DOI: http://dx.doi.org/10.1109/ICSC.2008.52

Semantic Search meets the Web

Miriam Fernandez[1], Vanessa Lopez[2], Marta Sabou[2], Victoria Uren[2],
David Vallet[1], Enrico Motta[2], Pablo Castells[1]

[1] *Escuela Politecnica Superior*
Universidad Autonoma de Madrid
C/ Francisco Tomas y Valiente 11, 28048 Madrid, Spain
{miriam.fernandez, david.vallet, pablo.castells}@uam.es

[2] *Knowledge Media Institute*
The Open University
Walton Hall, Milton Keynes, MK7 6AA, UK
{v.lopez, e.motta, r.m.sabou, v.s.uren}@open.ac.uk

Abstract

While semantic search technologies have been proven to work well in specific domains, they still have to confront two main challenges to scale up to the Web in its entirety. In this work we address this issue with a novel semantic search system that a) provides the user with the capability to query Semantic Web information using natural language, by means of an ontology-based Question Answering (QA) system [14] and b) complements the specific answers retrieved during the QA process with a ranked list of documents from the Web [3]. Our results show that ontology-based semantic search capabilities can be used to complement and enhance keyword search technologies.

1. Introduction

Semantically-aware search engines, and in particular those that use ontologies as enabling technologies, have gained considerable interest in the last few years but the actual fulfillment of the vision is still unclear. While ontology-based semantic search systems have been shown to perform well in organizational semantic intranets [13,17], there have not yet been convincing attempts at applying semantic search to the web as a whole. However, the ever growing amount of ontology-based semantic markup in the *Semantic Web (SW)* [1], provides an opportunity to start working towards a new generation of open intelligent applications [18].

We observe a number of weaknesses in existing semantic systems. The first is that they are restricted to a limited set of domains. The domain restriction may be identified by the use of just one specific domain ontology at a time [2,20,19,15], the use of a set of a priori defined ontologies covering one specific domain [11], or the use of one large ontology which covers a limited set of domains [13]. As a result, these approaches do not scale to open environments such as the Web, or heterogeneous document repositories, where an unlimited set of topics must be covered to successfully retrieve the information.

Secondly, they do not provide a semantic *document* ranking model: Semantic Portals [4,17] and QA systems [14,19] typically provide search functionalities that may be better characterised as semantic data retrieval, rather than semantic information retrieval. Searches return ontology instances rather than documents, and generally, no ranking method is provided. In some systems, links to documents that reference the instances are added to the interface [4], but neither the instances, nor the documents are ranked. While these solutions may be sufficient for small knowledge bases, they do not scale properly to massive document repositories, where searches typically return hundreds or thousands of results, and therefore they do not perform well if the retrieval space is large.

The work reported here is part of a larger effort in our labs to experiment with *open semantic search systems* that are not constrained by specific organizational ontologies, as is often the case today, but can exploit the combination, and scale, of information spaces provided by the Semantic Web and by the (non-semantic) World-Wide-Web. Specifically, we report experiments with a new system which builds upon two pre-existing semantic search approaches with complementary affordances. The first, drawn from the PowerAqua system [14], can search heterogeneous knowledge bases and generates answers to natural language queries. The second system, (reported in[3]), supports semantic search, based on formal domain knowledge, over non-semantic World Wide Web documents. The semantic search system exploits ontology-driven knowledge bases, and complements PowerAqua in two ways. If relevant ontologies exist and PowerAqua can provide an answer, it provides documentary evidence to help the user judge the validity of the answer. Alternatively, if PowerAqua cannot provide an answer, for example, because semantic markup is not available for the topic, the user still gets some answer in the form of relevant documents. The most important features are:

- It uses both relevant semantic data drawn from the Semantic Web, where it is available, and information found in standard web pages, to answer user queries.
- It provides an innovative and flexible solution to the problem of integrating data found in these two sources by dynamically creating links between web pages and semantic markup.
- It degrades gracefully when semantic markup is not available or incomplete.

Therefore, we propose this approach as a practical way to exploit the growing amount of semantic markup that is available on the Semantic Web, and to potentially enhance current search technology on the World Wide Web. The research presented here thus aims to make a step towards the design of semantic retrieval technologies which scale up to the open Web by: a) bridging the gap between the users and Semantic Web data and b) bridging the gap between the Semantic Web data and unstructured, textual information available on the Web. PowerAqua addresses the first aim by providing a natural language interface onto heterogeneous semantic data. The work reported in [3] addresses the second aim by making ordinary web pages open to semantic search.

