents is better than completing queries because documents sub-trees contain much more nodes than the queries ones. So there is more chance to get cases where the completed document sub-tree covers the query one.
The last case, completing both documents and queries gives the best results. It represents also the core of our approach. In fact this completing method differs from the classical ones such as document/query expansion as in our case, the expansion is done according not only to the document and the query taken separately, but to a common sub-tree including both document and query descriptions. Thus, a same query may be expanded in a different way according to two different documents, and a document may be expanded differently using two different queries. This "contextual" expansion method seems to bring the best results. Indeed, precision at top 10 brings 10% enhancement compared to the baseline. Compared to the run with document completion only, the results at top documents are comparable, however the results at MAP are clearly better for the case where both documents and queries are completed: in this case MAP increased with 6% compared to the baseline whereas only document completion decreases the MAP with -2%. This last case is the only one which increased the MAP. So, our completion method brings benefits compared to the remaining methods and the baseline at both top documents and MAP.

4.6.3 Completion + synset

As we mentioned above, another way to complete a document (query) is to add to it all the Synsets of a considered concept (node). To illustrate, the node *decision_making#n#1* (in Figure 3) which represents the sense one of the term *decision_making* is defined in WordNet as following:

decision making, deciding -- (the cognitive process of reaching a decision; "a good executive must be good at decision making").

The synset (synonym set) of this node is {*decision making, deciding*}, the rest which is not used represents the definition of the node (between ()) with an example from real world (between *""*). So, adding Synsets in this case consists to add the term *decidin*g to the document/query which contains the node *decision_making#n#1*.
Table 3 summarizes the results corresponding to adding Synsets to the nodes of the document and/or query sub-tree with or without completion of documents and/or queries.

It can be seen that as in Table2, the case where only queries are completed decreases the results when Synsets are added to the queries (at top documents and MAP). This confirms the results obtained in Table 2 when non Synsets are added. In fact adding Synsets to query nodes that are not shared by document sub-tree decreases the results. When only documents are completed, adding synonyms of nodes to the document description seems more interesting at top documents (+13% benefits at top 10) but remains with no sensitive effect on MAP. When both the document and the query are completed according to a same referential as in Table 2 (minimal sub-tree covering both document and query descriptions as described in section 4), the results are clearly better when Synsets of nodes are added to the document description

(+16% at top 10) with a slight benefits when adding Synsets also to the enlarged query nodes (+17% at top 10).
These results are also the best concerning MAP (respectively + 9% and +8% benefits).
Using the case giving the best results (completing both documents and queries), two other parameters are tested in these last experiments.

Run	Precisions						
	P5	**P10**	**%**	**P15**	**P30**	**MAP**	**%**
Baseline	0.700	0.630	-	0.576	0.438	0.414	-
▪**Docs** Comp : no ▪**Queries** Comp: yes Synsets: yes	0.67	0.615	-2%	0.567	0.45	0.376	-9%
▪**Docs** Comp: yes ▪**Queries** Comp: yes Synsets: yes	0.75	0.67	6%	0.61	0.51	0.431	4%
▪**Docs** Comp: yes Synsets: yes ▪**Queries** Comp: no	0,77	0,715	13%	0,613	0,468	0,405	-2%
▪**Docs** Comp: yes Synets: yes ▪**Queries** Comp: yes	0,76	0,73	**16%**	0,677	0,512	0,450	**9%**
▪**Docs** Comp: yes Synsets: yes ▪**Queries** Comp: yes Synsets: yes	0,77	0,735	**17%**	0,69	0,523	0,449	**8%**

Table 3. Impact of combining completion and adding Synsets on retrieval accuracy

The first one (case (I) in Table 4) concerns the method the Synsets are added to the description nodes found in documents/queries. Indeed it can arrive that a Synset length may be too large (may arise a dozen of elements for some nodes), so we decide to limit the number of Synset elements (synonyms) to add to the document/query description nodes. In these last experiments, the threshold is fixed to 2. The second parameter concerns the use of words no belonging to ontology (case (II) in Table 4). As mentioned above, only terms belonging to ontology are used to build the documents and queries descriptions (87% of the overall vocabulary used in the documents and 77% of the queries vocabulary). So, one can think to add the remaining words to these descriptions before comparing them in order to increase the chance of finding more relevant matching. The added words are lemmatized using Porter and weighed using the same formula given in 5.2.1. The results of these two cases are given in Table 4.

