2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)
July 6-11, 2014, Beijing, China

A Fuzzy-Ontology-Driven Method for A Personalized Query Reformulation

Hajer Baazaoui-Zghal and Henda Ben Ghezala
Riadi-GDL Laboratory, ENSI, Manouba University, Tunis, Tunisia
{*hajerbaazaouizghal, henda.benghezala@riadi.rnu.tn*}

Keywords: Fuzzy ontology, individual ontology, personalization, query reformulation.

Abstract: Ontologies have proven their utility in the area of Information Retrieval. However, building and updating ontologies manually is a long and tedious task. Moreover, crisp ontologies are not capable to support uncertain information. One interesting solution is to integrate fuzzy logic into ontology to handle vague and imprecise information. This paper presents a method for individual fuzzy ontology building. The key aspects in our proposal are: (1) an automatic building of an individual fuzzy ontology; (2) a query reformulation based, on the one hand, on the weights associated with the concepts and all existing relations in the fuzzy ontology and, on the other hand, on users' preferences, (3) an update of the membership concepts and relations' values after each users search, and (4) the use of the proposed fuzzy ontology and service ontology to individually classify documents by services. Our method has endured a twofold evaluation. Firstly, we have evaluated the impact of the update and the weights' variations on the search results. Secondly, we have studied how the query reformulation has led to a quality results improvement, both in terms of precision and recall.

1 INTRODUCTION

Ontologies are considered as knowledge structures allowing representation of the main concepts, relations, instances and properties in a specific domain. They have gained much importance, not only in the field of artificial intelligence but also in the fields of Information Retrieval (IR) and knowledge representation. On the one hand, ontologies have been used intensively in semantic web and contributed to the success of semantic search engines. On the other hand, fuzzy logic is used in Information Retrieval Systems (IRS) to solve the ambiguity and vagueness issues, by defining flexible queries (Tamani et al., 2013) or fuzzy indexes (Shih et al., 2011).
A fuzzy ontology is defined as an extension of crisp ontology by adding a set of membership degrees to each concept of the domain ontology and adding fuzzy relations among these fuzzy concepts (Parry, 2006). Several fuzzy ontologies' approaches have been developed ((Lee et al., 2005; Colleoni et al., 2009; Chien et al., 2010)) in which different aspects have been studied: fuzzy concepts and relations definitions, and integration of fuzzy ontologies to IR process... Some authors, have given special attention to how to construct fuzzy ontologies from fuzzy database models. For example (Quan et al., 2006) have proposed a fuzzy ontology framework (FOGA) that can generate a fuzzy ontology from uncertainty data, based on Formal Concept Analysis (FCA) theory. Nevertheless, fuzzy ontologies are faced with several main problems. Ontology defines relations between concepts, such as synonymy or hyponymy, which are too limited to describe the real world, with all its ambiguity and vagueness (Parry, 2006; Chien et al., 2010). Moreover, manual generation of fuzzy ontology from a predefined concept hierarchy is a difficult and tedious task. In addition, this latter often requires expert interpretation. As such, automatic generation of concept hierarchy and fuzzy ontology from uncertainty data of a domain is needed (Quan et al., 2006; Lee et al., 2005). Furthermore, users have different needs and specific goals when searching for information. In this context, the individual ontology generates personalized results and is useful for specific data domains ranging like genomics, human anatomical reference ontologies...(Coalter and Leopold, 2011). So, adapting fuzzy ontology to users needs (individual fuzzy ontology) could be an interesting track. Implicit feedback presents an interesting track for more personalized results. Several researches take into account positive user's feedback, but we believe that even negative preferences can significantly improve the retrieved results. To tackle

978-1-4799-2072-3/14/$31.00 ©2014 IEEE

these problems, we present a novel and complete method for individual fuzzy ontology building. In a previous work (Baazaoui et al., 2008) we detailed a SIRO system composed of three main modules: query processing and enrichment, search and document processing and finally a module for service classification. The possibility of its extension by fuzzy ontologies has been exposed. In this paper, our work aims at improving the performance of the query reformulation task and giving more personalized results. Indeed, we present a novel and complete method for individual fuzzy ontology building. We bring three main contributions in relation to query reformulation: (1) an automatic method for individual fuzzy ontology building; (2) a personalized query reformulation based, on the one hand, on the weights associated with the concepts and all the relations' types existing in the individual fuzzy ontology, and, on the other hand, on the users' preferences and (3) the use of the proposed fuzzy ontology and service ontology to individually classify documents by services.

