Electronic Publishing '01 – 2001 in the Digital Publishing Odyssey
A. Hübler et al. (Eds.)
IOS Press, 2001

SyDoM: A Multilingual Information Retrieval System for Digital Libraries

Catherine Roussey, Sylvie Calabretto, Jean-Marie Pinon

LISI – INSA de Lyon
Bâtiment Blaise Pascal
7, avenue Jean Capelle
69621 Villeurbanne Cedex
France
Tel: (33) 4 72 43 88 94
Fax: (33) 4 72 43 87 13
catherine.roussey@lisi.insa-lyon.fr, sylvie.calabretto@lisi.insa-lyon.fr, jean-marie.pinon@lisi.insa-lyon.fr

Abstract: In this paper, we present a multilingual information retrieval system based on knowledge representation model. This system allows document indexing and information retrieval in a multilingual document collection where documents are written in different languages, though each individual document may contain text in only one language. The underlying model permits to describe the semantic of document in a multilingual context. This model, called semantic graph, is an extension of the Sowa's model of conceptual graphs where different vocabularies are available. Indeed, in this model two kinds of knowledge are identified:

- Domain knowledge organises domain entity in two hierarchies of types (concept types and relation types),
- Lexical knowledge associates term, belonging to a vocabulary, to concept type or relation type.

Then, a same semantic graph can have different representations depending on the vocabulary chose. For example, the French representation of a semantic graph uses the French labels of types to display each graph component and the English representation of the same graph uses the English labels of types to display the graph.

Our proposition has been validated in the logical information retrieval system SyDoM. The system is dedicated to the needs of virtual libraries for managing XML documents. Thanks to the semantic graph model, SyDoM develops several functionality for multilingual information retrieval. This first evaluation gives better results than traditional information retrieval system.

Keywords: Multilingual Information Retrieval, information modelling, XML document, knowledge representation, Conceptual Graph, Digital Libraries

0. Introduction

Doc'INSA is the library of a scientific and technical university called INSA (National Institute of Applied Sciences) located in Lyon. This library offers various types of scientific and technical documentation and provides students and researchers with local resources in the engineering sciences: 85,000 books and 1,740 periodicals. The collection is divided into different topics such as computer science, mechanics, etc. Currently, Doc'INSA is working on the design of a Digital Library that allows its subscribers to retrieve and consult documents alongside the Internet network.

The indexing method in Doc'INSA involves a librarian in building manually a bibliographic record for each document. This bibliographic record is composed of several fields: the title of the document, the authors' name, the publishing date, keywords and an abstract. The keyword field contains a list of keywords describing the document content. Later, the librarian writes an abstract, which will be displayed after the retrieval procedure in order to obtain a more accurate description of the document.

The Information Retrieval system of Doc'INSA uses a Boolean model where a query is composed of keywords associated by Boolean operators. All bibliographic records whose keywords field matches the Boolean query, are retrieved and displayed.

Unfortunately, this indexing method has several limitations:

- Firstly, a document is represented by a list of keywords. Therefore, this method does not provide an accurate and precise description.
- Secondly, there is no control of keyword assignment which means that two documents dealing with the same subject could be indexed by a different set of keywords. Consequently, librarians can not handle correctly the vocabulary evolution.
- Finally, this indexing method is monolingual and builds only French indices. French indices are not adapted to the needs of a multilingual collection of documents. For example, a translation process is necessary to query in English these French indices.

In order to improve the indexing method of Doc'INSA, we propose a semantic graph model adapted for multilingual information retrieval (IR) purpose. This model combines domain knowledge and lexical knowledge, used to present domain knowledge in several languages such as French or English. Indices will be written in a language independent way thanks to the domain knowledge.

First of all, we present in chapter 2 a review of conventional Multilingual Information Retrieval approaches. Next, in chapter 3, we propose a formal presentation of our semantic graph model, following by our research algorithms (chapter 4). As shown in chapter 5, this graph model permits to develop several functionality for our multilingual information retrieval system SyDoM. And finally, we compare SyDoM with the Doc'INSA information system.

1. Related works

The major part of Cross-Language Information Retrieval approaches translate a query into target languages, using bilingual dictionaries [1], [8], bilingual corpora [4] or Machine Translation systems [12]. These approaches use monolingual Information Retrieval system, so they still deal with monolingual indexing.

Dumais & Co [5] have proposed a method of language-independent indexing based on parallel corpus. They have used a matrix reduction technique called Latent Semantic Indexing (LSI) to extract langage-independent terms and document representations. Unfortunately, parallel corpora are not common in general, tending rather to be restricted to specialised domains like legal domain of multilingual country (Switzerland, Canada).

P. Vossen & Co [17] describe a language-independent indexing on the basis of the multilingual database EuroWordNet (EWN). A Wordnet is a network of word meanings and one Wordnet per languages is created (English, Dutch, Spanish, Italian, German, French, Czech and Estonian). All Wordnets are connected by the Inter-Lingual-Index (ILI) which contains a set of universal concepts called ILI records.

