Semantic Approach based Multi-agent System for Information Search on the Web

NESSAH Djamel
University Center of Khenchela, Algeria
nhdjamel@yahoo.fr

KAZAR Okba
Biskra University, Algeria
kazarokba@yahoo.fr

ABSTRACT

The vector space model and various statistic /probabilistic approaches are widely used in models of information search to represent documents and user requests. The documents recovered are relatively relevant and generally troubled by noise and silence.
Our work is to propose a model whose objective is to improve the results towards a user query, this will be done by acting on measures of precision and recall, for this, first we use a multi agents system to reproduce the concepts of autonomy, cooperation and communication, which are inherent to this type of search systems, and secondly our approach will combine a syntactic search improved by the use of semantics that provides the WordNet taxonomy with a semantic search engine based domain ontology.
The knowledge base represented by the domain ontology is used to annotate documents by the concepts and the defined instances, therefore these form equivalent classes to classic indexing and constitute the semantic index on which the proposed search model is based.

Keywords: Semantic Web; Multi-Agent System; Semantic Search; Ontology; Semantic Index

1. Introduction

The semantic information search is a complex process, it has several stages, for example we have semantic annotation, query processing, and stage of evaluation and classification results. The complexity comes from the nature of information resources of semantic web that is not restricted to multimedia objects; also other objects may be people, places, and events exist.
Then, the semantic web doesn't work only with the known hyperlink relationship; there are several other types of relationships that link its different resources. We'll introduce in the following an architecture based agents, we deal mainly in our approach with the problem of how to develop and conduct a search engine based ontology and using paradigm agent techniques. Nevertheless, taking into account the lack and use of incompletes knowledge bases which limits the exclusive use of models based ontology; we chose to retain the use of keyword based search to complete this lack while making some improvements based on the semantics offered by WordNet taxonomy.
Section two describes some related works in semantic search domain, next section defines elements used for representing domain knowledge and performing semantic annotation process, in section four we'll present the different multi-agents system components and describe their interactions, the last section includes a conclusion and some prospects for future improvements.

2. State of the art

In order to improve quality of models developed for information search, many efforts have been deployed to annotate documents with semantic information. The related works to our approach concerns:

- Ontology based information search, which uses reasoning mechanism and ontological query languages to retrieve ontology instances (semantic layer) that annotate documents in the search domain.
- query expansion based information search, generally used with vector space models driven approaches.

Several studies exist, in [6] is described an architecture based agents and ontology which introduces the interest of restricting research on the web, the advantage of using software agents and ontology. The architecture is detailed in several levels of sub systems layers.
Another semantic search system based agents is proposed by [7], it uses the concept of conceptual graph closely related to natural language, and this type of knowledge's representation provides the ability to extract useful information by exploiting the logical relationships in the form of triplets (Relation, concept1, Concept2) between the terms in the documents collection.
In [8] we have a multi-agent system which performs an intelligent search; it is based on a

semantic approach and a process of enrichment of the ontological concepts by probabilistic notions.
The semantic approach is based semantic network associated to domain knowledge; in the graph, edges have weights that express the strength of semantic relationship which the edge carries between the nodes (concepts).
Another project developed at the university George Mason [9], is interested in searching information on heterogeneous databases (web), research is guided by ontology and is based multi agent system, this project uses a modular conceptual model expressed in OWL, and allowing the integration of other ontologies and other information resources.