Run	Precisions						
	P5	**P10**	**%**	**P15**	**P30**	**MAP**	**%**
(I) Completion for both docs and queries + Careful expansion (≤ 2 synonyms)							
	0,79	0,74	**17%**	0,687	0,547	0,469	**13%**
(II) Idem + Adding classical (terms no belonging to WordNet)							
	0,8	0,79	**25%**	0,696	0,533	0,48	**16%**

Table 4. Limiting the number of added synsets (Careful expansion) and adding non recognized terms

For the first case (I), the results show that a careful expansion method limiting the number of added synonyms enhances significantly MAP (+13%). However its positive impact is less important concerning precision at top documents (slight benefits at top 5 and no change at top 10). So this "careful expansion" method reduces the negative impact on MAP that can have a "blind expansion" method consisting of adding all the elements of a given Synset (all possible synonyms).

The second case (II) obtained by adding words that don't belong to ontology to the document and the query descriptions, we obtained the best results overall the cases. Indeed, precision at top 10 reaches +25% benefits compared to the baseline and MAP gives +16% enhancements. So the lack in the cover rate of the vocabulary used in documents and queries is corrected by adding the remaining words obtained by the classical indexing method. One can think that as long as this type of resources is not available, we still need to combine our concept-based approach with classical one in order to increase retrieval accuracy.

5. Conclusion

The work developed in this paper lies within the scope of the use of ontologies to concept-based indexing in information retrieval. We have introduced an approach which models both documents and queries as tree-like ontologies where nodes are weighted. The query evaluation process uses fuzzy connectives.

The preliminary experiments carried out on an IR collection indicate that the proposed approach is viable. The main interesting results that can be drawn from these experiments concern document completion. Indeed, most "of" IR "completion/expansion" approaches are used for query but never for documents. These preliminary experiments tend to indicate that completing documents and queries according to a common referential as suggested in the paper outperforms the classical keyword based approach.

Future works will focus on the evaluation of the concept-based approach on larger collection such the TREC collection (Vorhees et al., 1997).

Possible prospects within this work concern the use of the approach at inter-document level. Indeed, the sub-trees resulting from the projection of documents onto ontology could be compared for thematization. Thus, one can assume that documents with closest sub-trees could be regarded as covering a same subject. It remains to define the intersection function between two sub-trees.

6. References

Azzopardi, L., Girolami M. L., and C.J. van Rijsbergen. Topic Based Language Models for ad hoc Information retrieval. In *the Proceedings of the International Joint Conference on Neural Networks*, Budapest, Hungary, 2004.

Baziz, M., Boughanem, M., Aussenac-Gilles, N., Chrisment, C. Semantic Cores for Representing Documents in IR. In *Proceeding of the 2005 ACM Symposium on Applied Computing*, vol2 pp. 1011-1017, Santa Fe, New Mexico, USA, March 2005.

Baziz, M., Boughanem, M., Pasi G., Prade H., (2005). A Fuzzy Set Approach to Concept-based Information Retrieval. In : *the 4th Conference of the European Society for Fuzzy Logic and Technology and the 11ème Eleventh Rencontres Francophones sur la Logique Floue et ses Applications (Eusflat-LFA 2005 joint Conference)*, Barcelona, Spain, p. 1287-1292.

Bordogna G. and Pasi G., (1995). Linguistic aggregation operators of selection criteria in fuzzy information retrieval. *Int. J. of Intelligent Systems*, 10, 233-248.