Their aim is to index documents and queries in terms of language-independent ILI records. Unfortunately, the French wordnet is still under development.

In conclusion, we are aware of only few methods using real language-independent indexing methods. Moreover these methods represent documents as a traditional vector. So indices are a linear combination of terms. We proposed a semantic graph model in order to improve the document representation in a multilingual information system.

2. Semantic graph model

Our proposition takes Conceptual Graph approaches into account [6], [7], [11], by proposing a graph matching function optimized for the information retrieval needs. First of all, we present a graph formalism allowing the document description in a multilingual context. We have transformed the conceptual graph formalism to be closer to a documentary language. Concepts are limited to generic concepts because descriptors represent notions and not individuals or particular objects. We consider that processing a graph is equivalent to processing its normal form, in which a concept node appears only one time per graph. Thus, the number of possible indices for a document is limited.

2.1 Semantic thesaurus

We proposed a formalism in which two kinds of knowledge are identified:

- Domain knowledge organises domain entity in two hierarchies of types (concept types and relation types). This knowledge is formalised in a support. Domain knowledge are considered to be language independent entity.
- Lexical knowledge associates term, belonging to a vocabulary, to concept type or relation type. Lexical knowledge displays domain knowledge in an understandable way for human users.

Domain and lexical knowledge are associated in a semantic thesaurus. This thesaurus defines a multilingual documentary language.

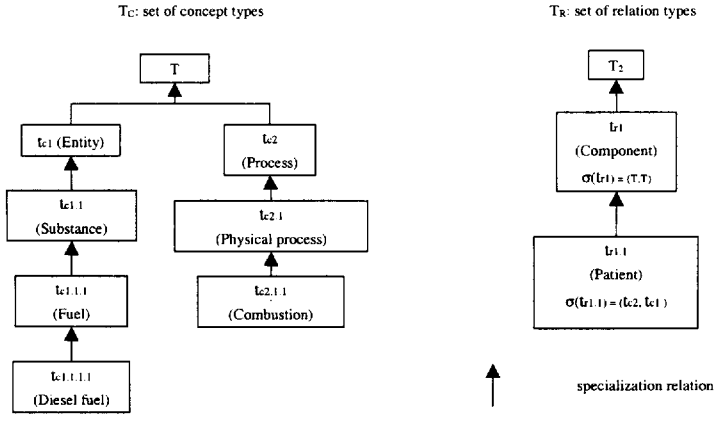

Fig. 1. An example of support.

2.1.1 Domain conceptualization or Support

A support S is a 3-tuple $S = (T_C, T_R, \sigma)$ such as:

- T_C is a set of concept types partially ordered by the specialization relation, noted $\leq$, and it has a greatest element, denoted T.
- T_R *is a* set of binary relation types[1] partially ordered by $\leq$ and it has a greatest element, noted T_2.
- σ, called *signature*, is a mapping which associates with any relation types the greatest concept type of its arguments. In a semantic graph, for any $t_r \in T_R$, the type of the k^{th} argument of t_r should be more specific than the type of the k^{th} argument of $\sigma(t_r)$. The i^{th} argument of $\sigma(t_r)$ is denoted $\sigma_i(t_r)$. The figure 1 contains two examples of signatures.

2.1.2 Semantic thesaurus

A semantic thesaurus, noted M, (composed of P languages) is a 4-tuple $M = (S, V, \lambda_C, \lambda_R)$ such as :

- S is a support composed of a set of concept types T_C , a set of relation types T_R, and a mapping σ which associates signatures with relation types.
- V is a set of vocabularies, split into set of terms belonging to the same language (a vocabulary). $V = V_{L1} \cup V_{L2} \cup ... \cup V_{Lj} \cup ... \cup V_{LP}$ such as V_{Lj} is a set of terms belonging to the language Lj.
- $\lambda_C = \{ \lambda_C^{VL1} ... \lambda_C^{VLj} ... \lambda_C^{VLP} \}$ is a set of P mapping such as $\lambda_C^{VLj} : T_C \rightarrow V_{Lj}$ is a mapping $\lambda_C^{VLj}(t_c)$ which associates a term of the language $Lj \in V_{Lj}$ with a concept type $t_c \in T_C$.
- $\lambda_R = \{ \lambda_R^{VL1} ... \lambda_R^{VLj} ... \lambda_R^{VLP} \}$ is a set of P mapping such as $\lambda_R^{VLj} : T_R \rightarrow V_{Lj}$ is a mapping $\lambda_R^{VLj}(t_r)$ which associates a term of the language $Lj \in V_{Lj}$ with a relation type $t_r \in T_R$.

[1] In general, a type of relation can have any arity, but in this paper, relations are considered to be only binary relations like case relations or thematic roles associated with verbs [11].