The rest of this paper is organized as follows: in section 2 we give an overview of works related to IR and fuzzy ontologies. Our method is introduced and detailed in section 3. Section 4 presents and discusses experimental results of our method. We conclude and give some future work in section 5.

2 OVERVIEW AND MOTIVATIONS

In general, fuzzy ontology combines fuzzy logic with ontological representation of knowledge. According to (Parry, 2006) "in a fuzzy ontology each index term or object is related to every term (or object) in the ontology, with a degree of membership assigned to the relationship and based on fuzzy logic" (usually ontologies'components include instances or objects). The fuzzy membership values μ is used for the relationship between concepts, where $0 < \mu < 1$, and μ corresponds to a fuzzy membership relation such as "strongly", "partially", "somewhat","slightly" etc, where for each concept: $\sum_{i=1}^{i=n} \mu_i = 1$; n is the number of relations a particular object has, where $n = (N-1)$, with N representing the total number of objects in the ontology (Parry, 2006).

The integration of the fuzzy ontology into the IR process is an interesting and challenging area of research and can lead to more relevant results than in the case where ontology and fuzzy logic are used separately (Chien et al., 2010; Bordogna et al., 2009; Calegari and Ciucci, 2006). Several existing IRSs (Chien et al., 2010; Calegari and Ciucci, 2006) use semi-automatic or automatic methods, which allow the fuzzification of "IS-A" relations. Classification based only on domain ontology could not take into account the dynamic aspect of fuzzy ontology, mainly when the aim is to improve query reformulation and information retrieval results. Moreover, all relations are important mainly in case of query reformulation. Otherwise, "individual ontology" is considered as a "person's ontology" as opposed to a "global" ontology more relevant to the individual (Soshnikov, 2005; Bennett and Theodoulidis, 2009). It is important to note that individual fuzzy ontologies could be used to generate more personalized results. In a previous work (Baazaoui et al., 2007) we defined three ontologies, namely a *generic* ontology of web sites structures, *domain ontology* and *service ontology*. Ontology of domain services specifies for each service, its provider, its interested users, possible process of its unrolling, main activities and tasks composing this service. This ontology contains axioms specifying the relations between domain services and precise main domain concepts which identify each service. We have proven that domain and service ontologies are strongly correlated, hence the interest of service classification which improves the semantic search. Based on such a study, and motivated by the desire to guide user's query reformulation, we propose our new method FuzzOntoPerQ (Fuzzy-Ontology-based method for a Personalized Query reformulation). The originality of the work described in this paper consists of the following points: (1) building an individual fuzzy ontology supporting all the relations present in the initial ontology (not restricted to "IS-A" relations), (2) updating the membership values of concepts and relations after each user's search to adapt fuzzy ontology to users' needs (individual fuzzy ontology for more personalized results), (3) taking into account positive and negative users' preferences and (4) using the service ontology in addition to the fuzzy ontology to classify by service the results related to a user query.

3 A FUZZY-ONTOLOGY-DRIVEN METHOD FOR A PERSONALIZED QUERY REFORMULATION

The general structure of our method is given in Figure 1. The subsections below provide details on our individual fuzzy ontology building and the proposed query reformulation steps.