3. Knowledge Representation

The system architecture includes the following components used for handling and representing the domain knowledge:
- WordNet Taxonomy
- Domain Ontology (Knowledge Base Annotation)
- Semantic Annotation

3.1 WordNet Taxonomy

WordNet is a lexical reference developed at Princeton University offering two separate services:
- A vocabulary describing the different meanings of words.
- A concept hierarchy describing the semantic relationships between words. Bringing together the terms of natural language (English), about 160,000 terms are organized in hierarchical taxonomies of names, verbs, adjectives and adverbs.
Each entry in WordNet is called "synset"; it is a set of synonyms with the same meaning. For example, the words "car","auto" and "automobile" are parts of the same synset.
Also, the same word can have several meanings, for the word "car" we find five possible senses. [1]
The synsets are connected at the top or bottom of the hierarchy by different types of relationships, most relationships are hypernym / hyponym i.e. "Is-a" relationship and holonym / meronym relationships "Part-of", in our approach we use only hypernym/ hyponym relations and representations of names that are commonly claimed to be the most representative form for the semantics of a language. these names are extracted from documents and queries, between the words, several methods for calculating semantic similarity were tested on the taxonomy WordNet, we have essentially two categories:

- Methods based ontology structure.
- Methods based information content of oncepts.

The important feature observed and which we tried to exploit in this approach is that, most methods of calculating semantic similarity try to assign a higher similarity to similar terms that are more specific, i.e. at the bottom of the hierarchy compared to similar terms but situated high in the hierarchy, i.e. the most general.

3.2 Domain Ontology

The fundamental objective of the semantic web is to extend current interfaces oriented to human understanding in a format automatically interpretable by programs, this requires developing a rich and a standard scheme of representation knowledge, it is named "domain ontology".
The concept of ontology has long existed, especially in philosophy, in computer science several definitions of ontology have been made, the most used is the one given by Gruber "*An ontology is an explicit specification of a conceptualization*".
The formalism of ontology as instrument of construction knowledge bases provides a controlled vocabulary to formulate queries, representing knowledge in (concepts, relationships and functions), classify the content of the documents, and make expansions of requests based on class hierarchy and rules on relationships. [2]
The ontology must be expressed in language enough expressive and carrying out reasoning mechanism, this understandable representation of the knowledge will allow software agents ability to find and handle domain entities.

3.3 Semantic Annotation Process

The semantic annotation allows agents who use a semantic search engine to decide intelligently about the relevance of the returned results, for these reasons the process of retrieving information depends largely on the quality of formal semantic annotations defined by domain ontology.
The domain's documents are annotated by concepts and instances of concepts, this annotation has two relational properties that are instances annotations and documents annotated and through which the concepts and documents are linked. So, terms (class, concept, datatype, object property, datatype property) defined in the ontology are used as metadata to annotate the content of the documents, these terms form the semantic index and are identified by URIs.
For our work we adopt OWL Lite as standard ontology specification because this language maintains a compromise between expressiveness to formulate domain knowledge and ensure reasoning decidability (e.g. Jena …).

3.3.1 Generating Equivalent Class

The concept of annotation and related techniques used such as the generation of equivalents annotation classes are not part of the objective of this work, we assume these classes have been generated by using an appropriate inference mechanism applied to the knowledge base that specifies the domain ontology; nevertheless, we give here an overview to clarify this concept.
Let Xi: a term of the semantic index, and [Xi] its equivalent class, so we can have Xi1∈ [Xi], Xi2 ∈ [Xi] …
Consider document "dj" A, B and C three strings in document dj semantically annotated by Xi1, Xi1,

Xi2. From the semantic point of view these three strings are equal even with their different syntax because A, B and C are semantically annotated with the same semantic index term Xi. (Figure.1)

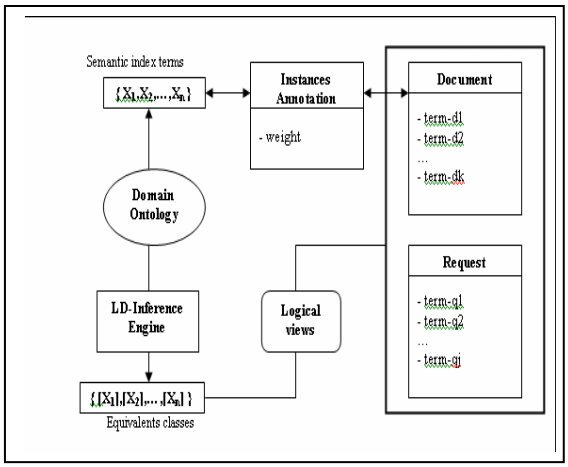

Figure.1: Generating equivalent annotation classes

3.3.2 Calculating Annotation Weight

Weights of the annotations are used to evaluate the relevance, the appropriateness and to implement a classification algorithm (Ranking) of documents obtained; these weights are generally calculated by the Tf-Idf algorithm.

