Towards an on-line Semantic Information Retrieval System based on Fuzzy Ontologies

Hajer Baazaoui, Marie-Aude Aufaure, Rania Soussi
Riadi-GDL Laboratoy, ENSI Campus Universitaire de la Manouba, Tunis, Tunisia, INRIA Paris-Rocquencourt - Domaine de Voluceau - 78 153
Le Chesnay Cedex France Hajerbaazaouizghal@riadi.rnu.tn, Marie-Aude.Aufaure@inria.fr, Rania.Soussi@riadi.rnu.tn

ABSTRACT: The huge number of available documents on the Web makes finding relevant ones a challenge. Thus, searching for information becomes more and more complex because of the growing volume of data and of its lack of structure. The quality of results that traditional full-text search engines provide is still not optimal for many types of user queries. Especially, the ambiguities of natural languages and abstract concepts are handled inadequately by full-text search engines. Ontologies provide a solution to these problems. They can help a user to find documents related to a specific domain. This paper proposes a new retrieval system based on ontologies. This system integrates results from traditional full-text engines, and thus supports a gradual transition from classical full-text search engines to ontology-based ones.

Keywords: information retrieval, semantic web, ontologies, fuzzy ontologies

1. Introduction
With the continual increase of the Web size, a lot of good information is available online. However, this information is not always the best one in the context of the user.

Full-text search is still the most popular form of search and is very useful to retrieve documents for which we know the keywords to search for. Indeed, full-text search is not suitable for finding relevant documents about a specific topic in the context of a given task.

Another problem relies in the huge number of retrieved documents. It is difficult for a user to deal with thousands of electronic documents. The techniques often used by search engines are based on statistical methods and do not permit to take account of the semantics contained in the user’s query as well as in the documents. Adding a semantic dimension to web pages is a possible response to the problems found in the current web and is known as the semantic web, which provides technologies to make data integration possible.

Semantic web technologies include methods for ontology learning and metadata generation. The need for domain and service ontology directly used for information retrieval (IR) has not been explored much yet. Some approaches have been developed to extract semantics and so, to better answer users’ queries. On the other hand, most of these techniques have been designed to be applied to the whole Web and not on a particular domain. It could be interesting to use an ontology to represent a specific domain.

In this paper, we propose and discuss the details of an online IR system that uses ontologies to reformulate queries and to retrieve relevant information using the vector model and the cosine measure to compute the similarity. In section 2, the related work will be presented. Section 3 describes the proposed system for on-line information retrieval using Ontologies. Section 4 provides description of the implementation and experimental results. Finally, we give some perspectives.

2. Related Works
Information retrieval (IR) deals with the representation, storage, organization and access to information. The representation and organization of the information items should provide an easy access to the information that the user is interested in. However, traditional Information Retrieval systems often fail to present relevant documents to the users. Moreover, some works in IR try to improve the retrieval process (text indexing, query formulation) with the help of ontologies. Ontology is a foundation for the semantic web (Fensel et al., 2005) which can ease the IR process (Masolo, 2004) from various heterogeneous data sources. Indeed, it can improve recall and precision by eliminating the ambiguity of natural language using axioms and concepts. Each word has only one definition. Experiments were carried out using Wordnet with a smart strategy showing that a significant gain is possible on the TREC dataset (Bazizz, 2004), (Voorhees et al. 1994). It can then help the user to detect his needs and find the appropriate keywords using existing concepts of the ontology and their description.

The knowledge embedded in ontologies can be used at various levels in the IR process. It can contribute to the documents indexing, called semantic indexing. Ontologies can also help the user to formulate their needs and to access the documents. Finally, it is possible that ontology involves matching between the query and the documents.