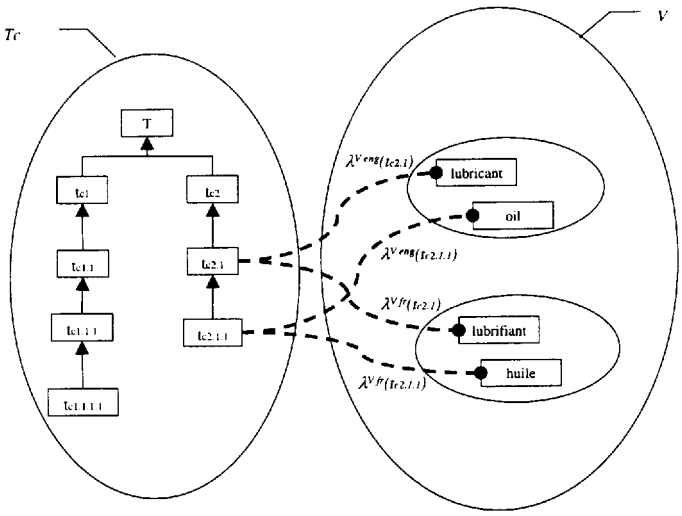

Fig. 2. An example of semantic thesaurus.

The figure 2 presents an example of mapping λ_C , in which V is composed of two vocabularies: an English vocabulary, denoted V_{eng}, and a French vocabulary , denoted V_{fr}. Each concept type is linked to a term of each vocabulary. For example, the type of concept $t_{c2.1} \in T_C$, is linked to the English term $\lambda_C^{V_{eng}}(t_{c2.1})$*="lubricant"* and it is also linked to a French term $\lambda_C^{V_{fr}}(t_{c2.1})$*="lubrifiant"*.

From this semantic thesaurus defining domain knowledge and lexical knowledge, our formalism, called semantic graph, is defined. A semantic graph is a set of concept nodes connected to each other by relations. Comparing to conceptual graph, the accent is made upon relations linking concepts. That's why we define the notion of arc as a couple of concept nodes labeled by a relation type.

Semantic graph

A semantic graph is a 6-tuple *Gs = (C, A, μ, labelC, ν, labelR)* related to a semantic thesaurus *M*, such that :

- *C* is a set of concept nodes[2] contained in *Gs*.
- $A \subset C \times C$ is a set of arches contained in *Gs*. For each arch $a = (c, c') \in A$; the ith concept node of *a* (also called argument of *a*) is denoted a_i: $(a_1 = c$ and $a_2 = c')$.
- μ: $C \rightarrow T_C$. μ is a mapping which associated for each concept node, $c \in C$, a label $\mu(c) \in T_C$, $\mu(c)$ is also called the ***type*** of *c*.
- *labelC* is a set of mappings *labelC* = { $labelC^{VL1}$, ... , $labelC^{VLj}$, ... , $labelC^{VLP}$} such that the mapping $labelC^{VLj}$: $C \rightarrow V_{Lj}$, associates a concept, $c \in C$, with a term of the language *Lj*, $labelC^{VLj}(c) \in V_{Lj}$. $labelC^{VLj}(c)$ is called the ***label*** of *c* for the language *Lj*.
- ν: $R \rightarrow T_R$. ν is a mapping, which associated for each arch, $a \in A$, a label $\nu(a) \in T_R$. $\nu(c)$ is also called the ***type*** of *a*.
- *labelR* is a set of mapping *labelR* = { $labelR^{VL1}$, ... , $labelR^{VLj}$, ... , $labelR^{VLP}$ } such that the mapping $labelR^{VLj}$: $A \rightarrow V_{Lj}$ associates an arch, $a \in A$ with a term of the language *Lj*. $labelR(a)^{VLj} \in V_{Lj}$. $labelR^{VLj}(a)$ is called the ***label*** of *a* for the language *Lj*.

[2] In this article, "concept node" and "concept" are equivalent expressions.

A semantic graph fulfils some constraints :

1. *A* fulfils the constraints fixed by the mapping σ, defined in the semantic thesaurus *M*. For each $a \in A$ such that $a = (c, c')$ and $\nu(a)=r$ then $\mu(a_i) \leq \sigma_i(r)$ so $\mu(c) \leq \sigma_1(r)$ and $\mu(c') \leq \sigma_2(r)$. For example, the relation type *Patient* defined in the Figure 4 has a signature such that $\sigma(Patient)=(Process, Entity)$. So all the relation typed by *Patient* should have a first argument, which concept type is more specific than *Process*, and a second argument, which concept type is more specific than *Entity*.
2. *LabelC* is the association of two mappings μ and λ_C. So, for all $c \in C$, $labelC(c)= \lambda_C(\mu(c)) = \lambda_C \circ \mu(c)$.
3. *labelR* is the association of two mappings ν and λ_R. So, for all $a \in A$, $labelR(a)= \lambda_R(\nu(a)) = \lambda_R \circ \nu(a)$.