The weights of annotations reflect the relevance of an instance for the semantic of the document where it appears, this model is based on the frequency of occurrence of annotation instances in each document and takes into account the principle of generating an equivalent annotation class described above, the adaptation of the algorithm Tf-Idf will consider the number of times an annotation's label of an equivalent class appear in a document, the formula is:

$$d_x = \frac{freq_{x,d}}{\max_y freq_{y,d}} * \log \frac{|D|}{n_x} \qquad (1)$$

d_x : weight of the instance "x" in document "d"

$freq_{x,d}$: number of occurrences in "d" of keywords linked with instance "x"

$\max_y freq_{y,d}$: The frequency of occurrence of the most repeated instance in the document "d"

n_x : The number of documents annotated by "x"

D : Total number of documents.

Based on works presented in [3], [4] and to simplify the calculations we'll only retain the importance of an instance in the document:

$$w_{xi} = \frac{freq_{i,d}}{\max_y freq_{yi,d}} \qquad (2)$$

Let D = (d1, d2 ... dn) the collection of documents in the search space.

([X1], [X2] ... [Xt]) the equivalents classes of the semantic index terms.

$freq_{xi,d}$: The total frequency of all elements of the equivalent class [Xi], appearing in all the semantic annotations of document "d".

$\max_y freq_{yi,d}$: Max (freq $_{x1,d}$,freq $_{x2,d}$…freq $_{xt,d}$)

Each document in space D will have a logical view relative to the weight of its annotation instances. So a document "dj" will be represented by the vector weight of annotation instances, and we write:

$$dj \cong (w_{1j}, w_{2j}, ...w_{ij}, ..., w_{tj}) \qquad (3)$$

Similarly, for a given request, we get its logical view to the whole equivalents classes ([X1], [X2] ... [Xt]) i.e.: $q \cong (w_{1q}, w_{2q}, ...w_{iq}, ..., w_{tq})$ (4)

4. System's Agents

The main advantage of using paradigm agent is to perfect the applications related to information research. The architecture that we propose as shown in Figure.2 is composed of three units:

- User Interface Unit
- Management Query Unit
- Research Information Unit

Agent "User-Interface" is considered to be the door through which the query is entered in the system, it receives and transmits to the system the user feedback and presents the search results.

Agent "Information-Research" collects in the area of interest relevant information resources; it may deal with several other subcontracts agents to accomplish this goal.

"Domain-Ontology" agent inspects and monitors the dynamic changes in information resources contents; it extracts and stores in an RDF base document's links that are annotated by concepts and instances of the specified ontology.

In the management query unit (processing) is situated the "Query-Treatment" agent which coordinates the activities of the system. It formulates and refines (prepare) the query to be submitted to the agent "Information-Research".

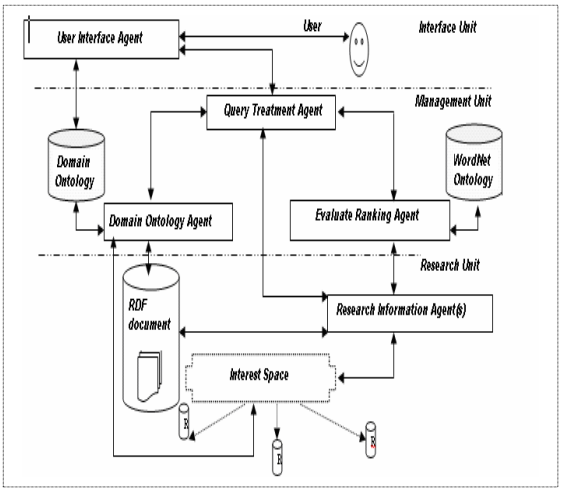

Figure.2: System's Architecture

Finally results returned will be analyzed and evaluated by the "Evaluate-Ranking" agent to determine their degree of relevance and decide to accept or reject results.

