Augmenting semantic queries using Personalization techniques

Yannis Plegas, Evangelos Sakkopoulos, Athanasios Tsakalidis

Research Academic Computer Technology Institute
Internet and Multimedia Technologies Research Unit
N. Kazantzaki Str. 26500 Rion, Patras, Greece
Computer Engineering and Informatics Dpt
University of Patras,
GR-26504, Rio Patras, Hellas
plegas@ceid.upatras.gr, {sakkopul, tsak}cti.gr

Abstract

The semantic Web has brought exciting new possibilities for information access and electronic commerce. Semantic Web is already adopted in several applications and solutions. The main objective is to create a powerful solution to unify a great percentage of different operations that concern the personalization/ categorization of semantic questions. Key aim of the proposed solution is to enable semantically enriched searching techniques. In this work, we discuss the conducted experiments upon an implemented prototype. Results have been encouraging and indicated strongly that the proposed solution is effective.

keywords: semantic queries, personalization techniques, ontology, domain, group of categories, synonyms, hyponyms, factor of similarity.

1. Introduction

The Web was designed to be a universal space of information and mainly offers unstructured and semi-structured natural language data. The Semantic Web [Berners-Lee et al (2001)] is specifically a web of machine readable information whose meaning is well defined. The ontologies [Dogac et al (2002)] for the Semantic Web are an emerging technology which offers a promising infrastructure as far as the harmonization of the heterogeneous representations of web resources is concerned. In this direction, ontologies offer a common understanding of a domain that can be a mean of communication among application systems and people.

A typical hypermedia application is static and serves the same interface and the same set of links to all users. In order to improve usability, adaptive web-based applications make it possible to deliver personalized views of a hypermedia document. These views differ for users with diverse needs and knowledge backgrounds which have access to the system [Adamopoulou et al (2005)].

The main objective of this work is to present a promising approach that will unify a number of different operations for the personalization of semantic questions. It is an early approach that investigates the effects of personalization techniques upon semantically enhanced query engines. Aim of this work is to present a solution that will at the end personalize the results of semantic questions by categorizing the results into a well-known widely used set of categories (i.e. ODP).

Search engines partially address this problem by ranking pages returned as a result, according to their popularity in the web. Conclusively, users usually browse the results with a view to selecting the answers of their interest. Nevertheless, the majority of users spend long time to browse the results. Successively, the users have in their mind specific categories of results that correspond to their information need. The latter fact (even subconsciously) plays a critical role for the users so as to rule out all the unwanted results.

Search engines, such as Google and Yahoo!, tag their search results according to the categories they belong to. The maintenance of search categories and the tagging of results constitute a semi-automatic procedure. It is carried out mainly by human intervention, due to the vast volumes of web pages which crop up by the web every day. This manual procedure is assisted in part with the development of the Open Directory Project (ODP), a publicly available hierarchy of thematic categories in RDF format.

Our solution combines the information provided though ontologies with the information of ODP categories. In particular, we make a semantically enhanced question. Then, we compute the factors of similarity among the classes of the question's results and the categories of ODP and finally and we match the categories which the results belong to.

The outline of our approach in steps is as follows: We first make a semantic question in the ontology with our search engine. We analyze the results and we find the classes of ontology in which these results belong to. Then we find the synonymous words in these categories with WordNet and we construct groups of categories. Having the categories (with their synonymous words as group) and the categories of ODP we calculate the factors of similarity among them using the piece of WordNet that find the similarity between two words. In the end we return the top-k most relative categories for every group, categorized by the factor of similarity. These steps are depicted in short in the Figure 1 below. Analysis of the functional specifications follows in section 3.

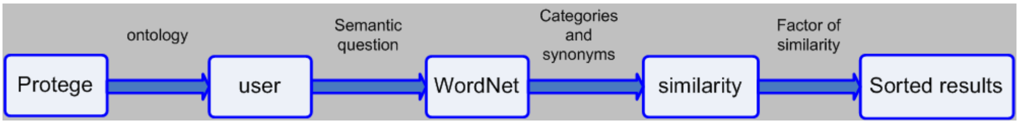


Figure 1. *High level schematic of the proposed functionality*

In the sequel, the paper is organized in the following sections. In section 2, we give our motivation and relative work. In section 3 we present the design and the functional parts of our approach. In section 4, we give shortly the technological background utilized. In section 3; we discuss our experimental paradigm using a number of different queries to verify and test our method. Finally in section 6 concludes the paper and presents future steps.