Thanks to previous definition there exists different representations for the same semantic graph depending of the label used.

1. The first representation of semantic graph labels each graph component with its type. For example, in the semantic graph presented in the figure 3, the first concept node *c* is labeled by its type $\mu(c)= t_{c2.1.1}$. This type is defined in the support of the figure 1.

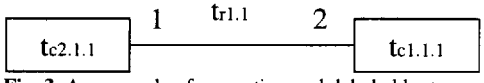

Fig. 3. An example of semantic graph labeled by types.

2. The second kind of semantic graph representation labels each graph component with a term chosen in a vocabulary defined in the semantic thesaurus. For example, if the language used is English, only the terms belonging to English vocabulary, noted *Veng* (cf Fig. 2), are used to label the graph component. So for a concept node *c*, its label is $labelC^{Veng}(c)= \lambda_C^{Veng}(\mu(c))$. Indeed, there exists several representations of the same semantic graph depending of the vocabulary chose. The figure 4 presents two representations of the semantic graph defined in the figure 3. In the first one, all the graph components are labeled by French terms and in the second representation, graph components are labeled by English terms. As a consequence, index composed of semantic graph can be displayed in different languages.

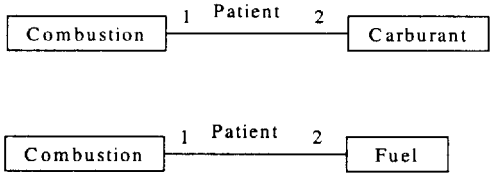

Fig. 4. Examples of semantic graph labeled by terms.

Now, we presents the similarity function between graphs used as a matching function in our multilingual information retrieval.

Similarity functions

Information retrieval systems based on Sowa 's Conceptual Graph use the projection operator as a matching function between index graphs of documents and query graphs. But this operator gives boolean results: projection between graphs

exists or does not exist. To use this operator as a matching function in a IR system, the projection operator should better returns a float value (ranging between 0 and 1) thus making it possible to order the results. To this end, we will present various similarity functions used to define a similarity function between graphs. Each similarity function (between types, arches, graphs) returns a normalized float value ranging between 0 and 1.

First, from specialization relation presented in the semantic thesaurus, similarity functions between types will be defined.

2.1.3 Similarity function between types

The similarity function, noted *sim*, between types is an asymmetrical function. *sim* is defined as follows:

$sim : T_C \times T_C \cup T_R \times T_R \rightarrow [0,1]$

- If two types are not comparable then the similarity function returns a value equal to 0.
- If two types are identical then the similarity function returns a value equal to 1.
- If a type t_1 generalizes another type t_2 directly, i.e. there is not intermediate type between t_1 and t_2 in the type hierarchy, then the similarity function returns a constant value lower than 1 equal to V_G. V_G is fixed arbitrarily.
- If a type t_1 specializes another type t_2 directly, i.e. there is not intermediate type between t_1 and t_2 in the type hierarchy, then the similarity function returns a constant value lower than 1 equal to V_S. V_S is fixed arbitrarily.
- If a type t_1 specializes or generalized another type t_2 not directly, i.e. there is an intermediate type t between t_1 and t_2 in the type hierarchy then the similarity function between t_1 and t_2 is the product of the similarity functions between (t_1, t) and (t, t_2).

According to similarity function between types, we defined similarity function between arches and graphs.

2.1.4 Similarity function between arches

The similarity function between two arches, noted Sim_A is defined as follows:

a_H is an arch such as $a_H = (c_H, c'_H)$ and $v(a_H) = t_{rH}$ and a_G is an arch such as $a_G = (c_G, c'_G)$ and $v(a_G) = t_{rG}$.

$$Sim_A\ (a_H, a_G) = \frac{sim(v(a_H), v(a_G)) + sim(\mu(a_{H1}), \mu(a_{G1})) + sim(\mu(a_{H2}), \mu(a_{G2}))}{3}$$

Or

$$Sim_A\ (a_H,\ a_G) = \frac{sim(t_{rH},\ t_{rG}) + sim(\mu(c_H),\ \mu(c_G)) + sim(\mu(c'_H),\ \mu(c'_G))}{3}$$

Sim_A computes the average of the similarity between each type of arch component.

2.1.5 Similarity function between graphs

The similarity function, noted sim_G, between a graph $H = (C_H, A_H, \mu_H, labelC_H, v_H, labelR_H)$ and a graph $G = (C_G, A_G, \mu_G, labelC_G, v_G, labelR_G)$ is a function returning float value ranging from 0 to 1.

$$sim_G(H,G) = \frac{\sum_{a_H \in A_H, a_G \in A_G} Max(sim_A(a_H,\ a_G)) + \sum_{c_H \in C_H, c_H \in C_G} Max(sim_C(\mu_H(c_H),\ \mu_G(c_G)))}{|C_H| + |A_H|}$$

sim_G computes the average of the similarity function between each graph component (arch and concept node) of H and a graph component of G. Because a graph component of H can have several comparable components in G, we take the maximum of the similarity function between a component of H and a component of G.