System's architecture including agents is given in Figure.2; the internal structure of each agents who compose the system will be detailed below.

4.1 User-Interface Agent

The interface agent resides on the desktop user; it provides the interface to interact with the system. For a search session it records the user request in terms of keywords.

Possibly the user can define its research domain and introduce various preferences such as his favorite search engine (default Google), and a set of variables defining thresholds calculations. Also this agent presents the user the search results when they arrive, it can implement an intelligent behavior and learn from past experiences and user feedback on earlier requests.

4.2 Query-Treatment Agent

In our multi-agents system, this agent manages the cooperative execution of the user request; it has knowledge about each agent which includes the identification of the agent and roles that it can perform ordered by its capabilities. (Figure.3) According to their various skills it allocates them tasks to achieve their common goal. Through interactions that the agent maintains with the "Evaluate-Ranking" agent it performs various substitutions involving:

- The weight of keywords: the weights of keywords are replaced by values calculated by a heuristic evaluation of similarity, these values are provided by the agent "Evaluate-Ranking". In other words, the keyword Mi with weight Wi is assigned a new weight Wsi calculated by the expression:

$$Wsi=Sim* Wi. \qquad 0<= Sim <=1 \qquad (5)$$

- The keywords by Hyponym / hypernym: This task aims to assist the user to reformulate its request by offering him choices, i.e. the synsets excerpts from the WordNet hierarchy.

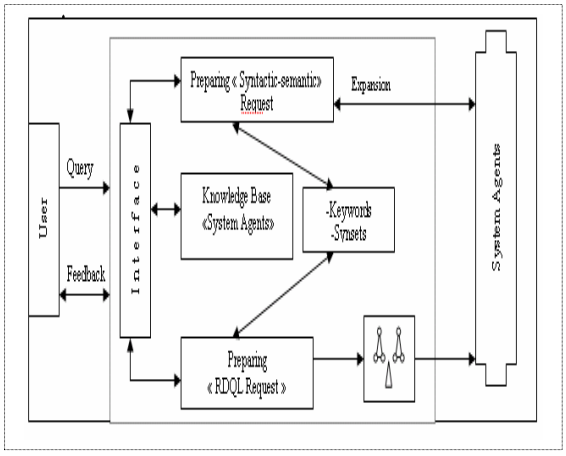

Figure.3: Query-Treatment Agent Structure

Another module that complements the first one prepares the same query based concepts; it is a semantic search which uses relationships between concepts as follow:

An RDQL query is generated by the module from the keywords expressed in the original request. Also it may be done by the "User Interface" agent who in this case reaches the domain ontology and help the user to explicitly select classes and introduce the desired values of properties.

The "Query-Treatment" agent interacts with the agent "Domain-Ontology" to run on the pattern of domain ontology and instances of concepts specified in OWL the RDQL query, the result is a set of instances that strictly satisfy conditions of the RDQL query. (Standard engine such as "Jena" is used to execute RDQL queries). This means instantiation operations of concepts of the ontology's scheme OWL by values of variables used in the query and the invocation of reason Jena to infer the related knowledge.

4.3 Information-Research Agent

The first research component of this agent is based syntactic keywords and targets the area of research through a traditional search engine, however, to improve research results purely syntactic a second component module of this agent performs semantics research.

Both modules operate in parallel each one receives input model adapted to query search mode prepared by the agent "Query-Treatment" (keywords to perform syntactic-semantic search and generated instances of concepts derived from the execution of RDQL query to perform a semantic search).