2. Motivation and Relative Work

Semantic web technology is already adopted in several web based applications and solutions [Makris et al (2006)], [Sakkopoulos et al (2006)] marking in this way a new era in the Internet technologies. The use of web structure and categories [Cooley (2003)] play an important role for the personalization of search results. There is a wide range of personalization techniques [Brusilovski (1998)] which optimize search results. A well-known technique includes the construction of user profiles [Garofalakis et al (2002)] that keep user model data in order to perform adaptive customization of results. However, users tend to avoid registration procedures and therefore such a technique is not advisable for user based search applications. In Dumais et al (2001) and in Garofalakis et al (2005), the search results have been concentrated into categories using powerful techniques such as SVM or pre-existing category tags on web pages. Personalization of searching results (using ODP categories) already adopted by a great variety of popular search engines such as Google and Yahoo. In the work of [Makris et al (2006)] an automated personalization technique is presented that takes into consideration the ODP categories implicitly from the user submitted queries to categorize (post) search results.

In this section we also present the new elements which are provided in our application. Furthermore, we compare the existing technologies with the corresponding parts of our application. The tools which used in the Web (like Protégé) return only the categories that match exactly (keyword matching) with the categories of the question. Because of this, these tools don’t use personalization in the results. In particular, the questions are static in Protégé; we cannot select the type and the parameters of the question. So, the questions are predetermined, not dynamic. What is more, there is no suppleness in the results. Protégé returns as a result only instances, while it would be better return classes and slots also. On the contrary, our solution is developed in purpose to personalize the results. In order to achieve this, we use the following approach:

We developed a tool in submission online questions in a semantic structure – ontology. The questions are dynamic and not static (as they are in Protégé). That is, we can select the type and the parameters of the question (while in the Protégé questions are predetermined). We created a function which takes the question's results, and returns the classes of these results (from the database of ontology).
The questions return as a result instances and classes-categories and not only instances as in Protégé takes place. So, we don't need always to process the instances in order to take their classes-categories for results. Based on the analysis of classes we make a second categorization which gives us the desirable personalization of the system. We improve the results because we expand the categories with their synonymous words. So, the results are satisfactory even if the categories of the results are few. We find the factor of similarity between the group of initial categories of instances (and their synonymous), and the categories of ODP (rather, the existing technologies usually do keyword searching). With a view to finding the factor of similarity we use the tree structure in which the words are organized in the WordNet. In that structure, we apply the function of similarity. The function of similarity computes the distance Wu and Palmer [Wu and Palmer (1994)]. We create a list which keeps for each group of categories, the top-k (top-10) highest factors of similarity. In this list, all the categories are imported in. Every time that an element is imported in, the list checks if the number of elements in a group is bigger than ten and it extinguishes the last one. We return the personalized results, based on the classification of the question's results and its similarity to the categories of ODP (the answers are categorized simultaneously with their entry in the above list).

Conclusively, our application finds the similarity between two words though they are not the same. So, we earn in suppleness and in quality of results. The first one is the most important because we can express precisely what we think about, without trying to adjust it (to keyword searching). Finally, the quality of the results is improved in because we increase the results of questions in finding the synonyms of the question's results. In the next section we present the design and the functional specifications of our approach.

3. Design and Functional Specifications

In this section, analyze the functionality of our approach. In Figure 2 appears the total structure of our solution, it separated in five sub systems, the Protégé one (creates or imports an ontology), the user's one (makes a semantic question in our data base), the sub system of WordNet (finds the synonyms of categories), Similarity one (computes the factor of Similarity among the categories) and last but not least, the sub system of personalized results (sorts the categories of ODP). These subsystems sequentially produce our results. The input of the whole system is this one of subsystem Protégé and the output is this one of subsystem results. Every output of subsystems is the input of the next subsystem. The total process that we follow appears in Figure 2. We present analytically the sub systems that constitute our application and afterwards.

3.1 Ontology Editing and Knowledge Base

We can create our ontology in Protégé or import an existing one. The ontology is written either in RDF or in OWL [Dieter et al (2003)]. The Protégé provides various tools with which we can graphically draw the ontology. As the ontology is created by these tools, the database of the ontology (knowledge base) contains the information for all the classes, the slots and the instances. Whenever we change something in the ontology, the Protégé updates the database of the ontology.