The rest of the paper is organised as follows. The proposed system architecture is described in section 2. Section 3 explains how the system makes it possible for users to get answers from the Semantic Web, while section 4 describes how Semantic Web data can be used to enhance the retrieval process of non-structured information. We report an evaluation of the system in section 5 in which we compare it to simple keyword search and a query expansion method. Conclusions are drawn in section 6.

2. System Architecture

Figure **1** depicts the two main steps of the overall retrieval process. The first step aims to bridge the gap between the user and the SW by retrieving answers from a set of ontologies as a reply to a NL query. The second step aims to bridge the gap between the SW data and the unstructured information available on the Web by augmenting the answer with relevant Web documents.

Step1: Understanding the NL user request and retrieving an answer in the form of pieces of ontological knowledge. The user's NL query is processed by the ontology-based QA component, PowerAqua [14]. This component operates in a multi-ontology scenario where it translates the user terminology into the terminology of the available ontologies and retrieves a list of ontological entities as a response. To ensure fast access to the (online) available ontologies, these are indexed a priori. This addresses the issue of semantic search being restricted to a few predefined domains; the response is obtained from a potentially unlimited number of ontologies which cover an unrestricted set of domains. E.g, given the query *"which are the members of the rock group Nirvana?"* and two ontologies covering the term "Nirvana" (one about spiritual stages and one about musicians), PowerAqua is able to: 1) select these two ontologies containing the term Nirvana; 2) choose the appropriate ontology after disambiguating the query using its context and the available semantic information and; 3) extract from this ontology an answer in the form of ontological entities. In this case it returns a set of individuals, i.e., Kurt Cobain, etc.

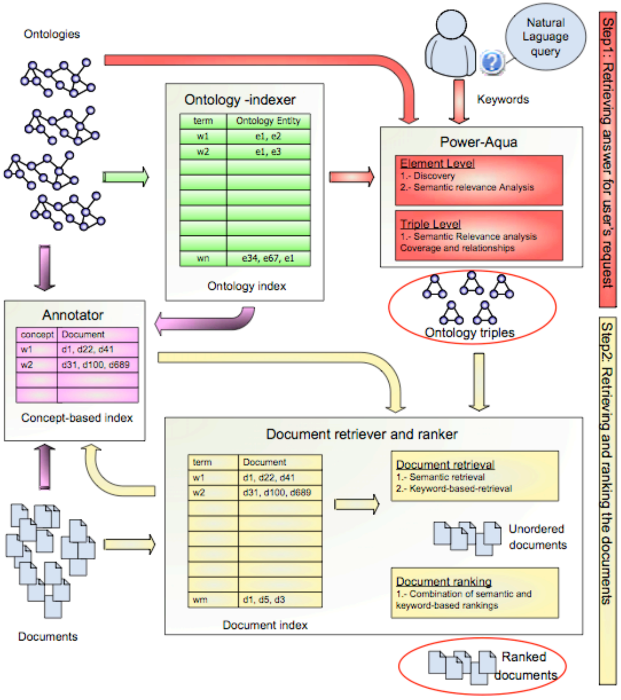

Figure 1: System architecture

Step2: Retrieving and ranking relevant documents based on the previously retrieved pieces of ontological knowledge: Once the exact answer to the user's query has been retrieved, the system performs a second step to retrieve Web documents. This phase addresses the two limitations described in Section 1 because a) documents are retrieved without any domain restriction, and b) they are automatically indexed in terms of the ontology concepts, retrieved and ranked using a semantic model that could potentially scale up to large document repositories.

Both steps are carried out using four main architectural components: (1) the ontology indexing module, which pre-processes (online) available semantic information; (2) the PowerAqua module, which answers the NL query in the form of ontology triples; (3) the annotator module, which generates a concept-based index between the semantic entities and documents; and (4) the document retrieval and ranking module, which retrieves and ranks documents relevant to the ontology triples obtained by PowerAqua. The output of the system consists of a set of ontology elements that answer the user's question and a complementary ranked list of relevant documents.

3. Bridging the gap between the Semantic Web and the User

The first step of our system relies on PowerAqua to exploit large-scale semantic data. Unlike its predecessor, AquaLog [15], which derived an answer from a single ontology, PowerAqua performs QA on multiple ontologies. As such it is part of a new generation of SW tools which dynamically reuse and combine information drawn from heterogeneous ontologies [18].

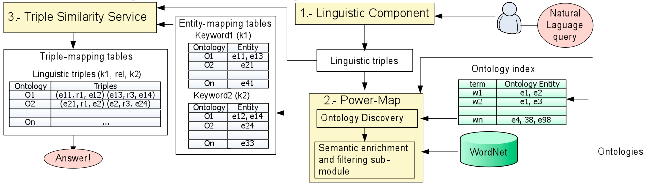

Figure 2: PowerAqua components in detail.