The agent can contract several other agents to complete the research, choosing an agent of such research can depend on the agent capability and the nature of the information sought. (Figure.4)

4.3.1 Syntactic-Semantic Search Module

Uses a search engine (e.g. Google, Yahoo …) to find syntactically in the area of interest documents that satisfy the submitted query.

It is a search of purely syntactic correspondence between the keywords in the query and terms indexing the documents available in space research.

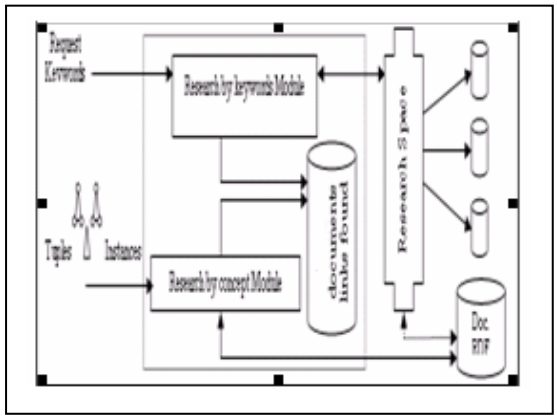

Figure.4: Information-Research Agent Structure

4.3.2 Semantic Search Module

This module research in the RDF documents base, the RDF annotations that match tuples instances recovered by the "Query-Treatment" agent.

The module receives input tuples which are the results of the RDQL query, then, documents whose links have been stored in the RDF database are analyzed and those annotated by these tuples instances are found, they are considered semantically relevant. The agent records in a temporary file the following details: Links of resources found and their evaluated similarities.

4.4 Evaluate-Ranking Agent

The "Information-Research" agent stores links of resources found in a temporary file to which the "Evaluate-Ranking" agent accesses, so it is a type of memory that can be modeled by a blackboard. For each entry, the "Evaluate-Ranking" agent download page referenced by the link. Keywords that syntactically index the page or semantic index instances are assigned weights according to the principle of vector model.

4.4.1 Evaluate Module

Let Wij: weight of term "i" (keywords) in page j.

$$W_{ij} = \frac{Freq\_t_i}{\max(freq\_t_j)_{j=1,n}} \qquad (6)$$

n: the number of keywords

User's query is also represented by the weight vector Q = (w1q, w2q, wiq ,..., wnq), where wiq is the weight of the keyword “i” in query Q, a keyword may be a keyword's synonym, its hyponym or hypernym. (Figure.5)

The semantic annotation is based ontology, the ontology defines the concept's terms used as metadata to form the semantic index, thus, these terms are identified by URIs and may be equivalent classes. By analogy with the space vector model, semantic annotations are assigned weights reflecting the importance of the annotation instance for the document, therefore in RDQL queries; the variables in the SELECT clauses are assigned weights according to the principle of vector model.

The formula of cosine is used to calculate the similarity document-query, so for a page "j" and a request "q" we used the expression:

$$Sim(Pj,q) = \frac{\vec{Pj}.\vec{q}}{|Pj|.|q|} \qquad (7)$$

The similarity evaluated is compared to a minimum threshold indicated by variable Rmin initially fixed by the user.

$\vec{P}_j, \vec{q}$: weight vectors associated to Pj, q.

$|p_j|,|q|$: respectively P_j and q vectors norms

When $Sim(Pi,q) >$ =Rmin, the page Pj is considered relevant, its link and the similarity's value are returned to agent "Information-Search" for final storage.

In the case $Sim(Pj,q)$ $<R_{min}$ the current page is ignored, the process ends when all the pages are crawled, at the end we will have obtained a set of relevant pages.

Following user feedback, if the number of relevant resources found is sufficient, an algorithm for grading results is executed to present the results according to their degree of relevance; this algorithm is implemented by the ranking module in the structure of this agent.