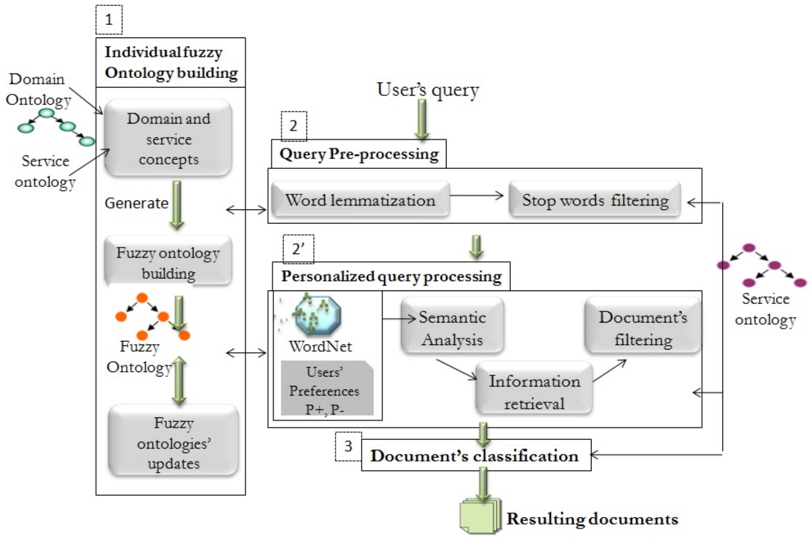

Figure 1: General architecture of FuzzOntoPerQ.

3.1 Individual fuzzy ontology building

In this subsection, we first give the Fuzzy Ontology definition.

Definition 1. *The formal fuzzy Ontology structure is defined as follows:*
$O_{fuz} = \{C, R, A\}$*, where C is a set of fuzzy concepts, R is the set of fuzzy relations and A is a set of Axioms expressed in a logical language.*
Let us consider an ontology set S=$\{O_{f1}, O_{f2}, \ldots O_{fn}\}$*, where* $O_{f1}, O_{f2}, \ldots, O_{fn}$ *are fuzzy individual ontologies.*

The individual user ontology represents knowledge about a user and covers the main aspects of the users' activities. The building process of the individual fuzzy ontology is shown in component 1 (*cf.* Figure 1) and is based on domain, service and fuzzy ontologies. The formal definition of the initialization and updating of membership values, are detailed below.

3.1.1 Initialization of membership values

Information Content (IC) is an important dimension of word knowledge when assessing the similarity of two terms senses. Information theoretic approaches propose to obtain the needed IC values by statistically analyzing corpora (Resnik, 1999). To compute *IC*, we use the formula introduced by (Seco et al., 2004) which is based on the structure of the ontology hierarchy. In fact, this frequency has the advantage of bringing the occurrence frequency of the concept itself and the concepts it subsumes, which allows supporting all relations' types.

$$IC(c) = 1 - \frac{\log(hypo(c)+1)}{\log(nc)} \quad (1)$$

where the function $hypo(c) \in N$ represents the number of c (concept) subclasses and nc the total number of concepts in the hierarchy. Initially, $IC(c)$ value is assigned to every concept c in the ontology and for every relation between two concepts c_1 and c_2. In (Jiang and Conrath, 1997) the authors suggested a new Link Strength (LS) function, which is simply the difference between the IC values of two concepts. As our goal is to obtain a weighted LS function, we assign weights depending on the relations' types according to equation 2.

$$LS(c_1, c_2) = \mid (IC(c_1) - IC(c_2)) \mid \times TC(c_1, c_2) \quad (2)$$

where $LS(c_1, c_2)$ is the link strength between c_1 and c_2, and $TC(c_1, c_2) \in [0, 1]$ a weight that reflects the relation type. For the choice of the TC value, we conducted several experiments by varying the TC value. Then, we concluded and adopted different ways to test the importance of this weight in the IR process, by varying TC values according to the type of relation (we associate it with the relations' types):

- For the specialization or synonymy relations (IS-a): $TC = 1$
- For part-of relations: TC depends on the number of concepts sharing this relation (for example, considering three concepts c_1, c_2, c_3 which are part of the same concept c then $TC = 1/3$),
- For all other relations : $TC = 0.5$ (to take into account relations which have a high value of LS).

3.1.2 Updating the membership values of the concepts and the relations

We suppose that a defined fuzzy ontology is not available in any context and so should be updated. Thus, it is necessary to define an update process of fuzzy values, taking into account the users' needs. The membership value should consider the previous values, the retrieved documents and the query. In the literature, there are researchers that have presented similar ways of updating membership values (Calegari and Ciucci, 2006).