In this paper we will mainly use domain and service ontologies. The first one allows the definition of a domain with concepts, relations and axioms. It is a quintuple composed of:

- two sets of concepts and relations noted respectively by C and R;
- a “Hierarchy of Concepts” as: HC C x C, HC (C1, C2). It means that C1 is a hyponym of C2;
- a Hierarchy of Relations that defines some non taxonomic semantic relations as HR R x R, HR (R1, R2), where R1 is a subclass of R2. Three functions are associated to this hierarchy:
  - The function “relation “: R C x C where relation(R) = (C1, C2);
  - The function “domain “: R C with domain(R) = C1;
  - The function “arranges “: R C with range(R) = C2.
- a set of axioms expressing restrictions on relations.

Keywords: information retrieval, semantic web, ontologies, fuzzy ontologies

1. Introduction
With the continual increase of the Web size, a lot of good information is available online. However, this information is not always the best one in the context of the user.

Full-text search is still the most popular form of search and is very useful to retrieve documents for which we know the keywords to search for. Indeed, full-text search is not suitable for finding relevant documents about a specific topic in the context of a given task.

Another problem relies in the huge number of retrieved documents. It is difficult for a user to deal with thousands of electronic documents. The techniques often used by search engines are based on statistical methods and do not permit to take account of the semantics contained in the user’s query as well as in the documents. Adding a semantic dimension to web pages is a possible response to the problems found in the current web and is known as the semantic web, which provides technologies to make data integration possible. Semantic web technologies include methods for ontology learning and metadata generation. The need for domain and service ontology directly used for information retrieval (IR) has not been explored much yet. Some approaches have been developed to extract semantics and so, to better answer users’ queries. On the other hand, most of these techniques have been designed to be applied to the whole Web and not on a particular domain. It could be interesting to use an ontology to represent a specific domain.

In this paper, we propose and discuss the details of an online IR system that uses ontologies to reformulate queries and to retrieve relevant information using the vector model and the
The second one is the service ontology which presents another view of the domain while putting in value these different actors, services, subservices, activities, tasks and processes of each service progress.

This ontology of services allows to specify:

- services related to a specific domain,
- the main activities done or requested respectively by services provider or customer,
- the activity tasks related to a service,
- the relations between services, subservices and activities,
- the management rules related to a service or specific activity.

The relation between the domain ontology and the service ontology of the domain is done by the association of each of the tasks to the triplet (Concept, relation, Concept). The goal is to determine the set of concepts and relations that identifies each service.

In order to take benefit from this gain in information retrieval and to satisfy the user’s needs, we have implemented an on-line retrieval information system based on ontologies and enriched it by fuzzy ontologies. This system is described in the next sections.

3. On-Line Semantic Information Retrieval

The proposed on-line IR system uses two ontologies (domain ontology and service ontology) and WordNet. The main idea is that, for a particular domain, it could be useful to associate a set of available services to specific domain ontology (tourism, medicine, cultural heritage, etc.). The service ontology is related to tasks, such as that we find in the tourism domain, hotel booking, car reservation, etc. In the future, this system will be integrated to a semi-automatically domain and service ontologies building prototype (Ben Mustapha et al., 2007). In next sub-sections, the system is detailed.

3.1. The building of Ontologies

The domain ontology which is manually built, is the basis of our on-line retrieval system, and contains the entire domain concepts and their properties. In addition, we have also used the service ontology.

Our desire to define an ontology related to domain services comes from studying the task ontology. In a web context, the most important problem is to find easily relevant knowledge satisfying a solicitation of a target service.

This ontology related to domain services has to be defined in order to specify:

A set of the domain services;
- The main activities involved in the user’s queries or in the provider’s services;
- The tasks related to these activities;
- The relations existing between services, subservices, activities and tasks;
- The management of rules related to the achievement of services;

This ontology decomposes every service in activities and every activity in tasks. Indeed, an activity is a set of elementary tasks or works executed by a person or a group and that leads to realize possessions or services. The conceptual model of the service ontology is illustrated by figure 1. The relation between the domain ontology and the service ontology is established by the association of each task to the triplet (Concept, relation, Concept). The goal is to determine the set of concepts and relations that identifies each service.