3. Information retrieval algorithm

After introducing our semantic graph formalism, we shall concentrate on the implementation of the matching function between graphs. Search algorithms find documents indexed by a semantic graph, called **index graph**, comparable with the query graph.

The semantic graphs are composed of arches and concept nodes. So there are two types of indexing entities (arches and concepts). Thus the document content is represented by two different indices: a list of arches and a list of concepts, from which the index graph of a document can be rebuilt.

Following the works of Ounis [11], our algorithms are based on the association of inverted files and acceleration tables. The inverted file groups in the same entry all the documents indexed by an indexing entity. This means that, given an indexing entity, we can immediately locate the documents indexed by it. The acceleration tables store, for each indexing entity, the list of comparable entities as well as the result of the similarity function between the comparable entity and the indexing entity. The construction of the inverted file and the acceleration table is done off-line, as part of the indexing procedure.

Our database is divided in two parts depending of the indexing entities used.

1. In the first part of the database, only arches of semantic graphs are taken into account. Each document is indexed by arches contained in its index graph. This part of the database is composed of an inverted file denoted *InvertedFileArc* and several acceleration tables denoted *FirstArgValue*, *SecondArgValue*, *RoleArgValue*.
2. In the second part of the data base, only the concept nodes of the semantic graphs are taken into account. Each document is indexed by the list of concepts contained in the index graph. This part of the database is composed of an inverted file, denoted *InvertedFileConcept* and a acceleration table, noted *ConceptValue*.

We present the principles of the search algorithm about arches. The search algorithm about concepts follows the same principles.

VARIABLES

GraphReq is a query graph composed of **nbArc** arches, noted **ArcReq**, and of **nbConcept** concept nodes

ListDocResult is a list of documents weighted by the value of the similarity function between the query graph **GraphReq** and the index graph of document.

ListArcIndex is the list of **ArcIndex** (arch indexing document) weighted by **WeightArc** the value of the similarity function between **ArcReq** and **ArcIndex**

ListDoc is the list of document **Doc** whose index contains **ArcIndex**

ListDocArc is the list of document **Doc** weighted by the value of the similarity function between **ArcReq** and their index graph.

```
ALGORITHM
ListDocResult ← empty
For each arch ArcReq of GraphReq do
   ListArcIndex ← empty
   ListDocArc ← empty
   ListArcIndex ←FindArcComparable(ArcReq)
   For each(ArcIndex, WeightArc) of ListArcIndex do
      ListDoc ← empty
      ListDoc ← FindListDoc(ArcIndex)
      // For an ArcReq, the document weight is the max
      of the similarity between ArcReq and an arch of
      document index
      For each Doc of ListDoc do
         If ListDocArc.Belong(Doc)
         Then
            Weight ←ListDocArc.FindWeight(Doc)
            NewWeight ← max(Weight, WeightArc)
            ListDocArc.ReplaceWeight(Doc, NewWeight)
         Else
            ListDocArc.Add(Doc, WeightArc)
         Endif
      Endfor
   Endfor
// Compute the final Weight of document.
   For each (Doc, WeightArc) of ListDocArc do
      If ListDocResult.Belong(Doc)
      Then
         Weight ← ListDocResult.FindWeight(Doc)
         NewWeight ← Weight + (WeightArc / (nbArc + nbConcept))
         ListDocResult.ReplaceWeight(Doc, NewWeight)
      Else
         ListDocResult.Add(Doc, WeightArc)
   Endfor
Endfor

FindArcComparable(ArcReq) returns a list of arches (ArcIndex)
comparable to ArcReq, weighted by the value of the similarity function
between ArcReq and ArcIndex, noted WeightArc.
```

In the search algorithm, the result documents are obtained in polynomial time, because most parts of the algorithm consist in union and intersection. Usually, the cost of a mapping between graphs is prohibitive because, in graph theory, it is equivalent to find a morphism between indefinite structures. To overcome this problem, we have limited the graph structure: we consider that a concept node is unique in a semantic graph.

4. The SyDoM prototype

We have developed a logical information retrieval module based on semantic graph. This module is a component of the documentary system called SyDoM (Multilingual Documentary System) [13]. SyDoM is dedicated to the needs of digital libraries for managing XML documents. The system is implemented in JAVA on top of the relational database system Access. The semantic thesaurus is the core of our system. Its goal is to formalize the conceptualization of the librarian's point of view for a particular domain. This thesaurus is then enriched by the various vocabularies contained in the document collection. This specific conceptualization will be presented to the various users in the language of their choice. SyDoM is composed of three modules:

1. The **semantic thesaurus module** manages the documentary language (addition of new vocabulary or new domain entity). Documentary language is used for indexing and querying a multilingual document collection. Actually the semantic thesaurus can be displayed in different languages (French and English).
2. The **indexing module** indexes and annotates XML document with semantic graphs using a set of metadata associated to the semantic thesaurus.
3. The **retrieval module** performs multilingual retrieval. The users introduce their queries through the query interface presented in figure 8. This figure corresponds to the French query "*modèle de combustion*" (combustion model).