Updating the membership values of the existing concepts in the user's query

Inspired from the *tf-idf* measure and based on the notion of context, the Context-Dependency (CD) measure has been used for the fuzzification of domain ontology (Sayed et al., 2007). In our work, we have extended *CD* to take into account the web retrieved documents and support the concepts' updates. When a query involving a particular concept, which exists in the fuzzy ontology, is performed, the method updates the membership value of this concept c using the following formula:

$$\mu_{new}(c) = \mu_{old}(c) + \frac{CD(c) - \mu_{old}(c)}{Q+1} \quad (3)$$

All the membership values of the concepts existing in the extended query will then be updated.

Updating the membership values of the relations related to the existing concepts in the user's query

After updating the concept's membership mentioned in the query, the method updates the membership values of the relations involved in this concept. For a relation R between the concept c_1 and c_2, the new membership value of $R(c_1, c_2)$ is:

$$\mu_{new}(R) = \mu_{old}(R) + \frac{|\mu(c_1) - \mu(c_2)| - \mu_{old}(R)}{Q+1} \quad (4)$$

To show the purpose of the given formulas, we take as an example a query sent by a user containing the concept "*catering*". The method computes the CD measures related to this concept and all its related concepts (like: restaurant, fast food. . .) using the returned documents selected by the user. Then, the membership value of this concept is updated using formula (3). Finally, the membership values of relations using formula (4) are also updated. For the same example the membership value of the relation between "*catering*" and "*restaurant*" will be updated.

3.2 The personalized query reformulation

The individual fuzzy ontology method that we propose to personalize the search results, has been integrated in the IR process to be used for query reformulation, and for documents and query indexing. The different steps of query reformulation are detailed below.

3.2.1 Query pre-processing

We consider an initial user's query $Q_{Ui}(t_1, \ldots, t_n)$. A process to eliminate stop words and lemmatization is first performed. For each term t_i:

- If t_i belongs to the ontology: this concept is then expanded to other linked concepts sharing a link weight greater than a fixed threshold α. We fixed α at 0.2 because after variation of this value, we remarked that under this weight, relations are not significant. The concept is also linked to its properties,
- If one of the term's synonyms, hyponyms or hyperonyms belongs to the ontology: this component, its concepts (which share a link weight greater than a fixed threshold α) and its properties are added to the query,
- If neither t_i nor its components belongs to any ontology concept, the term in the query is kept. Using the vector model, the query vector is weighted 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 membership value in the ontology.

After this step the updated query Q_{Ui} is obtained $Q_{Ui}(t_1, \ldots, t_k)$

3.2.2 Personalized query processing

First, an enrichment that relies on WordNet [1] is provided. It exploits synonymy, hyponymy and hyperonymy relations and the senses concepts in order to choose the most appropriate senses for the concept from the Wordnet definition list. A semantic disambiguation is important to choose only the most appropriate sense for the concept from its list of definitions (*cf.* Algorithm proposed for a model-driven approach of ontological components (Baazaoui et al., 2007) and

[1] http ://wordnet.princeton.edu/

adapted to this work).

Users' preferences enrichment

For more personalized results we analyze the terms of the query to keep only those that user prefers. The user's feedback is an essential step that enriches the query by relevant terms. User feedback provides, explicitly or implicitly users' interests and allows a continuous update of users' preferences. However, implicit feedback requires less intervention to users and avoid asking the user to explicitly evaluate the IR result. Moreover, both positive and negative users' preferences are important for query reformulation. Our idea is to enrich the query by using the implicit negative and positive implicit feedback.

Definition 2. *We consider a set of users' preferences: P_{Ui}={P^+_{Ui}, P^-_{Ui}} where P^+_{Ui} is a subset of positive preferences and P^-_{Ui} is a subset of negative preferences.*

We compare each element of P_{Ui} to Q_{Ui}, eliminate (intersection) terms of Q_{Ui} belonging to P^-_{Ui} subset and add (union) terms of P^+_{Ui} to Q_{Ui}.

The P_u set is updated after Wordnet enrichment. After the query is submitted to a web search engine (Google), we obtain a first list of documents, dynamically refreshed by Google.