Let the functions $\text{ProjectionService}$, $\text{ProjectionActivity}$ and $\text{ProjectionTask}$ defined by:

- $\text{ProjectionService} : \text{Service} \rightarrow \text{Concept of Domain}$,
- $\text{ProjectionActivity} : \text{Activity} \rightarrow \text{Concept of Domain}$,
- $\text{ProjectionTask} : \text{Task} \rightarrow \text{Concept of Domain}$.

We illustrate these functions by the following examples:

- Example 1: $\text{ProjectionService} (\text{"Lodging_hotel"}) = \{\text{"hotel"}, \text{"region"}, \text{"name"}, \text{"room"}, \text{"room_type"}\}$.
- Example 2: $\text{ProjectionActivity} (\text{"disponibility_verification"}) = \{\text{"hotel"}, \text{"room"}, \text{"room_type"}\}$
- Example 3: $\text{ProjectionTask} (\text{"verification_type_room"}) = \{\text{"room"}, \text{"room_type"}\}$

With these functions, an information retrieval system can deduce the concepts which define a given service.

Indeed, the concepts (“hotel”, “room”, “room_type”) describe the service (“lodging_hotel”) because $\text{ProjectionService-1}(\text{"hotel"}, \text{"room"}, \text{"room_type"}) = \text{"verification_type_room"}$ whereas $\text{ProjectionService-1}(\text{"region"}) = \{\text{"Travel"}, \text{"Guided_visit"}, \text{"Lodging_hotel"}\}$. This function can be useful for a semantic document classification by services for a fixed domain.

This relation between the domain ontology and service ontology can be shown in the example in figure 2. For example, the concept “hotel” has the properties “name”, “star number” and address. Each concept can have sub concepts, i.e. “room” and “suite” for the concept hotel. Each concept of the domain ontology is the subject of one or many services, activities or tasks. For example, the “lodging_hotel” service is associated to the “hotel” concept.

This relation gives the capability to improve retrieval and help the users to better express their needs.
3.2 SIRO Architecture

The proposed system SIRO, presented in figure 3, is composed of three main modules: query processing and enrichment, search and document processing and finally a module for service classification.

The query processing module reformulates the user query using the concept and relation of the domain ontology and WorldNet. The expanded query is then submitted to a web search engine. In a further step, documents will be classified by services using the service's classification module. This module guides the user to build a second query using posted services. These modules cooperate together and will be detailed in the next subsections.

The main originalities of this system are:

- The use of a domain ontology combined with a service ontology,
- The classification by service of the results of a query which can be used later to perform a search based on the corresponding services; we hope that the result classified by category will make the use of information easier,
- The adaptation of the well-known vector model in which we substitute terms by concepts.
3.2.1. Query enrichment.
To obtain a more relevant and domain specific query, the user’s query is enriched using concepts and relations of the domain ontology and WordNet (to find the synonym, hyperonym, hyponym of query’s keywords). With SIRO, the user can also use the domain ontology to choose the concepts of his query.

The user’s query is enriched by the morphological and the semantic analysis:

- Morphological Analysis
It allows the recognition of the various forms of words using a lexicon (dictionary, thesaurus). The lemmatisation allows the transformation of a term to its canonical form or lemma. For example, from the derivation rules, the basic form can be found. Example: constitutional -> constitution.

In SIRO, the lemmatisation of the user’s query is ensured by TreeTagger (Stenback et al., 2003).

- Semantic analysis
After the lemmatisation, a filtering is performed. It consists in eliminating empty words (we only keep verbs, different from get, do, etc., and names).

WordNet, organized in synsets (sets of synonyms), returns all the different meanings associated to words in English language.

For example, « room » has 5 meanings with WordNet. The noun room is associated to the four following meanings:

1. (527) room -- (an area within a building enclosed by walls and floor and ceiling; “the rooms were very small but they had a nice view”)
2. (36) room, way, elbow room -- (space for movement; “room to pass”; “make way for”; “hardly enough elbow room to turn around”)
3. (11) room -- (opportunity for; “room for improvement”)
4. room -- (the people who are present in a room; “the whole room was cheering”)

The verb room has one sense:

1. (1) board, room -- (live and take one’s meals at or in; “she rooms in an old boarding house”).