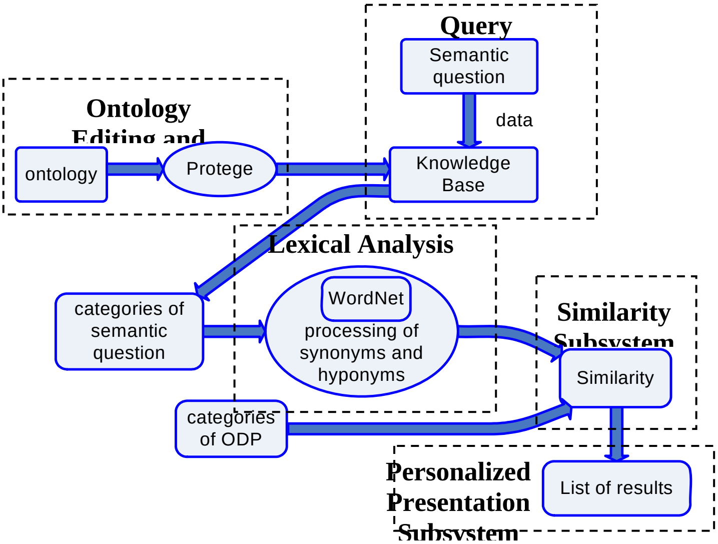


Figure 2. *Functional Flow Chart*

3.2 Query subsystem

We have implemented a set of functions with a view to posing semantic questions onto knowledge bases. Furthermore, each function accepts the entries of the question as parameters. The most basic entry, which is common for all the questions, is the location of knowledge base that contains all the information for the ontology (s1). The entries differ, and depend on the question we make. We use the main and the search API of Protégé in order to submit the questions. Then, we receive the results and we process in order to store the results' categories, which are the ontology's classes.

3.3 Lexical Analysis subsystem

The sub system of Lexical Analysis appears in Figure 4. In order to find the synonyms of the categories' results, we integrated WordNet functionality. For each protégé query resulting category (classes etc), we produce the synonyms – hyponyms with a view to effectively match the ODP categories finally. Particularly, we create as many groups of categories as the number of the initial semantic results categories (classes etc). Every initial category with their synonyms consist a group of categories. The name of these groups is the corresponded name of the initial category. In conclusion, we store the results as senses.

3.4 Similarity subsystem

The Similarity subsystem appears in Figure 3.

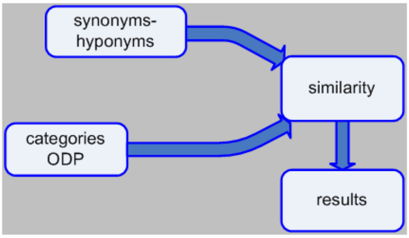


Figure 3. Similarity Sub system

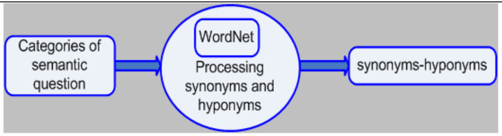

Categories of semantic question
WordNet
Processing synonyms and hyponyms
synonyms-hyponyms

Figure 4. Sub system WordNet

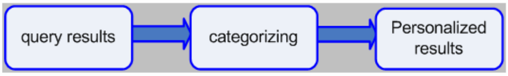


Figure 5. Sub system of results

The synonyms and hyponyms constitute the input of this subsystem which arises from the subsystem of WordNet and the categories of ODP.

In order to find the similarity among the categories, we use the algorithm Wu and Palmer which calculates distances and finds the factors of Similarity. This algorithm uses the tree structure with which the WordNet stores the words and according to the following formula calculates the factor of Similarity. The utilized [Wu and Palmer (1994)] similarity metric measures the depth of the two categories(a and w) in the WordNet taxonomy, and the depth of the Least Common Subsumer (LCS = c), and combines these figures into a similarity score.

$$Sw\&p = \frac{2*depth(c)}{depth(a)+depth(w)},$$

A number of different similarity measures is possible to be utilized without losing the generality – Comparative results of the similarity measures are not presented due to space limitations -. When the factor between two words is close to zero, entails that the meanings of the words are not similar, reversely when the factor is close to one, the meanings of the words are similar.

The output of this subsystem contains the name of the group, the categories of ODP and the factors of similarity. We replace the names of the categories in group with the name of the group because after this step, we compute the results of a group as an entity and not like different categories.