Semantic analysis

Each returned document (except pdf or word are downloaded from the corresponding URL into an HTML document) is parsed using DOM. The next step consists of extracting text and performing a morphological analysis with TreeTagger and getting the word's lemmatized form. The existing concepts appearing in both the domain ontology and in the user's query are then detected.

Document's filtering

Each document is represented by a vector $D_j = (d_{1j}, d_{2j}, d_{3j}, \ldots, d_{Nj})$ where d_{ij} is the weight of the word in the document, and each query by a vector $Q_{Ui} = (q_1, q_2, q_3, \ldots, q_N)$, where q_j is the weight of the word in the query. q_i is weighted as mentioned in the previous step. As indexing is an essential part of the IR task, a vector of index term weights is computed. The most often schema tf-idf (Salton and Buckley, 1988), is used in order to give an equal chance to all the documents without giving a greater importance to long documents. In order to rank documents, and find the most similar documents, the vector model presented by (Salton and Buckley, 1988) is adapted: terms are substituted by concepts. The similarity between the user's query and a document is computed with the cosine formula:

$$Sim(D_j, Q_u) = \frac{\sum_{i=1}^{N} d_{ij} q_i}{\sqrt{\sum_{i=1}^{N} d_{ij}^2 \bullet \sum_{i=1}^{N} q_i^2}} . \quad (5)$$

Therefore, a second ranked list by ascending order according to the documents' similarities with the query is obtained. As an example, the initial query sent by the user was: search catering in Beijing. This query is then morphologically analyzed and the stop words are eliminated. We obtain a first query catering Beijing. The sets of users' preferences are: P^+={Touristic_guide, Beijing, Aerial_Transport} and P^-={luxury_hotel, fast_food}, Beijing is not an ontological term so it is kept. *catering* belongs to the users fuzzy ontology and have several relations. The semantic analysis returns the following enriched query: Catering Beijing Chef Reservation Cuisine Restaurant Name Address. Since the concept fast_food belongs to the set P-, it is not added to the reformulated query. The query vector is $Q(1;1;0:8;0:9;0:7;1;1;1)$.

3.2.3 Document's classification

The classification that we propose classifies the results of a user's query by service. In addition to the fuzzy ontology; we have used the service ontology. The service ontology decomposes each service in activities and every activity in tasks (*cf.* Figure 2). Indeed, according to the big terminological dictionary, an activity is a set of elementary tasks or work executed by a person or a group and that lead to realize possessions or services. The goal is to determine

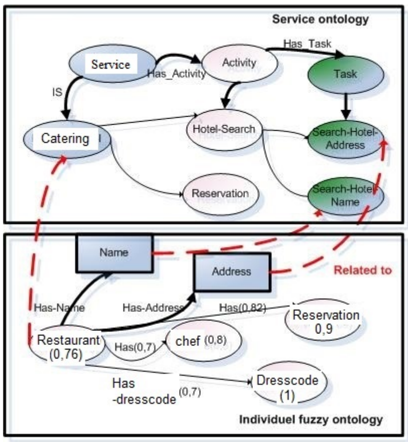

Figure 2: Domain and service ontologies of tourism domain.

the set of concepts and relations that identifies every service. This relation is inherited by the fuzzy ontology (because it is constructed from the domain ontology). The reformulated query contains a set of concepts contained in the fuzzy 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. Given the set of concepts

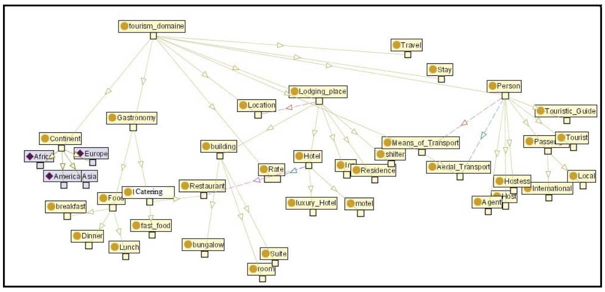

Figure 3: Domain ontology of tourism.