In our system, WordNet is used to resolve ambiguities; the users have the opportunity to select the appropriate meanings of keywords contained in their query.

Then, the following algorithm is applied to the lemmatised words:

- If the word is a verb, we add its lemmatised form to the query: the lemma of “travelled” is “travel”; so, “travel” will be used in the new query.
- If the term is a name: its synonyms and hyponyms are extracted from WordNet, using the meaning chosen by the user. In this case, (1) if the word or one of its components exists in the ontology, the word will be replaced by the corresponding concept and its properties, or (2) neither the word nor his components were present in the query, thus the word is eliminated.

Finally, we obtain a refined query built with the concepts and relations extracted from the domain ontology.

Example:
The initial query sent by the user is: “find hotel”. This query is then processed using Tree Tagger. The lemmatized form is returned: find VV find, hotel NN hotel (term, grammatical type, and lemma). The word “hotel” has a unique sense in WordNet which is: « a building where travellers can pay for lodging and meals and other services », there “find” is an empty verb and is removed from the query. The first query is “hotel”. The semantic analysis returns an enriched query, which is “Hotel Suite Room Address Name Price Star”.

3.2.2. Query processing.
We use the ontology to generalize words with concepts as explained in the previous steps. After its enrichment, the query is submitted to a web search engine (google). Google gives back a ranked list of results with the score of the algorithm Page Rank which measure the popularity of a web page.

Then, we obtain a first list of documents, which is dynamically refreshed by Google.

The resulting documents are processed as follows:

Semantic analysis: Every returned document, except pdf or word ones are downloaded from their URL into an HTML document. It is parsed using DOM (Stenback et al., 2003).

The next step consists in extracting text and performing a morphological analysis with TreeTagger and getting the word’s lemmatized form. We can then detect existing concepts appearing both in the domain ontology and in the user’s query.

Filtering with the Vectoriel model: In order to rank documents, domain and service ontologies, and to find the most similar documents, we use the vector model proposed by Salton (Salton, 1993).

We have adapted this model by substituting the terms by concepts. More precisely, in the similarity measure, we have replaced the term’s weight by the concept’s weight. A concept is represented by its synonyms, hyperonyms and hyponyms.

Each document is represented by a vector Dj = (d1j, d2j, d3j..., dNj), and each query by a vector Q = (q1, q2, q3..., qN), where dj or qi is the weight of the word in the document or in the query.

In SIRO, the weights of the terms in the query are always equal to 1 (the query is reformulated with the concepts and relations of the ontology, and without any repetition of the same term). More explicitly, if our reformulated query is for example “hotel room” so Q = (1, 1).

Concerning the documents, we use the TF-IDF standardized formula, in order to give an equal chance to all documents, without giving a greater importance to long documents:

\[ w_{ij} = \frac{w_i}{\sum_{i=1}^{N} w_i} \]

with \( w_{ij} = (1 + \log tf_{ij}) \cdot idf_{et} \)

\[ \text{idf} = \log \frac{N}{n_i} \]

The similarity between the user’s query and a document is computed with the cosine formula:

\[ \text{SIM} (D_j, Q) = \frac{\sum_{i=1}^{n} d_{ij} \cdot q_i}{\sqrt{\sum_{i=1}^{n} d_{ij}^2} \cdot \sqrt{\sum_{i=1}^{n} q_i^2}} \]

or more precisely,

\[ \text{SIM} (D_j, Q) = \frac{\sum_{i=1}^{n} d_{ij} \cdot q_i}{\sqrt{\sum_{i=1}^{n} d_{ij}^2} \cdot \sqrt{\sum_{i=1}^{n} q_i^2}} \]

With n is the number of concepts in the query.

We compute the similarity of all the documents. Therefore, we obtain a second ranked list by ascending order according to the documents’ similarities with the query. Only those having a strictly positive similarity are proposed to the user.
3.2.3. Document classification.
Classification is a process that affects a document in one class or subclass. It is expected that a result classified by category makes the use of information easier. Our classification module classifies the results of a user query by service as explained below.