If we want against include more resources (depends on user feedbacks), the "Evaluate-Ranking" agent will explore the relationships between concepts defined in the ontology WordNet to extract sets of synonyms, hyponyms and hypernyms.

The expansion of the query will use the "synsets" in the limits of depths set by the user, but generally when using an expansion with hypernym synsets the depth is set to "1" because the similarity tends to decrease when generalizing sense.

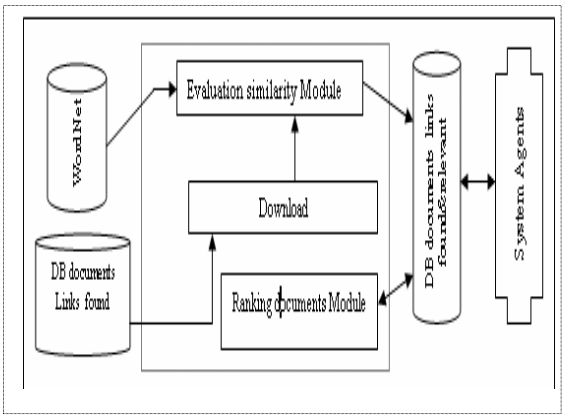

Figure.5: Evaluate-Ranking Agent Structure

4.4.2 Calculating Similarity Module

Based on user feedback to choice the terms to be used to extend the query, we use the formula (Figure.6) to assess similarities and update the weight of query keywords. The forms used are improved from those presented in [5], our choice to use these forms is justified by opportunity for considering the structure of the ontology through two parameters:

- The length of the path linking concepts C_a and C_b
- The depth of concepts C_a, C_b in the hierarchy

$$\textbf{Sim(Ca,Cb)=}\begin{cases} \dfrac{1}{2^{\log\frac{D*n}{Min(D_a,D_b)+1}}} & C_b,C_a \text{ Hypernyms} \\ \dfrac{1}{2^{\log\frac{D*n}{Max(D_a,D_b)}}} & C_b,C_a \text{ Hyponyms} \\ \mathbf{1} & C_b,C_a \text{ Synonyms} \\ \mathbf{0} & C_b,C_a \text{ Not Linked} \end{cases} \qquad (8)$$

Figure. 6: Forms of Calculating Similarity

D: depth hierarchy; n: minimum path length between concepts Ca and Cb (number of arcs). D_a,D_b concepts depths.
The expansion for synonyms synsets is implied, the terms of a synonymous synsets are assumed to have a similarity of "1" i.e. they are identical.
For synsets hyponyms / hypernym and following the user choice, a similarity value is sent with the term to "Query-Treatment" agent to recompose query and restart a new syntactic search. (Figure.5)
The process is repeated until all sets constructed would be entirely explored.

4.4.3 Ranking Module

It is common to present the user the obtained results by their relevance order. The agent accesses the temporary database of relevant documents, for each one it estimate its similarity with submitted query. The final similarity is calculated by an expression of type:

$$Sim_F = a*Sim_{syn} + b*sim_{sem} \; ; \; a \in [0,1] \; ; \; b = 1-a \quad (9)$$

Documents returned and witch have a high similarity are those with a # 0 and b # 0.

4.5 Domain-Ontology Agent

Attached to the domain ontology, this agent maintains the ontology on the fly. Also the agent uses a standard engine (e.g. Jena and racer) to infer knowledge and execute the RDQL query, results (instances) would be communicated to the "Query-Treatment" agent.
The other task of the agent is to browse the web (area of interest) at regular intervals to detect documents annotated RDF consistent with the specified domain ontology, the document's links found will be stored in the database documents annotated RDF. Agent can therefore take into account the dynamics of information on the web in independently and proactively manner.

5. Case Study

We chose to implement our model the tourism domain, tourist sites on the web are annotated by their owners by RDF triples and instances defined in the domain ontology used.
Tourism has several domains (transport, entertainment, sports, scientific conferences, etc.), but to simplify the analysis, we will limit study to hotel domain for which we associate a domain ontology named “hotel”.

