

A fuzzy set approach to concept-based information retrieval

Mustapha Baziz *

Mohand Boughanem*

Gabriella Pasi**

Henri Prade*

* IRIT, 118 Route de Narbonne, 31062 Toulouse cedex 04 – France, {baziz, boughanem, prade}@irit.fr

** ITC-CNR, Via Bassini 15, 20133 Milano, Italy, gabriella.pasi@itc.cnr.it

Abstract

In this paper an information retrieval approach is proposed based on the use of a fuzzy conceptual structure used both to index document and to express user queries. The conceptual structure is hierarchical and it encodes the knowledge of the topical domain of the considered documents. It is formally represented as a weighted tree. The evaluation of conjunctive queries 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 uses different multiple-valued degrees of inclusion, which are discussed. The proposed approach generalizes standard fuzzy information retrieval. Its evaluation is also presented.

1 Introduction

Textual Information Retrieval (IR) is based on keywords or expressions (generally associated with importance weights) extracted from documents and employed as the building blocks of both document and query representations: both are expressed in terms of (weighted) keywords. However, keywords may have different levels of generality. 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 that are based on concepts rather than keywords, thus, modeling document representations at a higher level of granularity, and trying to

describe the topical content and structure of documents. These efforts gave rise to 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 [7]. To this aim, sets of words, phrases, names are related to the concepts they encode.

In this paper, a fuzzy set approach to 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 query is then interpreted as computing a degree of inclusion between sub-trees.

The paper is organized into five main sections. In section 2 a synthetic overview of some approaches to concept-based IR is presented. In section 3 the proposed concept-based representations of documents and queries are introduced. In section 4 the process of the conceptual query evaluation is described. Finally, some experiments with their results discussion are drawn in section 5.

2 Concept-based information retrieval

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 as a bag of weighted terms [13]. As a consequence, also queries 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 IR. In concept-based IR, sets of words, names, noun phrases are mapped into the concepts they encode [7]. By this model a document is represented as a set of concepts: to this aim a crucial component is a conceptual structure for mapping document representations to concepts. Conceptual structures can be general or domain specific. In [7] an analysis of conceptual structures and their usage to improve retrieval is presented. These structures, including dictionaries, thesauri and ontologies [6], can be either manually or automatically generated, or they may pre-exist. WordNet and EuroWordNet are examples of (thesaurus-based) ontologies widely employed to improve the effectiveness of IR system. In [8] an approach to detect the topical structure of a set of documents is presented. In this paper, we do not face the problem of generating conceptual structures, we take as a starting point the existence of a conceptual structure (more precisely an ontology) describing the contents of a considered document collection.

3 Concept-based representation of documents and queries

In our approach, the query evaluation of textual documents is supposed to be 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 keywords, expressing specialised 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 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 and α_i is its importance weight (index term weight) computed by an occurrence-based indexing function that is usually employed in IR [13]. $k(d)$ is the

number of terms in document d . We then compute the projection H_d of d on H , in the following way:

- i) 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 $n_j = w_i$, or n_j is known as a *strict equivalent* of 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, it is assumed that they are counted as a unique expression. In this case the weight is computed by taking the maximum weight among those of the equivalent expressions w_i in H .

- ii) H_d is the minimal sub-tree 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.

N.B. If necessary, one may restrict $N(H, d)$ to those expressions for which γ_j is sufficiently high, thus specifying an acceptance threshold.

Let q be a query obtained by selecting a collection of labels (concepts) in H , with possibly an importance weighting, namely q is a set $\{(l_k, \delta_k), k = 1, r(q)\}$. We assume that the query is viewed as a (weighted) conjunction of the concept labels. A query q is also modelled 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 .

4 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. First, a minimal sub-tree of H containing both subsets is determined, then an inclusion degree is computed for evaluating to what extent a document includes all the features of the query. The use of various implication connectives in the definition of the inclusion degree is discussed. Lastly, some possible procedures for completing the description of the document or extending the query, by propagating weights in H , are discussed.