The reformulated query contains a set of concepts which belongs to the domain ontology. These concepts are linked to a set of services belonging to the service ontology. More precisely each concept is related to one or more services. For example, the “restaurant” concept is associated with the service “Restaurant trade”. This relation is manually built during the construction of the service ontology.

Given the concepts added to the query and the links between the domain ontology and the service ontology, we can extract all the services related to these concepts.

The vector model is used again to represent a service by a vector: Servi= (c1, c2,….Cn) with N is the number of concepts related to a service. For each selected document, we compute its similarity with the services by using the cosine formula. The document is then affected to the most similar service. Finally, the documents are displayed by service.

In our previous example, the initial query was “find hotel”. The services, activities and tasks detected from the concepts in the service ontology are:

Service: Lodging_Hotel
Activities: Reservation, Hotel_Search, Modify_Reservation, Disponibility_Verification,
Tasks: Verify_RoomNumber, Verify_RoomType, Search_HotelName, Search_HotelAdress.
Each document having a similarity greater than 0, is attributed to service, activity and task.

3.2.4. Query reformulation.
Classification gives the user the opportunity to determine the categories of tasks, activities and services which are related to its query. The users can then choose the services corresponding to their needs.

The query processing module captures the selected services and uses the relation between the service ontology and the domain ontology to formulate a new query including the new concepts and relations detected. For example, by choosing the service “Restaurant trade”, the concept “restaurant” will appear in the new query. The latter is then sent to the search engine and the refinement process is iterated.

In our example, if the user chooses the “Search_HotelName” task, the new query will be “Hotel Name” because the concept “Hotel” and the property “Has_Name” are linked to this task.

3.3 Limits of SIRO
Using ontologies in SIRO allows us to improve the precision and the recall of the returned results while integrating semantics to classic research methods. But, in spite of this efficiency, in certain cases some problems of linguistic ambiguity have persisted. These problems are related to the fact that:

- The same word is not understood in the same way by different people (for certain person, the concept “Safari” is an activity of adventure while for others it is a relaxation activity),
- The relations between concepts don’t have the same value from one person to another (For example different people think that a hotel must solely have a restaurant which is translated in ontology by the relation “has-restaurant” between the concept “hotel” and the concept “restaurant” while for another person this relation is not necessary).
- The same concept can be localized in multiple sites for the same domain ontology. For example “Safari” could be found in “Adventure” and “Sightseeing”.

These samples of problems could not be solved by a simple ontology. However, the theory of fuzzy sets can solve uncertain or fuzzy information problems. This theory has been used in the context of information retrieval (Bordogna et al., 2000) (Pasi, 2002) and gave interesting results. Therefore, a solution to solve uncertain information problems and to improve the IR process is to combine ontologies with fuzzy logic in order to build fuzzy ontologies.

3.4 Fuzzy ontologies construction
Fuzzy logic was conceived to adjust the problem of uncertain fact description and to allow the characterization of these elements of graduation. It has been introduced by L.A. Zadeh by the end of 60’s as an extension of the Boolean logic. Using the fuzzy logic, facts can be represented even if they are very small, fairly small, and normally small, etc.

Fuzzy logic provides a theoretical framework for the representation and treatment of these data with their imperfections. It doesn’t try to eliminate them; on the contrary, it tries to preserve them. Its goal is therefore to make settings of representation and treatment of knowledge smooth, and it is inspired from the human mental process. It leans on the mathematical fuzzy sets theory. This theory is an extension of the classic set theory for holding sets defined in an imprecise way.

3.4.1 Fuzzy ontologies
Fuzzy ontology combines the language of fuzzy logic with an ontological representation of knowledge (Parry, 2004). With the development of the semantic web, important needs for representation of uncertain information have appeared. Domain ontology can be extended to fuzzy ontology by adding a set of membership value to each concept of domain ontology and fuzzy relations between fuzzy concepts (Liang et al., 2006).

The relation between terms is an important concept in fuzzy ontology. The fuzzy ontology can be constructed from different kinds of relations between terms.