3.5 Personalized presentation subsystem

The Personalized presentation subsystem appears in Figure 5. The input of this subsystem is the factors of similarity among the categories of ODP and the groups of categories (names of groups). We process the query results and divide the results. In each part of the results we take the ten different categories with the highest factors of similarity. We refer to different categories because in the previous results every category of ODP exists more than one times. This happens, because the previous results contain the factors of similarity between every category of the question (initial categories an synonyms-hyponyms) and the categories of ODP. The factor of similarity for a category of ODP is the highest one among the others. Finally, since we check all the previous results, the final results contain the ten biggest categories for each group and consequently the groups of categories of ODP which string together the question.

3.6 Implementation issues

In this section we give the types of semantic questions that we can make with our search engine. These questions include meanings like classes, instances and slots. The classes are general entities, while the instances are specific instances of the classes. The slots finally, are attributes that have the instances of the classes. We use all these entities below, in the six types of questions that implement our application.

Global Knowledge base Search on Browser Text.
Global Knowledge Search on Any Slot Value.
Class Search for Instances.
Class Tree Search for Classes.
Class Tree Search for Slots.
Instance Tree Search for Instances through a Named Slot.

In the following, we discuss the technological framework of our approach briefly.

4. Technological Background

Before proceeding with the discussion of the experimental paradigm, in this section we briefly present the technologies involved. In this capital we don’t refer to Protégé directly, although it is the most important external tool of our application, because the several parts of Protégé will analyze simultaneously with the relative parts of application.

4.1 RDF and RDF Schema (RDFS)

A prerequisite for the Semantic Web is machine-processible semantics of information. RDF is a foundation for processing metadata; it provides interoperability among applications that exchange machine-understandable information on the Web. Basically, RDF defines a data model for describing machine-processible semantics of data.

The modeling primitives that are offered by RDF, are very basic. Therefore, the RDF Schema specification defines further modeling primitives in RDF. That is, RDF Schema extends (or enriches) RDF by giving an externally specified semantics to specific resources. This is only because of these external semantics that RDF Schema is very useful. Moreover, these semantics cannot be captured in RDF: if it could, then there would be no need for RDFS.

4.2 WordNet

WordNet is an online lexical reference system whose design is inspired by current psycholinguistic theories of human lexical memory. English nouns, verbs, adjectives and adverbs are organized into synonym sets, each representing one underlying lexical concept. Different relations link the synonym sets. It, is an electronic lexical database, is considered to be the most important resource available to researchers in computational linguistics, text analysis, and many related areas.

4.3 Asp.net

ASP.NET is a programming framework built on the common language runtime that can be used on a server in order to build powerful Web applications. ASP.NET offers several important advantages over previous Web developed models:

5. Experimental Paradigm

In order to validate our methodology we have developed an experimental application prototype using a series of technologies presented in Section 4. It is based on a Web application interface which is presented in Figure 6. In this section, we discuss the possible queries/questions that may be performed, the similarity validation procedures and the final categorizing of the results. We will discuss the experimental paradigm step by step and we will show how each one cooperates in order to achieve our final personalization aim.

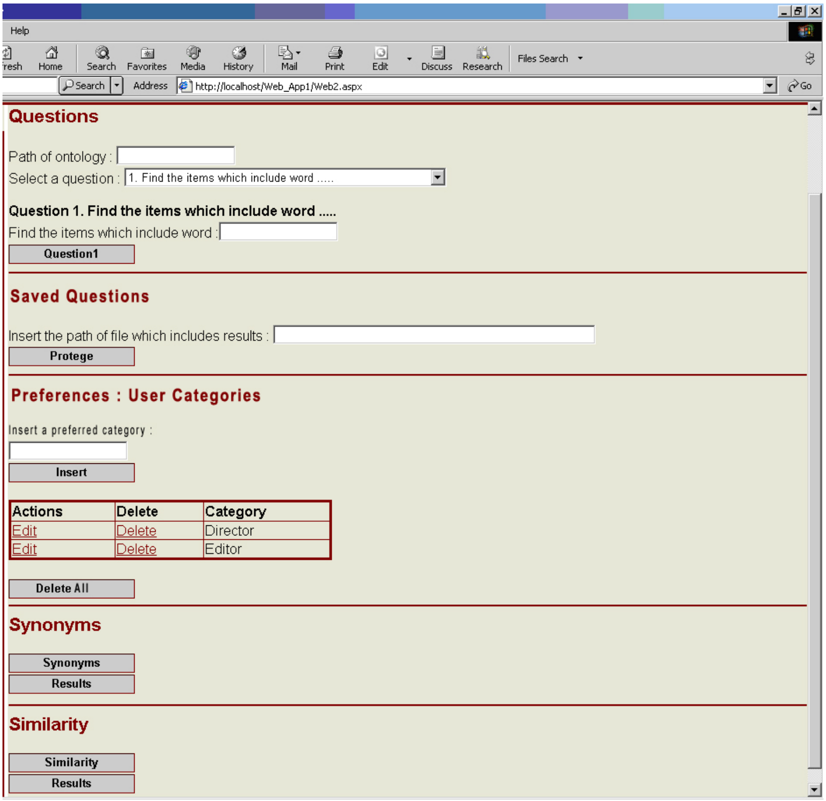
We will use the ontology newspaper which has the structure that appears in Figure 7. as the tree of classes. In particular, in the circles are the classes that have subclasses and in the squares are the classes that don’t have.