Indexing module

Our indexing method enrichs XML documents with two types of metadata:

1. XML documents are annotated in order to identify significant terms. Annotations have two functions: first of all, language dependent expressions are linked with domain concepts, that is to say terms are identified as labels of concepts and is used to update one of the vocabulary of the semantic thesaurus. Secondly, the set of annotations generates an index composed of a list of concepts.
2. Starting from the annotation index, librarians refine it. This new index is composed of concepts linked by relations. This index is a precise and short description of the document content. Users can estimate the purpose of the document when reading it, and compare it to their information needs.

4.1 Document annotations

The document annotations specify the expression of particular concepts inside the document. So, a new XML element called **term** is added inside the XML structure of the document. For example, we want to annotate the XML document whose title is "A practical approach to jet engine lubricant evaluation". This XML document is:

```
<Document>
<Title>A practical approach to jet engine lubricant evaluation</Title>
<Introduction>
<Title>.... </Title>
<Paragraph>...</Paragraph>
</Introduction>
...
</Document>
```

Fig. 5. An XML document dealing with « *A practical approach to jet engine lubricant evaluation* »

After annotation, the XML document becomes:

```
<Document>
<title>A practical approach to <term conceptkey="TC1.1.1.1.1.1.5"
xml:lang="en-GB" status="1">jet engine</term>
    <term conceptkey="TC1.1.2.1" xml:lang="en-GB"
status="1">lubricant</term>
    <term conceptkey="TC2.1.3.1" xml:lang="en-GB"
status="1">evaluation</term>
  </title>>
</Document>
```

Fig. 6. An annotated XML document

The **term** markup has three attributes:

1. The first one called **conceptkey** is the type of the concept. Indeed, the expression, contained in the mark-up, references a concept. This concept is characterized by its type.
2. The second one, called **xml:lang** defined in the XML standard, indicates the language used.
3. The third one, called **status,** indicates the preference degree for the annotator. If the annotator think that the expression should the a label of the concept type **status** equals 1 otherwise **status** is less than 1.

The semantic thesaurus is automatically updated by parsing annotated documents that is to say to link an expression with a concept type. So document annotations enable to take into account the vocabulary evolution. The users interface of SyDoM displays the semantic thesaurus in the user's native language, to facilitate the annotating process.

4.2 Document index

After annotating a document, the librarian has a list of concepts referenced inside the document. These concepts can be displayed in any language belonging to the semantic thesaurus. The librarian selects the most important concepts and link them by relation. He could also add other concepts, he judges relevant for the document. Indeed, he builds a semantic graph where relations connect concepts. This indexing graph is transformed into a set of XML elements. In order to be language-independent, types, rather than labels, identify each component of the graph. Three kinds of mark-up are used:

- <Role type="TR1">...</Role> identifies a relation and its type.
- <Argument number="1"> </Argument> identifies arguments of a relation and its position in the list of arguments (number = 1).
- <Concept type= "TC2.1.3.1"/> identifies a concept and its type.

These markups are encapsulated in an index one added to the beginning of the document. Let us point out that none of these mark-ups has any word content so they do not alter the document presentation. These mark-up provide meta information about the document content.

```
<index>
    <role type="TR1.1.2.2.2.2">
        <argument number="1">
            <concept type="TC2.1.3.1"/>
        </argument>
        <argument number="2">
            <concept type="TC1.1.2.1"/>
        </argument>
    </role>
    <role type="TR1.1.2.1.2.1">
        <argument number="1">
            <concept type="TC2.2.1"/>
        </argument>
        <argument number="2">
            <concept type="TC1.1.2.1"/>
        </argument>
    </role>
    <role type="TR1">
        <argument number="1">
            <concept type="TC1.1.1.1.1.1.5"/>
        </argument>
        <argument number="2">
            <concept type="TC2.2.1"/>
        </argument>
    </role>
</Index>
```

Fig. 7. XML Index of a document dealing with « *A practical approach to jet engine lubricant evaluation* »

Retrieval module

4.3 Query formulation

The users introduce their queries through the query interface presented in figure 8. To formulate his query, the user goes through the semantic thesaurus presented in his native language to choose concepts and relations.