added to the query and the link between the fuzzy ontology and the service ontology, the method extracts all the services related to these concepts. The method adds the services related to these concepts and which have at least a relation with one of the added concept. This relation must have a membership value superior to a threshold 0.2 (this value is chosen after several empirical tests). Then, the vector model (Salton and McGill, 1984) is used to represent a service: $Serv_i = (c_1, c_2 \ldots, c_n)$, where n is the number of concepts related to a service. For each selected document, the similarity with the services using the cosine formula is computed. The document is then affected to the most similar service regarding this document. Finally, the documents are displayed by service.
To show the purpose of the given formulas, we take as an example the concept “Restaurant” that exists in the extended query and has the relation “has-chief” with the concept ”*Chief*”. The method will add the services related to the concept ”*Chief*” only if $has-chief(Restaurant, Chief) > 0.2$. In a previous example, the initial query was ”find hotel in Paris”. The services, activities and tasks detected from the concepts in the service ontology are:

- Service= Lodging_hotel, Restoration;
- Activity= Hotel_search, Restaurant_Search, Availabilities_Verifiation, Reservation;
- Task= Search_hotel_name, Search_restaurant_address, Search_Chief_Name, Find_menu.

The classification gives the user the opportunity to determine the categories of tasks, activities and services which are related to his/her query. The user can then choose the services corresponding to his/her needs. The query processing module captures the selected services and uses the relation between the service ontology and the fuzzy ontology to formulate a new query including the new concepts and relations detected. For example, by choosing the service ”‘*restaurant trade*”’, we will find the concept ”*restaurant*” in the new query. This latter is then sent to the search engine and the refinement process is iterated. For 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.

4 EXPERIMENTATION AND RESULTS ANALYSIS

In order to show that our method can have a great interest in a query reformulation and can contribute to improve the performance of the retrieval task, a system supporting our method has been developed. The system is implemented in Java, providing an online service and using the Jena Api to handle ontologies and Google Api to search through the Web. Several experiments were conducted to investigate the performance of our method.

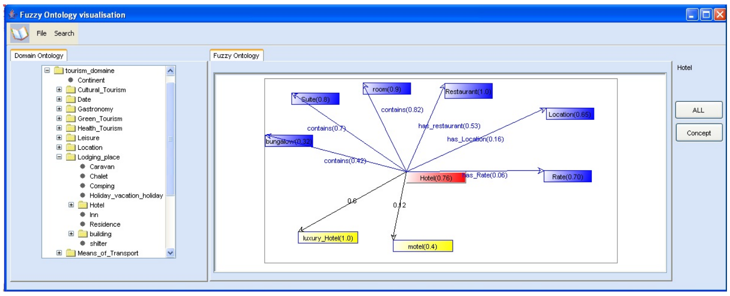

Figure 4: Concept and Relation’s visualization.

4.1 Experimental Setup

Our method has endured a twofold experimental evaluations aiming: (1) to evaluate the impact of individual fuzzy ontology (concepts’/relations updates and TC variations) on the query reformulation improvement, and (2) to compare our query reformulation results to those obtained with existing engines. We adopted a user-centered protocol. The used data for the experimentation and the evaluation were composed of a domain ontology and users’ queries. Indeed, we considered 30 queries in the tourism’s domain and 20 users. We have limited our corpus of documents to 200 documents because in general users are interested only to the 20 or 30 retrieved documents. The first 100 documents are evaluated by users in order to compute precision and recall. We designed five different scenarios to evaluate our proposal, which are:

- query reformulation based on domain ontology (denoted ***Sc1***),

- query reformulation based on individual fuzzy ontology without update and TC variable (denoted ***Sc2***),
- query reformulation based on individual fuzzy ontology without update and TC=1 (denoted ***Sc3***),
- query reformulation based on individual fuzzy ontology with 4 updates and TC=1 (denoted ***Sc4***),
- query reformulation based on individual fuzzy ontology with 4 updates and TC variable (denoted ***Sc5***),

To measure the query reformulation improvement, we compared our average recall and precision rates with two other systems: the first one is Google, a traditional search engine system based. The second one is Exalead (http ://www.exalead.com), a semantic search engine (we consider the simple keywords search results). We point out that we consider the simple keywords search results of the cited engines. Besides Table 1 depicts : (1) the results in terms of precision (P) and recall (R), for Top 5, 10, 20, 30 and 50 retrieved documents, and (2) the improvement rates (*Imp. FuzzOntoPerQ vs Exalead* and *Imp. FuzzOntoPerQ vs Google*).