5.1 UML specification of the ontology

The ontology ("hotel") is specified in the language OWL-Lite; this ontology is associated with a UML diagram specifying the classes (concepts), the properties and relationships between concepts and examples of instances of concepts. (Figure 7)

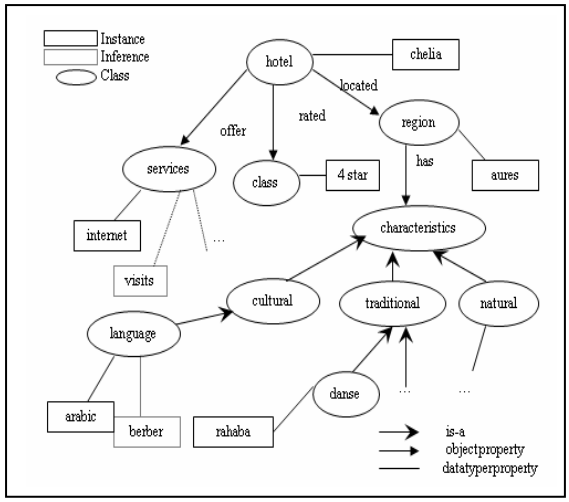

Figure 7: UML diagram of ontology “hotel”

An inference engine applied to the ontology schema and instances defined, will infer knowledge other than those explicitly declared, inference is a mechanism that is based on the expressiveness of the language (OWL-Lite) and its formal semantics based on description logics, especially this concerns restrictions on the classes, the properties among classes and the axioms on the classes.
For example, we specify that a 5 star rated hotel must have as service “visits” by the class: visits=((hotel) ∩ (> = 4 rated.star)).

5.2 Inference models

The integration of the Jena API allows the system to derive additional RDF assertions included in the OWL knowledge base; this mechanism supports the languages RDF / S and OWL and uses an inference model which has two components:
• The schema of the model
• The instances of the model
The example below is an illustration of an inference model used by inference engine RDFS.
Inference is performed by the transitive relation on properties which defines 'room service 'as a sub property of the property "hotel service".

5.2.1 Schema of model

```
<?xml version="1.0"?>
<!DOCTYPE rdf:RDF [ <!ENTITY hotelerie
'http://mydomain/ontology/infohotel/'>
<!ENTITY rdf 'http://www.w3.org/1999/02/22-rdf-
syntaxns#'>
<!ENTITY rdfs
'http://www.w3.org/2000/01/rdfschema#'>
<!ENTITY xsd
'http://www.w3.org/2001/XMLSchema#'>]>
<rdf:RDF xmlns:rdf="&rdf;" xmlns:rdfs="&rdfs;"
xmlns:xsd="&xsd;"
xml:base="http://mydomain/ontology/infohotel/"
xmlns="&hotelerie;">
<rdf:Description rdf:about="&hotelerie;room-service">
<rdfs:subPropertyOf
rdf:resource="&hotelerie;hotelservice"/>
</rdf:Description>
<rdf:Description rdf:about="&hotelerie;hotel-service">
```

<rdfs:range rdf:resource="&hotelerie;Hotel"/>
<rdfs:domain rdf:resource="&hotelerie;Serviceh"/>
</rdf:Description>
<rdf:Description rdf:about="&hotelerie;classement">
<rdfs:range rdf:resource="&xsd;integer" />
</rdf:Description>
</rdf:RDF>