For example, a French user wants to query the mechanical collection of Doc'INSA. He is looking for documents dealing with combustion model. Thanks to the visualization of the French ontology, he builds the graph presented in the next figure:

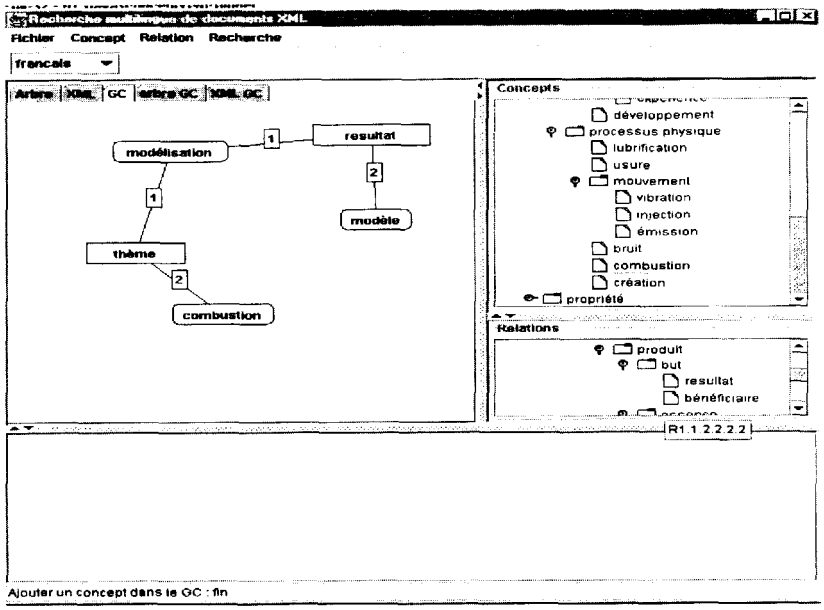

Fig. 8. SyDoM interface

As shown in Figure 9, this graph is transformed (French labels are replaced by types) to generate a language-independent query.

[TC2.2.5]→(TR1.1.2.2.2.2)→[TC2.1.3.3]←(TR1.1.2.2.1.1)←[TC1.2.1.2.1]

Fig. 9. A language-independent query graph

4.4 Visualization of result documents

Result documents could have different presentations depending on the user's choice. First of all, if the user wants a brief description of document centers of interest, he could see the index of the document written in his native language. As shown in Figure 10, this index is composed of concepts connected by relations where types are replaced by terms (labels of concept) thanks to semantic thesaurus.

Relation theme:
Argument 1: évaluation
Argument 2: lubrifiant
Relation instrument:
Argument 1: lubrification
Argument 2: lubrifiant
Relation composant:
Argument 1: moteur à réaction
Argument 2: lubrification

Fig. 10. The French index of the document dealing with "A practical approach to jet engine lubricant evaluation.

Next, if he believes that the document is useful, he could visualize the whole document. If the document is written in a foreign language, the document is translated partially. That is to say that every expression of concept, identified by annotation, is replaced by term in the user's native language.

Title: A practical approach to jet engine lubricant evaluation
...

Fig. 11. The English document

Title: A practical approach to (moteur à réaction) (lubrifiant) (évaluation)
...

Fig. 12. A partial translation of the English document

5. Evaluation

The library Doc'INSA gives us a test set of English articles. These articles deal with mechanics and they are called *pre-print of the Society of Automotive Engineers (SAE)*. The first step of the experiment was to build a semantic thesaurus for mechanics. Thanks to a mechanical thesaurus, we selected around one hundred mechanical concepts to supply our semantic thesaurus. On top of the concept hierarchy, we added the thirty-five relations proposed by Sowa in his Knowledge Representation Book [15]. During manual indexing, only titles are taken in account. For our first experiments, we have manually indexed approximately fifty articles and performs ten queries. The average index graph consists of four arches and the average query graph consists of two arches.

To evaluate our system, we compare it to the IR system used at Doc'INSA. In this system, documents and queries are represented by a list of keywords. The matching function between documents and queries evaluates the number of common keywords. The indices of this system were generated automatically from the index graphs of SyDoM, to avoid to take the index variability into account. The next figure presents the comparison of SyDoM with the Doc'INSA system. We compute the average precision for ten recall intervals. The constant value of similarity function are arbitrarily fixed[3] (V_G = *0.7 and* V_S= *0.9*). The trend of the curve can be explained by the fact that our collection size is small. Therefore, most part of the queries deals only with few documents. Because these documents are retrieved with an important weight (more than 0.8), the precision is good whatever the recall could be. We can notice that relation processing and hierarchy inference improve significantly the quality of the answer even for manual indexing.