4.2 Result's evaluation and discussion

In this section we present and discuss obtained results with the two carried out series of experiments.

First series of evaluation: Results of the assessed exact precision obtained with the five evaluation scenarios, are given by Figure 5. We observe that for 5 and 10 returned documents (P5 and P10), our best reformulation scenario, namely *Sc5*, provides the highest precision ($0,89$), which corresponds to the highest improvement rate ($27,14\%$) in comparison with a query reformulation based on domain ontology. Moreover, query reformulation scenarios (given by precision increasing order *Sc2, Sc4, Sc3*) lead an improvement in the exact precision at low recall (P5 and P10). Indeed, results highlight that using separate update and TC variation for query reformulation lead to an improvement of IR that depends on number of updates. The overall results prove the interest to jointly update the fuzzy ontology and to vary TC value. Furthermore, this means an increase in the number of the retrieved documents put in the head of the top ranked list.

Second series of evaluation: The results given by Table 1 highlight that our query reformulation proposal increases the exact precision at low recall (P5 and P10) versus simple keywords search. In fact, the best exact precision improvement ($34,84\%$) and the best recall improvement ($46,87\%$) were performed with our FuzzOntoPerQ (the scenario *Sc5* was considered for this evaluation) at 5 documents, in comparison to the Exalead engine. The highest improvement occurred at Top-5 average precision ($15,58\%$) comparatively to the Google engine. We can conclude that our proposal gets higher average recall rates and higher average precision rates than the simple search based on domain ontology. Moreover, our best query reformulation scenario (*Sc5*) yields significant improvement, which could be explained by the fact that our reformulation is based on users' preferences allowing the search process to support the set of positive and negative preferences. We can conclude that our proposal can be used to design personalized query reformulation taking into account the semantic aspects of the available information on the Web. This issue shows that documents are better ranked, and reformulating the query with personalized fuzzification, update and TC variable weighting improves the relevance of search results.

5 CONCLUSION AND FUTURE WORK

In this paper, we proposed a personalized query reformulation based on a fuzzy-ontology building method. Our proposal takes place in three main components: (1) the individual fuzzy ontology building, (2) the query reformulation taking into account fuzzified concepts,relations and users' preferences, and to be efficient, these two components necessitate another important one: (3) document's classification. So, our first contribution concerns the individual fuzzy ontology's building process. Our method considers automatic fuzzification of a domain ontology taking into account both taxonomic and non taxonomic relations. Our second contribution concerns the integration of our fuzzy ontology method into the query reformulation process, which is based on the weights associated with all the relations existing in the fuzzy ontology, and after the fuzzy ontology is used to classify documents by services. Finally, a system supporting our method has been implemented. Experiments and evaluations have been carried out, which highlight that overall achieved improvement are obtained thanks to the integration of fuzzy ontologies into IR process, updates and weights' variations. A fuzzification of positive and negative preferences is currently in progress, aiming a better improvement of query reformulation. In our future work, we aim at proposing an approach extending our fuzzy ontologies building from the web to use it along with fuzzy profile ontology for the web mining and IR.

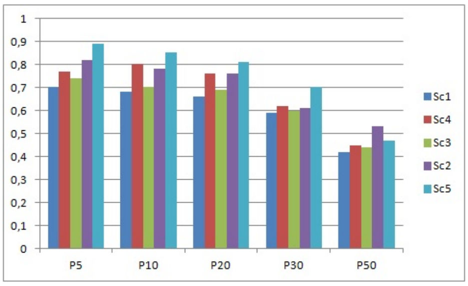

Figure 5: Query reformulation result's comparison.