PowerAqua consists of three main components as shown in Figure 2. First, its linguistic component (detailed in [15]) uses GATE [5] to translate a NL query into its linguistic triple by identifying triple associations that relate terms together. For instance, our example query is translated to *<what-is*, *members, rock group nirvana>*. Second, PowerMap [14], maps the terms of each linguistic triple to semantically relevant ontology entities (Section 3.1). Finally, the triple similarity service (Section 3.2) selects the ontological triples that best represent the user's query. An answer is then generated from these triples (e.g., as a list of instances).

3.1. The PowerMap algorithm. PowerMap is a knowledge-based mapping algorithm that maps linguistic terms into entities drawn from different ontologies. The output of PowerMap is a set of *Entity Mapping Tables* each corresponding to one linguistic term. Each table contains the ontology elements (from different ontologies) to which the term was matched (See Table 1).

The PowerMap Ontology Discovery sub-module identifies, at run time, the set of ontologies likely to provide the information requested by the user. To take advantage of the knowledge provided by the entire SW in real time, we are currently integrating our tool with the Watson SW Gateway [6]. However, such integration had not yet been completed at the time of carrying out the experiments reported in this paper, and therefore these rely on a large amount of ontologies and semantic data (about 2GB), which we gathered directly on the SW and indexed using Lucene[1]. The semantic entities are indexed based on the entity's local name and its rdfs:label property and, optionally, from any other ontology property. A second index level is also generated, which contains taxonomical information about each semantic entity. PowerAqua makes use of both levels of indexing to search for approximate syntactic entity matches in real time. To broaden the search space, and bridge the gap between the user and ontology terminology, it uses not just the terms of the linguistic triple but also lexically related words obtained from WordNet. Moreover, it initiates a spreading activation search across ontologies (i.e., synonyms are found through owl:sameAs). In our example, the compound "rock group nirvana" does not produce any mappings as such, unless it is split into its parts. The linguistic triple is then split into: *<which-is, members, nirvana>, <rock, ?, nirvana>* and *<group, ? nirvana>*

Table 1. Entity Mapping Tables example for rock

ATO[2]	foo:bar#Rock [class, exact, {Synset#1: rock, stone – material …}]
MUSIC	http://www.uam.es/music.owl#rock [inst,Synset#2: rock_n_roll,]
NALT[3]	http://agclass.nal.usda.gov/nalt/2006.xml#rock_gardens [class, Synset#3: garden of rocks]

Once, the set of possible syntactic mappings have been identified, the **PowerMap semantic enrichment and filtering sub-module** determines the sense of the mapped entities and, when enough information is available, discards those that are semantically inappropriate. The semantic similarity between the triple terms and the concepts from distinct ontologies is computed by taking into account their meaning, as given by their place in the hierarchy of the ontology, through the use of a WN-based methodology evaluated in [10]. Therefore, it determines that mapping for "rock" has the synset *#material, stone* in ATO (parent "substance"), and *#a genre of music* in the music ontology (parent "specific-genre"). The mapping "rock gardens" in the NALT ontology, with synset *#a garden featuring rocks* (parent "gardens"), is discarded as semantically valid match for "rock".

3.2. The Triple Similarity Service. The Triple Similarity Service is a completely novel module invoked after all linguistic terminology has been meaningfully mapped at the element level. From these individual mappings spread over several ontologies (the *Entity Mapping Tables*), ontology relations are analyzed and the ontology compliant triples that link those mappings are created. This step will return a small set of ontologies that jointly cover the user's query. The output is represented as *Triple Mapping Tables* that relate each linguistic triple to all the equivalent ontological triples obtained from different ontologies. Finally, all the ontology triples that are related to actual instances, and can be used to generate an answer, are selected, giving priority to the ontologies that contain the most matches to the individual terms of a linguistic triple.