5.2.2 Instances of model

```
<?xml version="1.0"?>
<!DOCTYPE rdf:RDF [ <!ENTITY hotelerie
'http://mydomain/ontology/infohotel/'>
<!ENTITY rdf 'http://www.w3.org/1999/02/22-rdf-
syntaxns#'>
<!ENTITY rdfs
'http://www.w3.org/2000/01/rdfschema#'>
<!ENTITY xsd
'http://www.w3.org/2001/XMLSchema#'>
]>
<rdf:RDF xmlns:rdf="&rdf;" xmlns:rdfs="&rdfs;"
xmlns:xsd="&xsd;"
xml:base="http://mydomain/ontology/infohotel/"
xmlns="&hotelerie;">
<Hotel rdf:about="&hotelerie;Chelia">
<room-service rdf:resource="&hotelerie;internet" />
<class>3</class>
</Hotel>
</rdf:RDF>
```

The execution of code associated with this model produced the following results:

Type: Chelia is
http://mydomain/ontology/infohotel/chelia rdf:type
*http://mydomain/ontology/infohotel/**hotel***
Type: Chelia is
http://mydomain/ontology/infohotel/chelia rdf:type
*http://mydomain/ontology/infohotel/**serviceh***

6. Conclusion

The proposed semantic research model based multi-agent system and using domain ontology illustrate the concept of cooperative resolution of distributed problems, the process combines a search engine based ontology with a traditional search-based keyword which include relations of synonymy and hyponymy provided by the WordNet taxonomy.

The semantic search uses as support an RDQL query generated from query keywords, then an inference engine such as "Jena" will use the ontology scheme to retrieve defined instances in correspondence with keywords in the query, these instances will be sought in the RDF database and return the documents that they annotate.

As prospects for research in this area and in relation with our model, we propose to enrich the knowledge base agents with techniques for formulation query including explicit rules and policy decision.

This will allow the "Query-Treatment" agent to optimize the request in an intelligent way, it is true that over the query is well-defined, better relevant results are obtained.

Also, to take advantage of new technologies applied to artificial intelligence systems, we intend to couple the agent "Query-Treatment" with a system of reasoning from cases (CBR), this will enable and perfect the search process by reasoning from cases already resolved and stored in the CBR data base.

References:

[1] G. Varelas, E. Voutsakis, P. Raftopoulou "Semantic similarity methods in WordNet and their application to information retrieval on the Web", *WIDM'05*, November 5, 2005 Bremen, Germany.
Copyright 2005 ACM 1-59593-194-5/05/0011.

[2] T. Osman, D. Thakker, G. Schaefer, P. Lakin, "An integrative semantic framework for image annotation and retrieval"
International Conference on Web Intelligence
0-7695-3026-5/07, 2007 IEEE/WIC/ACM

[3] S. Jun-feng ; Z. Wee Ming ; X. Wei-dong, Li Guo-hui ; Xu Zhen-ning
"Ontology-based information retrieval model for the semantic Web"
School of Information System and Management, the National University of Defense Technology, Changsha 410073,China

[4] P. Castells, M. Fernandez, D. Vallet, "An adaptation of the vector-space model for ontology-based information retrieval"
IEEE transactions on knowledge and data Engineering, Vol. 19, No. 2, February 2007
1041-4347/07

[5] E. Toch, A. Gal, "A semantic approach to approximate service retrieval"
ACM Transactions on Internet Technology, Vol. 8, No. 1,
Article 2, Publication date: November 2007.

[6] B.Espinasse ; S.Fournier ; F.Freitas
"AGATHE : Une architecture générique à base d'agents et d'ontologies pour la collecte d'information sur domaines restreints du web"
CORIA 2007

[7] Tanveer J Siddiqui ; U.S tiwary
"Integration notion of agency and semantics in information retrieval :An intelligent multi agent model"
University of Allahabad IEEE 2005

[8] Carmine Cesarano ; Antonio d'Acierno ; Antonio Picariello
"An Intelligent search agent system of semantic information retrieval on the Internet"
ACM 2003

[9] Larry Kershberg ; Mizan Chowdhury ; Alberto Damiano ; Hanjo Geong ; Scott Mitchell ; Jingwei Si ; Stephen Smith
"Knowledge Sifter: Agent based ontology-driven search over heterogeneous databases using semantic web services"
George Mason University USA 2004