5.1 Searching options and Question types

A number of different query/question types may be posed through the application prototype in order to find the best possible information for personalization. Each one of them is presented in short below.

5.1.1 General search (Global Knowledge base Search on Browser Text).

When we make the question, the application searches the entire database and every registration which has the text, returns back. Then, the application checks these results of the question and it finds the categories (classes) of the results.



Questions

Path of ontology :

Select a question : 1. Find the items which include word .....

Question 1. Find the items which include word .....

Find the items which include word :

Saved Questions

Insert the path of file which includes results :

Preferences : User Categories

Insert a preferred category :

Actions	Delete	Category
Edit	Delete	Director
Edit	Delete	Editor

Synonyms

Similarity

Figure 3. *Experimental paradigm*

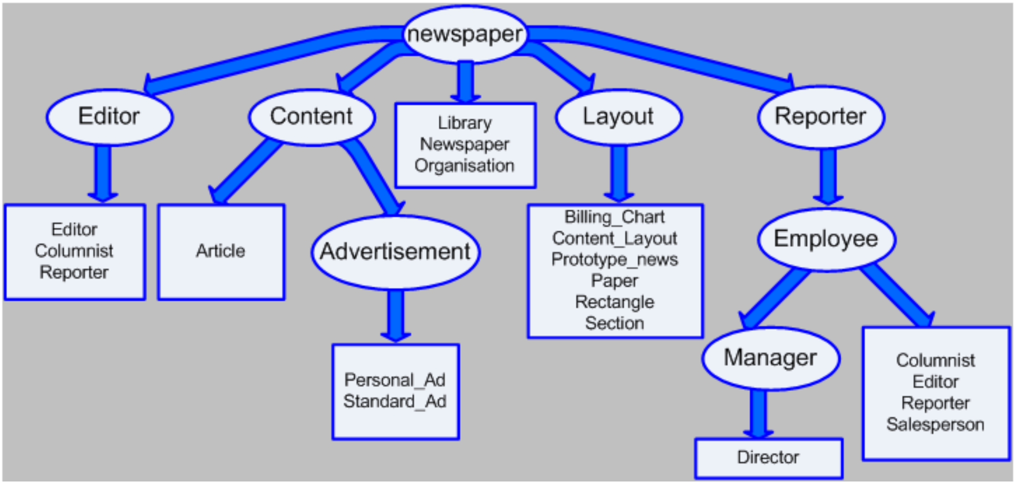


Figure 4 : *Ontology Schematic Newspaper*

5.1.2 General search in the data of all slots (Global Knowledge Search on Any Slot)

When we make this question, the application searches all the slots and every registration which has the text, returns back. Then the application checks these results of the question and it finds the categories (classes) of the results.

5.1.3 Search for instances in a class (Class Search for Instances).

When we make a question of this type, the application searches all the instances in the class and every registration which has the text, returns back. Then the application checks these results of the question and it finds the categories (classes) of the results.

5.1.4 Search for classes in a concrete tree of classes (Tree Search for Classes).

When we make this question, the application searches all the classes in the tree of classes and every registration which has the text, returns back. Then the application checks these results of the question and it finds the categories (classes) of the results.