Boughanem M., Dkaki, T. Mothe J and C. Soulé-Dupuy. Mercure at TREC-7. *In Proceeding of Trec-7*, (1998).

Buitelaar, P., Steffen D., Volk, M., Widdows, D., Sacaleanu, B., Vintar, S., Peters, S., Uszkoreit, H, (2004). Evaluation Resources for Concept-based Cross-Lingual IR in the Medical Domain. In *Proc. of LREC2004*, Lissabon, Portugal.

Chen, H., Lynch K. J., Basu K., and Ng D. T., (1993). Generating, integrating, and activating thesauri for concept-based document retrieval. *IEEE EXPERT, Special Series on Artificial Intelligence in Text-based Information Systems*, 8(2):25-34.

Croft, W. B., Turtle, H. R. & Lewis, D. D. (1991). The Use of Phrases and Structured Queries in Information Retrieval. In *Proceedings of the 4th Annual International ACMSIGIR Conference on Research and Development in Information Retrieval*, A. Bookstein, Y. Chiaramella, G. Salton, & V. V. Raghavan (Eds.), Chicago, Illinois: pp. 32-45.

Dubois D., Prade H., (1996). Semantics of quotient operators in fuzzy relational databases. *Fuzzy Sets and Systems*,78, 89-93.

Gonzalo J., Verdejo F., Chugur I., Cigarrán J., (1998). Indexing with WordNet synsets can improve text retrieval, in *Proc. the COLING/ACL '98 Workshop on Usage of WordNet for Natural Language Processing*.

Guarino N., Masolo C., Vetere G., (1999). OntoSeek: Content-Based Access to the Web. *IEEE Intelligent Systems*, pp 70-80

Haav, H. M., Lubi, T.-L., (2001). A Survey of Concept-based Information retrieval Tools on the Web. In *Advances in Databases and Information Systems, Proc. of 5th East-European Conference ADBIS*2001*, (A. Caplinkas and J. Eder, Eds), Vol 2., Vilnius "Technika", pp 29-41

Huang, X. and Robertson, S.E., (2001). Comparisons of Probabilistic Compound Unit Weighting Methods. In *Proc. of the ICDM'01 Workshop on Text Mining*, San Jose, USA.

Kan M. Y., Klavans J. L., McKeown K. R., (2001). Synthesizing composite topic structure trees for multiple domain specific documents, *Tech. Report CUCS-003-01*, Columbia University.

Miller G., (1995). Wordnet: A lexical database. *Communication of the ACM*, 38(11):39--41.

Montes-y-Gómez M., López-López A., and Gelbukh A., (2000). Information retrieval with Conceptual Graph matching. In *Proc. DEXA, 11th Int. Conf. on Database and Expert Systems Applications*, Greenwich, England. LNCS 1873, Springer-Verlag, pp. 312–321.

Pasi G., (1999). A logical formulation of the Boolean model and of weighted Boolean models. *Workshop on Logical and Uncertainty Models for Information Systems (LUMIS)*, University College London.

Ponte, J. M. and Croft W. B., (1998). A language modeling approach to information retrieval. In *Proceedings of the Twenty First ACM-SIGIR*, Melbourne, Australia. ACM Press, pp. 275–281.

Porter M., (1980). An algorithm for Suffix Stripping. *Program, Vol. 14(3)*, pp 130-137.

Salton G., and M.J. McGill. (1984). Introduction to Modern Information retrieval. *McGraw-Hill Int. Book Co*.

Thomopoulos R., Buche P., Haemmerlé O., (2003). Representation of weakly structured imprecise data for fuzzy querying. *Fuzzy Sets and Systems*, 140, 111-128.

Van Rijsbergen, C. J. (1979). Information retrieval. *Butterworths, London, 2. Edition*. http://www.dcs.glasgow.ac.uk/Keith/Preface.html.

Vorhees, E. M., and Harman, D. K., (1997) “Overview of the sixth Text REtrieval Conference (TREC-6)”, in *Vorhees, E. M., and Karman, D. K.* (eds.).