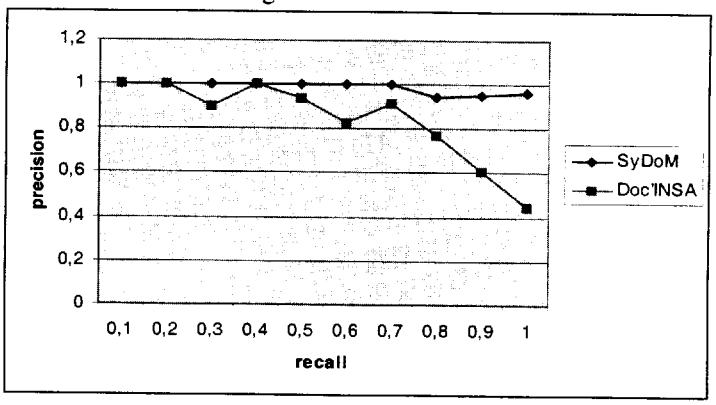

Fig. 13 Evaluation of SyDoM and Doc 'INSA system

[3] We chose the same value as [3]

6. Conclusion

In this paper, we propose a solution to the challenge using complex knowledge representation formalism for information retrieval purpose. Moreover, we present a graph formalism used to describe the semantic of document in a multilingual context. This formalism is an extension of the Sowa's formalism of Conceptual Graphs. Starting from recent works, we propose a new comparison operator between graphs taking the specific information retrieval needs into account. Our operator is based on the generic model suggested in [9] by considering the graphs in normal form. This choice enables us to decrease the complexity of the search algorithm. Our proposition is validated by the SyDoM prototype dedicated to digital libraries. SyDoM is a multilingual information retrieval system with several useful functionalities. It has been evaluated using an English collection of articles. We have already noticed that SyDoM gives better results than traditional documentary system.

7. References

[1] L. Ballesteros. W.B. Croft. « Statistical Methods For Cross-Langage Information Retrieval », *Cross-Language Information Retrieval*, Chapter 3. G. Grefenstette (ed.) Kluwer Academic Publishers, Boston, 1998.

[2] L. Bourrelly, E. Chouraqui. « Le système documentaire SATIN1: description générale et manuel d'utilisation ». *Centre National de Recherche Scientifique*, Paris, 1974, 398 pages.

[3] P.D. Bruza, M. Lalmas. « The use of logic in information retrieval modeling ». *Knowledge Engineering Review*, 13(2):1-33, 1998

[4] M.W. Davis. « On The Effective Use Of Large Parallel Corpora In Cross-Langage Text Retrieval », *Cross-Language Information Retrieval*, Chapter 2. G. Grefenstette (ed.)Kluwer Academic Publishers, Boston, 1998.

[5] S.T. Dumais, M.L. Littman, T.K. Landauer, T.A. Letsche. « Automatic Cross-Language Retrieval Using Latent Semantic Indexing », *Working Notes of AAAI Spring Symposium on Cross-Language Text and Speech Retrieval*, Stanford, CA, 1997. http://www.ee.umd.edu/medlb/filter/sss/papers/dumais.ps.

[6] D. Genest, M. Chein. « an experiment in Document Retrieval using Conceptual Graph ». Proceeding of 5th ICCS Conference, Washington, USA, p 489-504, august 1997.

[7] D. Genest. « Extension du modèle des graphes conceptuels pour la recherche d'information ». PhD Thesis, Montpellier University, Montpellier, France 2000.

[8] D.A. Hull, G. Grefenstette. « Querying Across Languages: A Dictionary-Based Approach to Multilingual Information ». *Proceedings of the 19th ACM SIGIR Conference in Research & Development in Information Retrieval*, Zurich, 1996, p 49-57.

[9] P. Martin, P. W. Eklund. « Knowledge Retrieval and the World Wide Web ». IEEE Intelligent Systems (Special Issue on Knowledge Management and the Internet) 2000.

[10] J.Y. Nie. « un modèle logique général pour les systemes de recherche d'informations. Application au prototype RIME ». PhD Thesis, Joseph Fouriei University, Grenoble, France 1990.

[11] I. Ounis, M. Pasça. « RELIEF: Combining expressiveness and rapidity into a single system ». Proceeding of 18th SIGIR Conference, Melbourne, Australia, p 266-274, august 1998.

[12] K. Radwan, F. Foussier, C. Fluhr. « Multilingual Access to Textual Databases ». Proceeding of the Conference on Intelligent Text and Image Handling RIAO91, Elsevier, avril 1991, p 475-489.

[13] C. Roussey, S. Calabretto, J. M. Pinon « Un modèle d'indexation pour une collection multilingue de documents». Proceeding of the 3rd CIDE Conference, Lyon, France, p 153-169, July 2000

[14] J. Sowa. « Conceptual Structures: information processing in mind and machine ». The System Programming Series, Addison Wesley publishing Company, 1984.

[15] J. Sowa. « Knowledge Representation: Logical, Philosophical, and Computational Foundations ». Brooks Cole Publishing Co., Pacific Grove, CA., 2000.

[16] C.J. van Rijsbergen. « A new Theoritical Framework for Information Retrieval ». Proceeding of the 9th SIGIR Conference, Pisa, p 194-200m septembre 1986.

[17] P Vossen, W. Peters, J. Gonzalo. "Towards a Universal Index of meaning". Proceedings of the ACL-99 Siglex Workshop, University of Maryland, 1999.