5.1.5 Search for slots in a concrete tree of classes (Class Tree Search for Slots).

When we make this question, the application searches all the slots in the tree of classes and every registration which has the text, returns back. Then, the application checks these results of the question and it finds the categories (classes) of the results.

5.1.6 Search of instances in a concrete tree of instances via a named slot.

When we make this question, the application searches all the instances in the tree of these ones that are reported in this instance from the slot which we import in. Every registration which has this text returns back. Then, the application checks these results of the question and finds the categories (classes) of the results.

5.1.7 Using saved results compiled at the Protégé Querying Interface.

To take advantage of the Protégé query interface, the application gives us the possibility to process the results from a question performed using Protégé. Protégé saves the former results into files, whose path may be configured and be utilized by our application. As a next step, the application checks the results of the question and finds the categories (classes).

5.1.8 User defined categories.

Our application presents the results and gives the opportunity to the user to explicitly insert/choose extra categories besides the ontology based ones. One may modify a preference on his own.

Overall, the application prototype allows us to utilize any of the three different procedures (a) user query through the web interface of the application prototype, b) saved results compiled at the Protégé Querying Interface and c) user defined categories) to choose categories of interest which suit better to the question posed so that the results will be personalized in the best possible way.

5.2 Similarity Implementation and Verification

Every semantic question towards an ontology instance returns possibly a number of instances as results (classes-categories). Our approach stores the categories of the results in which the instances belong to or directly the classes (categories) which are returned depending on the submitted question to take advantage of them in combination with the ODP categories.

Our application pulls the different categories stored and detects/ matches the different meanings, the synonymous words as well as hyponyms (senses). Hyponyms are ancestors (categories) in the tree of categories in WordNet. The synonymous words are also found using WordNet. This process returns the different meanings, as well as the synonyms of each meaning and the hyponym.

Then, as the Figure 3 presents, we check the similarity of category (with both of synonyms and hyponyms) to the categories of ODP which are found in the table topics. We utilize the similarity that implements the Wu and Palmer [Wu and Palmer (1994)] algorithm. For demonstration purposes our prototype enables us to see the synonymous words and the hyponyms as it appears in Figure 8.

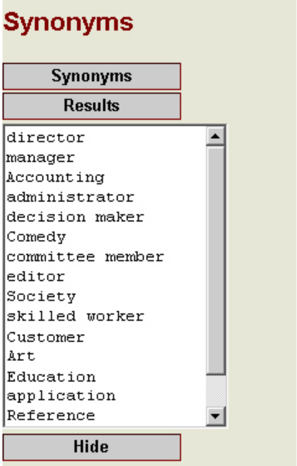
5.3 Results: Top-k categories that best match the initial query Semantic Results

According to the categories, produced using the ontology based information of the semantic results, we further match the ODP categories which best suit them. Successively, we return the semantically enriched results grouped and sorted by the factor of similarity. We use a list with a view to storing the groups of categories which suit better in the categories of question that we made in the ontology.

6. Conclusions

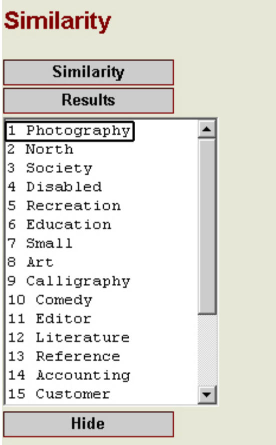
Overall, this work presents a novel approach that provides effective personalization of results produced by semantic questions. After implementing a lexical analysis of the ontology, it categorizes the results of questions into well-known and wide used ODP categories, which are used as a basis to categorize results by the Web's largest and most popular search engines and portals, including Netscape Search, AOL Search, Google, Lycos, HotBot, DirectHit, and hundreds of others.

The main objective of this work is demonstrate that semantic querying can be further augmented using ontology meanings analysis and personalization techniques. It aims o create a powerful solution that unifies a number of different operations for the personalization of semantic questions. Specifically, when we make a question in the ontology, the proposed solution returns the categories of ODP that suit [Yan et al (2002)] better with the categories of the results that we take from the question and categorizes them. A prototype has been implemented to present the effectiveness of the approach, which contrary to existing tools personalizes semantic query results integrated with the Protégé.



director
manager
Accounting
administrator
decision maker
Comedy
committee member
editor
Society
skilled worker
Customer
Art
Education
application
Reference