	Top 5		Top 10		Top 20		Top 30		Top 50	
	P	***R***	***P***	***R***	***P***	***R***	***P***	***R***	***P***	***R***
Exalead	0,66	0,32	0,68	0,34	0,63	0,65	0,58	0,59	0,40	0,39
Google	0,77	0,40	0,75	0,40	0,71	0,72	0,62	0,65	0,42	0,42
FuzzOntoPerQ	0,89	0,47	0,85	0,46	0,81	0,80	0,70	0,71	0,47	0,45
Imp. FuzzOntoPerQ vs Exalead (%)	34,84	46,87	25,00	35,29	28,57	23,07	20,68	20,33	17,50	15,38
Imp.FuzzOntoPerQ vs Google (%)	15,58	17,50	13,33	15,00	14,08	11,11	12,90	09,23	11,90	07,14

Table 1: A comparison of the average recall rates and the average precision rates of the FuzzOntoPerQ with Google and Exalead.

REFERENCES

Baazaoui, H., Aufaure, M.-A., and Mustapha, N. B. (2007). A model-driven approach of ontological components for on-line semantic web information retrieval. *J. Web Eng.*, 6(4):309–336.

Baazaoui, H., Aufaure, M.-A., and Soussi, R. (2008). Towards an on-line semantic information retrieval system based on fuzzy ontologies. *JDIM*, 6(5):375–385.

Bennett, B. R. and Theodoulidis, B. (2009). Towards a notion of individual ontology. In *AIS-ADM*, pages 247–262.

Bordogna, G., Pagani, M., Pasi, G., and Psaila, G. (2009). Managing uncertainty in location-based queries. *Fuzzy Sets and Systems*, 160(15):2241–2252.

Calegari, S. and Ciucci, D. (2006). Towards a fuzzy ontology definition and a fuzzy extension of an ontology editor. In *ICEIS (Selected Papers)*, pages 147–158.

Chien, B.-C., Hu, C.-H., and Ju, M.-Y. (2010). Ontology-based information retrieval using fuzzy concept documentation. *Cybernetics and Systems*, 41(1):4–16.

Coalter, A. B. and Leopold, J. L. (2011). Roar: A reference ontology for anatomical relations. In *CIBCB*, pages 176–183.

Colleoni, F., Calegari, S., Ciucci, D., and Dominoni, M. (2009). Ocean project a prototype of aiwbes based on fuzzy ontology. In *ISDA*, pages 944–949.

Jiang, J. J. and Conrath, D. W. (1997). Semantic similarity based on corpus statistics and lexical taxonomy. *CoRR*, cmp-lg/9709008.

Lee, C.-S., Jian, Z.-W., and Huang, L.-K. (2005). A fuzzy ontology and its application to news summarization. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 35(5):859–880.

Parry, D. (2006). Evaluation of a fuzzy ontology-based medical information system. *IJHISI*, 1(1):40–51.

Quan, T. T., Hui, S. C., Fong, A. C. M., and Cao, T. H. (2006). Automatic fuzzy ontology generation for semantic web. *IEEE Trans. Knowl. Data Eng.*, 18(6):842–856.

Resnik, P. (1999). Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. *J. Artif. Intell. Res. (JAIR)*, 11:95–130.

Salton, G. and Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing and Management: an International Journal*, 24:513–523.

Salton, G. and McGill, M. (1984). *Introduction to Modern Information Retrieval*. McGraw-Hill Book Company.

Sayed, A. E., Hacid, H., and Zighed, D. A. (2007). Using semantic distance in a content-based heterogeneous information retrieval system. In *MCD*, pages 224–237.

Seco, N., Veale, T., and Hayes, J. (2004). An intrinsic information content metric for semantic similarity in wordnet. In *ECAI*, pages 1089–1090.

Shih, W.-C., Yang, C.-T., and Tseng, S.-S. (2011). Fuzzy folksonomy-based index creation for e-learning content retrieval on cloud computing environments. In *FUZZ-IEEE*, pages 965–970.

Soshnikov, D. (2005). Multi-agent approach for community clustering based on individual ontology annotations. In *AIS-ADM*, pages 247–262.

Tamani, N., Lietard, L., and Rocacher, D. (2013). A relational division based on a fuzzy bipolar r-implication operator. In *FUZZ-IEEE*, pages 1–7.