4.1 Comparison based on the minimal common sub-tree

Let H_E be 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 evaluation of a *conjunctive* 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 a degree of inclusion of H_q into H_d , namely

$$rel_c(d; q) = \min_{n \in H_E} \mu_{H_q^*}(n) \rightarrow \mu_{H_d^*}(n) \quad (1)$$

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.

4.2 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. Among the possible choices we just consider the following ones, as they have clear semantics in a retrieval context (see, e. g. [2] in a database context and [5] in an IR context):

- i) using Dienes implication $a \rightarrow b = \max(1 - a, b)$ amounts to view the $\mu_{H_q^*}(n)$'s (antecedent a) as levels of importance in the following sense. The impact of having n absent in document d ($\mu_{H_d^*}(n) = 0$) leads to an evaluation that will be upper-bounded by $1 - \mu_{H_q^*}(n)$, a bound which is higher as the importance $\mu_{H_q^*}(n)$ of n in the query is smaller. It is assumed that $\max_n \mu_{H_q^*}(n) = 1$ (at least one weight in the query is maximal)

- ii) using Gödel implication $a \rightarrow b = 1$ if $a \leq b$ $a \rightarrow b = b$ if $a > b$, it amounts to use the weight $\mu_{H_q^*}(n)$ as a demanding threshold, since the corresponding term in conjunctive aggregation (1) is equal to 1 as soon as $\mu_{H_d^*}(n)$ is larger than $\mu_{H_q^*}(n)$.

- iii) using Lukasiewicz implication $a \rightarrow b = \min(1, 1 - a + b)$, we mix the two above effects (namely $a \rightarrow b = 1$ if $a \leq b$ and $a \rightarrow b = 1 - a$ if $b = 0$).

As explained in [4], by delocalizing the weights and using the first implication, we can easily build an ordered weighted minimum (Owmin) aggregation, and model a query asking for the satisfaction of *most* of the terms of the query (rather than all).

Remark 1

Note that a prototypical document can be directly used in this approach as a query. If the document is itself present in the base, it will be retrieved with the maximal estimated degree of relevance 1. Thus, the approach is appropriate for handling a case-based querying process.

Remark 2 The evaluation of a *disjunctive* query q by a degree $rel_d(d; q)$ of a non-empty intersection between H_d and H_q can be computed as follows:

$$rel_d(d; q) = \max_n \min(\mu_{H_d^*}(n), \mu_{H_q^*}(n)) \quad (2)$$

This expresses that at least one of the important concepts of the query is somewhat relevant to the document.

Remark 3 One may think of introducing equivalence connectives in place of implications in (1) for requiring that the topic of the document corresponds exactly to the topic of the query. However, note that looking for exact matches 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.

Indeed in information retrieval best matching is usually preferred to exact matching. A simple function that allows for best matching which we will also evaluate in this paper is the sum:

$$rel_d(d, q) = \sum_{n \in H_E} \mu_{H_q^*}(n) \rightarrow \mu_{H_d^*}(n) \quad (3)$$

4.3 Completing the description of a document / Enlarging the query

In the above procedure, the weights both in H_d^* and H_q^* have been used as they are. However, it may be advisable to modify some of the zero weights both in the description of the document, and in the query, due to different reasons. Indeed, 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 is to recursively update the weights of the nodes starting from the leaves by having the revised weights computed as:

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

where $\text{disc}(s)$ 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 enlargement 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 [14] 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).

Remark 4 One might also think of enlarging 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).

5 Experiments and results

The aim of the experiment is to evaluate the effectiveness of 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 IRS [3]. We detail the experiments settings.

5.1 Document collection

The test collection we used in these experiments is issued from the MuchMore project¹. 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 relevant documents. These assessments were established by domain experts.

5.2 Ontology

WordNet [10] is used as a 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 determined by the *ISA* relation.

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. We noticed that about 87% (respectively 77%) of terms used in documents (respectively queries) appear in WordNet.