Table 2. Triple Mapping Tables example

Show me all cities in usa : <cities, ?, usa>	
SWETO	<city, attribute_country, usa >
UTexas	<city , isCityOf , State> <State, isStateOf , usa>

[1] http://lucene.apache.org/

[2] http://reliant.teknowledge.com/DAML/ATO_Ontology.owl

[3] http://www.nal.usda.gov/fnic/foodcomp/Data

The Relation Similarity Service (RSS) inspects and identifies the relations between the entity mappings covered by each selected ontology, in such a way that these relations are appropriate translations of the linguistic triples. As a result, a linguistic triple can be mapped into one or more ontology triples, and those may represent complete alternative translations of the linguistic triple, or partial translations to be combined. We briefly illustrate the RSS through the query "Show me all cities in USA" (Table 2), where no ontologies that contain a match for the relation can be found. This is a rather typical case, either because the linguistic relation is implicit, as in this example, or because the ontology relation has a label that is difficult to detect by syntactic techniques. Therefore the problem becomes one of finding ad-hoc or IS-A relations that link the two terms, i.e. *<city, has-attribute-country, USA>*. If no ad-hoc relations are found then IS-A relations between the arguments are inspected. If such relations are not found either, then the algorithm investigates the existence of indirect relations, i.e.: *<city, isCityOf, state> <state, isStateOf, USA>*.

If unlike the previous example the algorithm identifies a set of valid candidate mapped relations for the candidate mapped arguments, then matching and joining of triples is controlled by the domain and range information of the relations. In many cases, this is all we need to complete the triple, e.g., *<what-is, members, nirvana>* is translated to the ontological triple *<musician, has-member, nirvana>*. The RSS translates then the query "which are the members of the rock group Nirvana?" into the ontological triples *<musicians, has-members, nirvana> <nirvana, has-genre, rock> <nirvana, is-a, group>*. Note that other mappings of nirvana, rock and group in other ontologies did not produce any relevant ontological triples.

4. Bridging the gap between the Semantic Web and the Web

The second step of our system relies on the work reported in [3], to provide the user with a semantic-based ranking of Web documents. This system has been significantly extended and enhanced in our present work to deal with a multi-ontology scenario. In particular, the annotation and weighting algorithms (section 4.1) are completely novel techniques, while the adaptation of the traditional keyword-based IR model used during ranking process (section 4.2) has evolved from our previous work by building the query input from the ontology entities retrieved after the QA phase.

4.1. Annotating documents. As introduced in Section 1, our research is aimed to the integration of data found on the SW with information available in standard web pages in order to enhance the performance of current Web search technologies. A key step in achieving this aim lies on linking the semantic space to the unstructured content space by means of the explicit annotation of documents by semantic metadata. In such dynamic and changing environments, annotation must be done in a flexible way. The solution we are exploring in this paper does not require hardwiring the links between web pages and semantic markup. On the contrary, these are created dynamically in such a way that the two information sources may remain decoupled.

To achieve these goals, we adapt traditional keyword-based IR techniques to the task of generating ontology based conceptual indexes. Just as keyword indexes are needed to enable real time query answering in current Web search engines, in our view semantic search engines should similarly rely on conceptual indexes to properly scale up to large document repositories. Similarly, just as traditional ranking algorithms are based on keyword weighting, our approach relies on measuring the relevance of each individual association between semantic concepts and web documents.

Given this assumption, the problem lies in how to achieve the annotation process and generate the conceptual indexes in an efficient and dynamic way that can potentially scale to the Web. The overall annotation process consists on the following steps:

1) Extract and process Semantic Web and Web information: Typically, Semantic data and Web documents are crawled to populate the index. In our evaluation the "Web" documents come from the TRECWT10G collection. For Web documents, we generate a standard keyword index using Lucene.

2) Extract the textual representation of semantic entities: For each ontology, we analyze its semantic entities in order to find the documents that are likely to be associated with them. Each semantic entity is usually defined by one or more textual representations in the ontology. To make the process ontology-independent we assume that such lexical variants are present in the ontology as multiple values of the local name or rdfs:label property of the entity. For example, an individual entity describing Apple company could be labeled as "Apple", "Apple Inc.", etc. Note that a single textual representation can be shared by different semantic entities, e.g., the textual representation "Apple" can be shared by the fruit and the company concepts.

3) Find the set of potential documents to annotate: The identified representations of a semantic entity ("Apple", "Apple inc", etc.) are then searched in the index using standard keyword-based search and ranking processes, in order to find the documents that can be potentially associated with them.

4) Extract the semantic context of the entity: Some of the found documents may contain an ambiguous meaning of the entity, e.g. we may find a document referencing the entity "Apple" as the fruit instead of the company name. To address disambiguation issues we exploit the ontological relations. We define the context of an entity as the set of entities directly linked in the ontology by explicit relations. Hence, for example, the context of the entity Apple may contain concepts such as Company, Technology, etc.

5) Find the semantically contextualized documents: The above set of semantically related entities is processed to extract their textual representations, which are then searched in the keyword index. The set of documents extracted after this step is taken to be the set of semantically contextualized documents.