3.4.2 Fuzzy ontologies and IR
Research work dealing with fuzzy ontology are not numerous to describe their construction methods (Thanh et al., 2004) (Parry, 2004) (Widyantoro et al., 2001). The insertion of the fuzzy logic or the ontology in the process of information retrieval has improved the quality and the precision of the returned results. Thus, integration of the fuzzy ontology in the IR process is an interesting area of research and can lead to more relevant results than in the case where ontology and fuzzy logic are used individually.

4. Enrichment of SIRO by fuzzy ontologies
Following the survey that we made on methods dedicated to fuzzy ontology construction, we have noticed that these methods are created from database or existing ontology.
Our proposed system for on-line information retrieval is based on a domain ontology. Thus, only the methods taking domain ontology in entry are adapted. These methods differ themselves according to the following points: initialization methods of adherence values, treated relations, update methods and implication of user profile. The enrichment of SIRO by fuzzy ontologies consists in query reformulation using fuzzy concepts.

We consider an initial query \(Q(t_1, ..., t_n)\). A process of elimination of the empty words and lemmatisation must be performed first (the same process that is in SIRO). Terms used in the query \(Q\) can be ontological terms. For each term \(t_i\):

- If \(t_i\) is in the ontology: components that are in relation with this concept or one of its components are added such as the weight if they respect the following condition: the weight must be superior to a given threshold,
- If one of the term’s synonyms, hyponyms or hyperonyms belongs to the ontology: this component and its concepts are added when the weight of the relation is superior to given threshold,
- If neither \(t_i\) nor its components belong to ontology term, the term in the query is kept.

By using the vector model the query vector will be pondered as follows:
- If the term belongs to the initial query, the weight is equal to 1,
- If the term is added from the ontology, the weight is equal to its adherence value in the ontology.

5. Implementation and experimentations
The first step was the implementation and experimentation of the proposed system without the fuzzy ontology.

5.1 SIRO
SIRO is implemented in Java. It provides an online service and uses the Jena Api to handle ontologies and Google Api to search through the Web. The resulting Web pages are parsed using DOM.

In this section, we detail an example of execution of SIRO. In this example the user wants to search for hotels. He then enters the query: “find hotel” and requires a maximum number of document equal to 10.

Thus, the system displays a list of possible meanings related to the query’s keywords. The user chooses the word’s meaning which seems adequate. He can also view the tourism domain ontology and use some of its concepts in the query. This is schematized in figure 4.

After the enrichment process and the processing of the documents found by Google, the system displays the url's of the selected documents (figure 5). The system can also display the similarity’s measures relative to the most relevant documents. The user can also view the corresponding Web pages.
The user can then activate the option “find services” to classify the most relevant documents and to reformulate his query described in figure 6.

In order to evaluate our system, we use the classical precision measure. We compare these measures with two other systems; the first one, Lucene, is a traditional information retrieval system based on keywords search. The second one is based on a query reformulation, but without using the vector model. Figure 7 presents the results in terms of precision.
Figure 7: SIRO comparison with other IR systems.

The test was realized on 15 queries in the tourism domain and evaluated by 10 users. Our system is better than the two other systems in terms of precision.

5.2 SIRO Fuzzy

Initially we have the domain ontology described in figure 8.

Every user of our system will have his own fuzzy ontology expressing his needs. This is due to the fact that, after each new research values of adherences are updated, and then adapted to the user's queries.

As an example, let us consider two users and show how their queries are reformulated according to their own fuzzy ontology.

The fuzzy ontologies depicted in Figure 9 and Figure 10 are obtained after executing queries.