5.3 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 one 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 and at the end removing stop-words according to a standard list [13]. A weight is then assigned to each term [3]. The same process is applied to queries. A vector-based model [13] 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 as follows. Keywords (single

¹ <http://muchmore.dfki.de/> (last visited 02/03/05).

terms, phrases) are first extracted from each document (query) using an approach developed in [1]. The keywords are then projected onto WordNet. As these keywords could appear in different synsets, a disambiguation process developed in [1], which

Table 2. Comparison of classical /conceptual approaches.

	P5	P10	P15	P30	MAP	
(1) Classical (baseline)	0,7400	0,6400	0,6233	0,4900	0,4171	
(2) CO_Lukas_conj	0.4200	0.3100	0.2333	0.1900	0.1447	
(3) CO_Lukas_disj	0,1600	0,1550	0,1367	0,1483	0,1432	
(4)	No completion	0,7000	0,6300	0,5767	0,4383	0,3747
	Completion of docs.	0,7200	0,6400	0,5733	0,4350	0,3749
	Completion of queries	0,7200	0,6550	0,6267	0,4733	0,3977
	Both docs and queries are completed	0,7500	0,7100	0,6667	0,5150	0,4450

selects for each keyword a unique synset; is then applied. The result of this stage is that each document (query) is represented by a set of weighted synsets (the weight of the keyword is assigned to the corresponding synset).

Once nodes representing the documents (queries) are identified, the corresponding sub-trees, (Hd*) and (Hq*) are built by computing recursively the nodes subsuming the extracted ones until having a unique sub-tree covering all the concepts. It may arise that the highest nodes of the resulting sub-trees are located in the immediate vicinity of the root of ontology. As these nodes are abstract, they are not useful, so they are omitted using a pruning method.

Once the document and the query sub-trees are built, the query evaluation is carried out. It is based on Lukasiewicz, Gödel and Dienes implications using three aggregation functions namely, min, max and sum described above in Equations 1, 2 and 3 respectively. In order to measure the effectiveness of the concept-based approach two sets of results were built. The first set concerns the classical approach based on vector space model. The second set concerns the concept based results. In this latter several sets of results were built by combining the three following factors: the implication, the aggregation function and the completion procedure.

The experimental method follows the TREC protocol [15]. For each query, the first 1000 retrieved documents are returned by the search engine and precisions are computed at different points.

5.4 Results and discussion:

The most representative results are summarized in Table 2. It describes the 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.

The first run, *Classical*, describes the results of the classical approach, the second *CO_Lukas_conj* those combining Lukasiewicz and conjunctive aggregation function and the third, *CO_Lukas_disj*, those combining Lukasiewicz and disjunctive function. It can be seen that the precisions of the *Classical* are strongly better than those of *CO_Lukas_conj* and *CO_Lukas_disj* at all considered precision levels. Concerning conjunctive function, detailed results of *CO_Lukas_conj* (for each query) show that only the queries having relevant documents containing all terms are well evaluated, which is the case of 12 queries from the 20 used. Actually, 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 match is used in Information Retrieval [13]. In the case of disjunctive aggregation, the problem is the reverse. Indeed, in this case, when carefully analysing the set of retrieved documents, we can see that relevant documents are selected by *CO_Lukas_disj* but not ranked at the top. This is due to the fact that the disjunctive function is not suitable for rank-ordering. For instance, a document having query evaluation values (0.6, 0, 0, 0) is ranked better than another having much more common terms with the query but with lower weights (0.5, 0.5, 0.5, 0.5). When completion is used for queries and/or documents, the results (which are not given in Table2) of conjunctive and disjunctive cases are still very low compared to the classical approach.