6) Generate the final annotations: The intersection between the previous set of candidate documents and the set of semantically contextualized documents provides us not just the set of documents that are likely to contain the concept, but also the set of documents that are likely to contain the contextual meaning of the concept. Once this set of documents is identified, a new annotation is created between the semantic entity and each of the documents.

7) Weight the annotations: In our system, annotations are assigned weights that reflect how well the semantic entities represent the meaning of the document. Weights are computed automatically from the ranking of documents previously extracted in steps 3 and 5. After step 3 we extracted a ranked list of documents for each textual representation of the semantic entity. The obtained lists of documents and their corresponding scores are used to create a final unique ranked list using the fusion methodology reported in [8]. This list, $L_{se,}$ contains a score for each document that can potentially generate a new annotation. The same is done after step 5, obtaining a list of documents for each contextual concept, L_{c1}, L_{c2},…,$L_{cn..}$ All these lists are fused again, obtaining the final list of semantically contextualized documents, L_c After filtering some documents, as discussed in step 6, the final list of documents and scores, L is computed as:

$$L = \alpha * L_{se} + \beta * L_c$$

Where α represents the relevance of the semantic entity, and β represents the relevance of its contextual meaning. These values have been obtained empirically. In the conducted experiments, we set $\alpha = 0.6$ and $\beta = 0.4$. The scores obtained in L are assigned as weights of the generated annotations.

In summary, our approach takes advantage of traditional IR models and techniques to dynamically link semantic data with documents. The approach provides a weighting schema on this base, and generates the conceptual-indexes that are used later in the semantic search and ranking processes.

4.2. Retrieving and ranking documents. The document-retrieval approach presented here is a slightly modification of [3]. In this work, the traditional keyword-based IR model is semantically adapted, replacing the traditional keyword query and document vectors by semantic query and document vectors. The query vector represents the importance of each semantic entity in the information need expressed by the user, while the document vector represents the relevance of each semantic entity within the document. To generate the query vector, the document retrieval system takes as input the pieces of ontological knowledge previously retrieved in response to the NL query during the QA phase. To generate the document vector it uses the annotations and weights computed during the annotation phase.

4.3. Dealing with semantic incompleteness. During the search process, it may happen that a) there is no available ontology that covers the query, or b) there is an ontology which covers the domain of the query but only contains parts of the answer. To cope with the problem of knowledge incompleteness, the semantic ranking is fused with a traditional keyword ranking. This allows the system to degrade 'gracefully' when no semantic information is available. The fusion of both ranked lists is made using a previously reported statistical approach [9].

5. Experiments

In contrast to traditional IR and QA communities, where evaluation using standardized techniques, such as those prescribed by the TREC annual competitions [12], has been common for decades, the SW community is still a long way from defining standard evaluation benchmarks that comprise all the required information to judge the quality of the current semantic search methods. Nonetheless, we wanted to test our system systematically and as rigorously as we could. To do so we had no choice but to build our own benchmark. We required a text collection, a set of queries and corresponding document judgments, ontologies that cover the query topics and knowledge bases that populate the ontologies, preferably using a source independent of the text collection.

The Document Collection and Queries: We decided to construct a benchmark taking the TREC 9 and TREC 2001 [12] test corpora as a starting point, because this provides us with an independently produced set of queries and document judgments. The IR collection we took as basis comprises 10 GB of Web documents known as the TREC WT10G collection, 100 queries, corresponding to real user logs requests, and the list of document judgments related to each query. These judgments allow the quality of the information retrieval techniques to be calculated using standard precision and recall metrics.

The Ontologies: As the SW is still sparse and incomplete [21], many of the query topics associated with WT10G are not yet covered by it. Indeed, we have only found ontologies covering around 20% of the query topics. In the remaining cases, ontology-based technologies cannot be used to enhance traditional search methodologies, and the system just relies on keyword-based search techniques to retrieve and rank web documents.

We have used 40 public ontologies, which potentially cover a subset of the TREC domains and queries. These ontologies are grouped in 370 files comprising 400MB of RDF, OWL and DAML. In addition to the 40 selected ontologies, our experiments also access another 100 repositories (2GB of RDF and OWL) stored in Sesame and indexed with PowerMap indexing structures (section 3).

The Knowledge Bases: Sparseness is an even bigger problem for knowledge bases than for ontologies. Current publicly available ontologies contain significant structural information in the form of classes and relations. However, most of these ontologies are barely populated. As a result the available knowledge bases are still not enough to perform large-scale semantic search testing. To overcome this limitation and provide a medium-scale test of our algorithms, some of the 40 selected ontologies have been semi-automatically populated from the independent information source: Wikipedia (Section 5.1). Wikipedia is a public encyclopedia comprising knowledge about a wide variety of topics. In this way, we endeavor to show how semantic information, which is publicly available on the Web, can be applied to enhance keyword search over unstructured documents.