User 1:
For the query: find hotel in Paris

The new query is: hotel Paris Suite Room Rate Restaurant luxury Name Address

Therefore, detected services for this query are:
- Service: Lodging_hotel, Restaurant trade
- Activity: Hotel_search, Restaurant_Search, Disponibilitie_Verification,Reservation
- Task: Search_hotel_name, Search_hotel_address, Search_restaurant_address, Search_Star_number, verify_room_number, Verify_room_type, Search_Chef_Name, Search_food_type,
User 2:
For the same query: find hotel in Paris
The new query is: hotel Paris Suite Room bungalow Rate motel Name Address
Therefore, detected services from this query are:
Service= Lodging_hotel.
Activity= Hotel_search, Disponibilitie_Verification, Reservation.
Task= Search_hotel_name, Search_hotel_address, Search_Star_number, verify_room_number, Verify_room_type.

By using fuzzy ontology, every user is allowed to have more and more (after a certain utilization of system by the same user) personalized requests which improve the precision compared to SIRO (see figure 11).
Actually, the implemented version of SIROF has all links pondered by the same formula and the request vector is pondered in the same way than in SIRO. Therefore, implementation of the other possibilities of this approach is under experimentation and validation.
Actually, the implemented version of SIROF has all links pondered by the same formula and the request vector is pondered like SIRO. Therefore, implementation of the other possibilities of this approach is under experimentation and validation.
5.3 Results analysis and future work

The classical evaluation measures, which are the most used to evaluate information retrieval system, are the precision and recall:

\[
\text{Precision} = \frac{\text{number(relevant document retrieved)}}{\text{number(retrieved document)}}
\]

\[
\text{Recall} = \frac{\text{number(relevant document retrieved)}}{\text{total number(relevant document)}}
\]

Our system is an online IRS and uses the web like a corpus so the difficulty is to know the total number of relevant document in the web. Also, the user of an online system is interested mainly in the precision of this system. For this reasons, we choose to evaluate only the precision measure.

Semantic based systems evaluation is a very hard task. The precision can not be calculated automatically, we make up a user driven evaluation protocol. We considered fifteen queries in the tourism domain and ten users. Every user sends queries to the system and obtains a list of documents. Actually, different experimentations with SIROF (SIRO Fuzzy) are under finalization: experiments supporting the fuzzy ontologies with two different weighting methods and two different request vectors. The results obtained are compared respectively according to recall and precision measure. The final tests are in progress. Results show a clean improvement in the SIROF.

After comparing the results obtained from SIRO and SIROF, with other systems we can notice a considerable improvement of results. In fact for 30 returned documents, SIROF presents a pertinence of 0.5 while the pertinence of SIRO is 0.45.

Figure 11: SIROF comparison with SIRO and other IR systems.

6. Conclusion

In this paper, we have presented our online information retrieval system, namely SIRO. The main principle of this system uses two ontologies, domain ontology and service ontology, in order to improve the relevance of the results presented to the user. We applied this system to the tourism domain and we noted that it makes possible to improve the precision of research and thus the relevance of the documents returned to the user. It gives the user the opportunity to detect the services of a precise domain. SIRO is a generic system. It can be applied to any field. In our future work, the user will have the possibility to upload from the Net other domain ontologies. This work also makes possible to provide preclassified and filtered documents in order to improve the construction of ontologies based on learning techniques (Ben Mustapha et al., 2007) (Karoui et al., 2006). In this context, we can easily detect new concepts and then enrich the domain ontology. SIRO has been extended to SIROF by integrating fuzzy ontologies. This extension using fuzzy ontologies consists in a user query reformulation. The obtained results are compared respectively according to recall and precision and show an improvement of 5%.

In the future, this system will be integrated to a prototype developed for building semi-automatically domain and service ontologies (Ben Mustapha et al., 2007). We will also automate the choice of the senses given by Wordnet with the domain. Moreover, our system can also take benefit from the generalisation/specialisation capabilities offered by the ontology. Another perspective is to spatially visualize the set of results, according to their similarity.

6. References


Thanh Tho Quan, Siu Cheung Hui1 and Tru Hoang Cao FOGA: A Fuzzy Ontology Generation Framework for Scholarly Semantic Web the 2004 Knowledge Discovery and Ontologies Workshop KDO’04


ACKNOWLEDGMENT
The authors want to thank Anne-Marie Vercoustre from INRIA Paris-Rocquencourt for the fruitful discussions about this work.