Now when the sum is used as aggregation function (*CO_Lukas_sum*), the results look differently. It can be seen that the sum, when no completion is used, achieves better ranking than both disjunction and conjunction aggregation. The results are in the same class as the classical approach, with a slight

advantage for the classical approach. However, when completion is used to queries and/or to documents the precisions improve in all considered cases with a real benefit when both queries and documents are completed. Indeed, in this latter, all precision cut-off are better than those obtained with the classical method. It should be noticed that some results such using Dienes and Godel implications are not described in Table2. Actually, the results showed that the implication has no effect. The results obtained with Lukasiewicz are in the same scale than those of Godel and Dienes.

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 could be useful to identify important concepts that are not appearing explicitly in both document and query, providing thus, a contextual search capability.

6 Concluding remarks

This paper proposes an approach to concept-based IR which models both document 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 in IR. The results showed that the concept-based approach outperforms the classical IR one based on vector space model. Future works concern firstly, the evaluation of the concept-based on larger collection such the TREC collection [1].

References

- [1] Baziz, M., Boughanem, M., Aussenac-Gilles, N., Chrisment, C., "Semantic Cores for Representing Documents in IR". In Proc. of the 2005 ACM Symp. on Applied Computing, vol2 pp. 1011-1017, Santa Fe, New Mexico, USA, March 2005.
- [2] Bordogna G. and Pasi G.. Linguistic aggregation operators of selection criteria in fuzzy information retrieval. *Int. J. of Intelligent Systems*, 10, 233-248, 1995.
- [3] Boughanem M., Dkaki, T. Mothe J and C. Soulé-Dupuy "Mercure at TREC-7". In Proceeding of Trec-7, (1998).
- [4] Dubois D., Prade H.. Semantics of quotient operators in fuzzy relational databases. *Fuzzy Sets and Systems*,78, 89-93, 1996.
- [5] D. Dubois, M. Nakata, H. Prade. Extended divisions for flexible queries in relational databases. In: *Knowledge Management in Fuzzy Databases*, (O. Pons, M. A. Vila and J. Kacprzyk, Eds.), Physica-Verlag, Heidelberg, Allemagne, 105-121, 1999.
- [6] Guarino N., Masolo C., Vetere G., OntoSeek: Content-Based Access to the Web. *IEEE Intelligent Systems*, May/June 1999, pp 70-80
- [7] Haav, H. M., Lubi, T.-L A Survey of Concept-based Information Retrieval Tools on the Web. In Proc. of 5th East-European Conference ADBIS*2001, (A. Caplinkas and J. Eder, Eds), Vol 2., Vilnius "Technika" 2001, pp 29-41
- [8] Kan M. Y., Klavans J. L., McKeown K. R.. Synthesizing composite topic structure trees for multiple domain specific documents, Tech. Report CUCS-003-01, Columbia University, 2001.
- [9] Loiseau Y., Prade H. and Boughanem M. Qualitative pattern matching with linguistic terms. *AICom*, 17(1), 25-34, 2004.
- [10] Miller G. Wordnet: A lexical database. *Communication of the ACM*, 38(11):39--41, (1995).
- [11] Montes-y-Gómez M., López-López A., and Gelbukh A.. Information retrieval with Conceptual Graph matching. In Proc. DEXA-2000, 11th Int. Conf. on Database and Expert Systems Applications, Greenwich, England, September 4-8, 2000. LNCS 1873, Springer-Verlag, pp. 312–321.
- [12] Porter M., "An algorithm for Suffix Stripping", *Program*, Vol. 14(3), pp 130-137, 1980.
- [13] Salton G., and M.J. McGill, *Introduction to Modern Information Retrieval*. McGraw-Hill Int. Book Co., 1984.
- [14] Thomopoulos R., Buche P., Haemmerlé O. Representation of weakly structured imprecise data for fuzzy querying. *Fuzzy Sets and Systems*, 140, 111-128, 2003.
- [15] Vorhees, E. M., and Harman, D. K., "Overview of the sixth Text REtrieval Conference (TREC 6)", in Vorhees, E. M., and Karman, D. K. (eds.), *Proceedings of the Sixth Text Retrieval Conference (TREC-6)*, 1998, forthcoming.