5.1. Populating ontologies from Wikipedia.

Here we present a simple semi-automatic ontology-population mechanism that can be further improved with more sophisticated ontology population techniques, but this is out of the extent of our current research. The algorithm here comprises two main functionalities: 1) populating an ontology class with new individuals; e.g., populating the class Hurricane with individuals such as Katrina, Rita, etc., and 2) extracting ontology relations for a specific ontology individual, e.g., extract relations for the individual Tom Hanks, such as his set of films, etc.

The algorithm comprises 5 steps: **(1)** The user selects the class of individuals he/she wants to populate or expand with new relations. **(2)** The system extracts the textual form of this concept: either from the localName, or from the standard property rdf:label. **(3)** The system looks for the textual form of the concept in Wikipedia. **(4)** The Contents section or index of the Wikipedia entry is used to generate new classes and/or relations. Note that new classes and relations are created if we can not previously find a mapping in the ontology. **(5)** The sections which point to a table or a list are used to populate the ontology.

With this algorithm, we generated around 20000 triples distributed along the 40 pre-selected ontologies. As said before this new data added to the Knowledge Bases have not been extracted from the TREC documents, but from Wikipedia, which maintains the independence assumption for our experiments between the SW data and the unstructured information to be retrieved.

5.2. Adapting the TREC queries. The selection of the TREC queries was constrained in two ways: a) the queries must be able to be formulated in a way suitable for QA systems; this means queries like "discuss the financial aspects of retirement planning" (topic 514) can not be tackled; b) ontologies must be available for the domain to test our algorithms. The second point is a serious constraint. In the end, we considered 20 queries.

As we can see in Table 3, the original TREC queries are described by: a) a title, which is the original user query extracted from users' logs, b) a description, which can be considered the NL interpretation of the query, and c) the narrative, which explains in more detail the relevant information that the user is looking for. We added, for the queries we used: d) a detailed request, suitable for a QA approach, e) notes on available ontologies[4].

The final evaluation benchmark comprises: a) The TREC WT10G collection of documents; b) 20 queries and their corresponding judgments extracted from the TREC9 and TREC 2001 competitions; c) 40 public ontologies populated from Wikipedia covering the domains of the 20 selected queries and d) 2GB of extra publicly available ontologies.

Table 3. Example of TREC query

num	Number: 494
title	nirvana
desc	Find information on members of the rock group Nirvana.
narr	Descriptions of members' behavior at concerts and their performing style is relevant. Information on who wrote certain songs and biographical information is relevant

5.3. Experimental conclusions. The experiments were designed to compare the results obtained by three different search approaches with increasing levels of semantic awareness. The aims are two fold. On the one hand, we are able to evaluate the results retrieved by PowerAqua. On the other hand, we evaluate the advantage of semantically processing documents, rather that just using the semantic information to complement user queries.

Keyword search: This type of search is performed with the widely used text search engine Lucene.

Semantic query expansion: Semantic information is used just to expand the user query. PowerAqua processes the user query and extracts a list of semantic terms that, hopefully, improve the user's request. This list of semantic terms is then added to the original query and used to perform a traditional keyword search.

Semantic retrieval: The third search approach uses our complete semantic retrieval system, including the query processing performed by PowerAqua and the semantic document retrieval. For the experiments two keyword-based lists are fused with the semantic results, the

1.[4] The complete list of our selection of TREC is publicly available in: http://kmi.open.ac.uk/technologies/poweraqua/eval.html one obtained after the original keyword search and the one obtained after the semantic query expansion.

5.4. Results and discussion. Table 4 shows the results. The fist column contains the set of topics evaluated while the following columns contain the results for the three methodologies presented in 6.3 using two standard TREC metrics: average precision and P@10 (precision at 10).

It is not our goal in this paper to compare ourselves with the best TREC search engine. As shown in Section 4 our document retrieval and ranking algorithms depend on the quality of the index keyword search mechanism used during the annotation process and it is part of our future goal to use better keyword-based index tools.

Using as the baseline the Lucene index, the results presented show that both the semantic query expansion and the semantic retrieval approach can improve the keyword-based approach in the 65% of the evaluated queries. For example, for queries 457, 523 and 524 the semantic results can return valuable documents when the keyword-search does not return any relevant result. Another positive conclusion is that for 75% of the queries the quality of the first 10 results is better using the semantic information than the simple keyword ranking, which means that the semantic data can help to enhance precision.