Figure 5. *Synonyms and hyponyms*



1 Photography
2 North
3 Society
4 Disabled
5 Recreation
6 Education
7 Small
8 Art
9 Calligraphy
10 Comedy
11 Editor
12 Literature
13 Reference
14 Accounting
15 Customer

Figure 6. *Top-k categories that match the Semantic Results*

On account of the above that mentioned we also implemented submission of online questions in a semantic structure – ontology. Experiments have shown that resulting categorization is improved when we expand the initial results' categories (classes etc) with their synonymous words (and their hyponyms). In this way, results are satisfactory even when categories are few. We compute the factors of similarity between the categories of instances (expanded), and the categories of ODP (outperforming the simple keyword matching). The similarity is performed based on WordNet and it computes the distance of the words in order to find the factors of similarity. We categorize the results providing in this way personalized views based on well-known categories wide used extending the work of [Dumais et al (2001)] into the semantic web world.

Future research steps include:

- Further experimentation of our approach using comparing the personalization effect when queries are submitted to different domains of knowledge by a single subject using time obsolescence to measure changes in user interests.
- The combination of semantically enriched questions in natural language form.
- Experimental evaluation of our method in free tagged information of Web 2.0 in order to assess the personalization capabilities.

7. References

Adamopoulou, P., Kanellopoulos, D., Sakkopoulos, E., and Tsakalidis, A.(2005), *Semantic learning interventions using web services technology*, in the proceedings of IASTED 2005 International Conference Web Based Education, (WBE 2005), Web Based Teaching and Learning Technologies track, 2/2005, pp. 528-533

Berners-Lee, T., Hendler, J., and Lassila, O. (2001), *The Semantic Web*, Scientific American, Vol. 285, No. 5, pp. 34-43.

Brusilovsky, P. (1998), *Adaptive educational systems on the world-wide-web: A review of available technologies*. In 4th International Conference in Intelligent Tutoring Systems, San Antonio, TX.

Cooley, R. (2003), *The Use of Web Structure and Content to Identify Subjectively Interesting Web Usage Patterns*, ACM Transactions on Internet Technology, Vol. 3, No. 2., pp. 93–116.

Dogac, A., Laleci, G., Kabak, Y., and Cingil, I. (2002), *Exploiting Web Services Semantics: Taxonomies vs. Ontologies*, IEEE Data Engineering Bulletin, Vol. 25, No. 4.

Dumais, S.T., Cutrell, E., and Chen, H. (2001), *Bringing order to the web: Optimizing search by showing results in context*. In Proceedings of CHI'01, Human Factors in Computing Systems, April 2001, pp. 277-283.

Garofalakis, J., Matsoukas T., Panagis Y., Sakkopoulos E., and Tsakalidis A. (2005), *Personalization Techniques for Web Search Results Categorization*, in the Proceedings of the 2005 IEEE International Conference on e-Technology, e-Commerce and e-Service (IEEE EEE 2005), 29 March - 1 April 2005 in Hong Kong, China, pp. 148-151

Garofalakis, J., Sakkopoulos, E., Sirmakessis, S., and Tsakalidis, A. (2002), *Integrating Adaptive Techniques into Virtual University Learning Environment*,in the proceedings of the IEEE International Conference on Advanced Learning Technologies, September 9- 12, 2002, Kazan Tatarstan, Russia, pp. 28-33.

Makris, Ch., Panagis, Y., Sakkopoulos, E., and Tsakalidis, A. (2006), *Category ranking for personalized search*, in the Data and Knowledge Engineering Journal (DKE), Elsevier Science, Vol. 60 , No. 1, pp. 109-125.

Sakkopoulos, E., Kanellopoulos, D., and Tsakalidis, A. (2006), *Semantic mining and web service discovery techniques for media resources management*, in Int. J. Metadata, Semantics and Ontologies, Vol. 1, No. 1, pp.66–75.

Yan, X., and Han, J. (2002) : *gSpan: Graph-based substructure pattern mining*. In Proc. of Int. Conf. on Data Mining (ICDM'02). Maebashi (2002) 721-724

Wu, Z., and Palmer, M. (1994), *Verb Semantics and Lexical Selection*, presented at 32nd Annual Meeting of the Associations for Computational Linguistics, Las Cruses, New Mexico, 1994.

Dieter, F., Hendler, J., Lieberman, H., and Wahlster, W. (2003), *Spinning the Semantic Web*.