For the queries that have not outperformed the keyword baseline, such as query 467, "Show me all information about dachshund dog breeders", a common reason is the scarceness of the semantic information obtained for the query but, in general, such queries perform no worse than the keyword baseline. The exception is query 489, "What is the effectiveness of calcium supplements". In this case, even though the semantic information retrieved is relevant and focused on the benefits of calcium like: bone_strength, muscle_mass, etc., the precision of the semantic search is worse than the keyword search. In the TREC evaluation it states "A relevant document must establish that the information comes from a qualified medical source". Since our algorithms only focus on content and do not analyze the linking structures between Web pages to evaluate the quality of the source they do less well than the baseline in this case.

To conclude, it is important to highlight that there is no qualitative improvement of the semantic retrieval over the semantic query expansion. We believe this is due to the fact that, even though we have achieved a flexible annotation process, the weighting algorithm is dependent on the quality of index keyword search mechanism. It is our aim and future work to evaluate the quality of the annotation process using different keyword index tools.

Table 4. Average Precision/P@10 metrics evaluation

Topic #	Lucene	Query expansion	Semantic retrieval
451	0.2850/0.5	**0.3970/0.7**	**0.4161/0.7**
452	0.0292/0.5	**0.0383**/0.2	**0.0383**/0.2
454	0.2569/0.8	0.0281/0.4	**0.2644**/0.8
457	0.0000/0.0	**0.0512/0.1**	**0.0486/0.1**
465	0.0017/0.0	**0.1414/0.2**	**0.1322/0.3**
467	0.1241/0.4	0.1225/**0.5**	0.0984/0.4
476	0.2820/0.3	0.1954/0.2	0.1265/**0.5**
484	0.1230/0.3	**0.1564**/0.2	**0.1916**/0.2
489	0.1078/0.3	0.0346/0.1	0.0881/0.2
491	0.0794/0.3	**0.4077/0.9**	0.0770/0.2
494	0.2158/0.8	0.1427/0.2	**0.4078/0.9**
504	0.0755/0.2	**0.1474/0.5**	**0.1349/0.2**
508	0.0345/0.1	**0.0732/0.4**	**0.1474/0.5**
511	0.1543/0.5	**0.3476/0.5**	0.0733/0.4
512	0.1165/0.2	0.0640/0.1	**0.2505/0.4**
513	0.0602/0.4	**0.0700**/0.0	**0.0786**/0.1
516	0.0323/0.0	**0.0755/0.4**	**0.0702/0.1**
523	0.0000/0.0	**0.2728/0.9**	**0.2860/0.9**
524	0.0000/0.0	**0.1853/0.4**	**0.1081/0.2**
526	0.0596/0.0	**0.1680/0.6**	**0.0863/0.1**

6. Conclusions

We have constructed a complete semantic search approach that covers the entire IR process, from an NL query to a ranked set of documents, by exploiting the complementary affordances of two existing systems. PowerAqua's ability to answer NL queries makes the user interface of our system more attractive than that of several search prototypes which rely on more complex ways to specify an information need (e.g., SPARQL queries). Also, this system can retrieve a concrete answer when the appropriate semantic data is available. The document ranking module of our system complements PowerAqua in two ways. First, it provides a list of semantically ranked documents in addition to the concrete answer that is retrieved. Second, if no answer is found by PowerAqua, then this module ensures that the system degrades gracefully to behave as an IR system. Indeed, we are not aware of any system that provides these functionalities.

Our experiments prove the feasibility of applying ontology-based retrieval models in unrestricted environments where an unlimited set of domains are covered. The initial results of the comparative evaluation are promising showing that, when enough semantic information is available, the precision, and the average performance of the proposed semantic search techniques enhances and, only does worse than keyword search in very rare cases.

Two interesting characteristic of our system are a) its semantic ranking model based on a flexible annotation model which keeps the two information spaces decoupled and b) our evaluation work that aims to be a contribution on its own as well, towards the formalization of evaluation methodologies and datasets for ontology-based retrieval, drawing from the IR tradition and standard resources.

Several issues remain nonetheless open. One of the distinctive features of our system is its openness to the number of topic domains. Indeed, unlike existing systems that are limited to a small set of domains by relying on a few pre-selected ontologies, our system can potentially cover an unlimited set of domains by making use of ontologies provided in a Semantic Web scenario. Our experimental evaluation has shown however that the potential of our system is overshadowed by the sparseness of the knowledge currently available on the Semantic Web. Indeed, we found that only 20% of the TREC topics were covered to some extent by online ontologies. Further, most of the relevant ontologies were only weakly populated with instance data. While this status of the Semantic Web caused a suboptimal behaviour for our system, any extension of the critical mass in ontologies and semantic data available online will result in a direct performance improvement of the proposed approach.

Acknowledgments

This work was funded by the Spanish Ministry of Science and Education (TIN2005-068 and FPU program) and OpenKnowledge (FP6-027253) projects.

7. References

1.Berners-Lee, T.; Hendler, J.; Lassila, O. 2001. “The Semantic Web”. Scientific American, May 2001

2.Bernstein, A., Kaufmann, E. (2006) GINO - A Guided Input Natural Language Ontology Editor. In Proc of the International Semantic Web Conference: 144-157

3.Castells, P., Fernández, M., and Vallet, D. An Adaptation of the Vector-Space Model for Ontology-Based Information Retrieval. IEEE Transactions on Knowledge and Data Engineering 19(2), February 2007, pp. 261-272.

4.Contreras, J., et al.: A Semantic Portal for the International Affairs Sector. In: Proceedings EKAW 2004.

5.Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V.: GATE: A Framework and Graphical Development Environment. In Proc of the 40th Meeting of the Association for Computational Linguistics (ACL'02). Philadelphia (2002).

6.D'Aquin, M., Baldassarre, C., Gridinoc, L., Angeletou, S., Sabou, M., Motta, E., (2007) Watson: A Gateway for Next Generation Semantic Web Applications. Poster, ISWC 2007.

7.Davies, J. Weeks, R. QuizRDF: search technology for the Semantic Web. System Sciences, 2004. In Proc of the 37th Annual Hawaii International Conference. Jan 2004.

8.Fernández, M., Vallet, D., Castells, P.: Using historical data to enhance rank aggregation. SIGIR '06: In Proc of the 29th annual international ACM SIGIR.

9.Finin T., Mayfield J., Fink C., Joshi A.: and R. S. Cost. Information retrieval and the Semantic Web. In Proc of the 38th Annual HICSS 2005.

10. Gracia, J., Lopez, V., d’Aquin, M., Sabou, M., Motta, E., Mena, E. (2007). Solving Semantic Ambiguity to Improve Semantic Web based Ontology Matching. In Proc. of the Ontology Matching Workshop at ISWC/ASWC’07.

11. Guha, R. V., McCool, R., and Miller, E.: Semantic search. In Proc. of the 12th International World Wide Web Conference (WWW 2003), Budapest, Hungary (2003) 700-709.

12. http://trec.nist.gov/

13. Kiryakov, A., Popov, B., Terziev, I., Manov, and D., Ognyanoff, D.: Semantic Annotation, Indexing, and Retrieval. Journal of Web Sematics 2, Issue 1, Elsevier (2004) 49-79

14. Lopez, V., Sabou, M. and Motta, E. (2006) PowerMap: Mapping the Real Semantic Web on the Fly, International Semantic Web Conference., Georgia, Atlanta.

15. Lopez, V., Motta, E., Uren, V. and Pasin, M. (2007) AquaLog: An ontology-driven Question Answering System for organizational semantic intranets. Journal of Web Semantics, Volume 5, Issue 2, June 2007.

16. Madala, R., Takenobu, T., Hozumi, T.: The use of WordNet in information Retrieval. Conference on the Use of WordNet in Natural Language Processing Systems. Montreal, Canada.

17. Maedche, A., Staab, S., Stojanovic, N., Studer, R., and Sure, Y.: SEmantic portAL: The SEAL Approach. In: Fensel, D., Hendler, J. A., Lieberman, H., Wahlster, W. (eds.): Spinning the Semantic Web. MIT Press, (2003) 317-359.

18. Motta, E., Sabou, M. Next Generation Semantic Web Applications. in 1st Asian Semantic Web Conference. 2006. Beijing.

19. Cimiano, P., Haase, P., Heizmann, J. (2007) Porting Natural Language Interfaces between Domains -- An Experimental User Study with the ORAKEL System. In Proc of the International Conference on Intelligent User Interfaces.

20. Rocha, C., Schwabe, D., and de Aragão, M. P.: A Hybrid Approach for Searching in the Semantic Web. In Proc. of the 13th International World Wide Web Conference 2004.

21. Sabou, M., Gracia, J, Angeletou, S., d'Aquin, M., Motta, E., (2007) Evaluating the Semantic Web: A Task-based Approach. The 6th International Semantic Web Conference, Korea.

22. Salton, G., and McGill, M: Introduction to Modern Information Retrieval. McGraw-Hill, New York (1983).