PowerAqua: A Multi-Ontology Based Question Answering System – v1

OpenKnowledge Deliverable D8.4

Vanessa Lopez, Enrico Motta, Marta Sabou, Miriam Fernandez,

Knowledge Media Institute, The Open University.
{v.lopez, e.motta, r.m.sabou}@open.ac.uk
miriam.fernandez@uam.es

Abstract. In this report, we present PowerAqua, a multi-ontology-based Question Answering (QA) system, which takes as input queries expressed in natural language and is able to return answers drawn from relevant distributed resources on the Semantic Web. In contrast with any other existing natural language front end, PowerAqua is not restricted to a single ontology and therefore provides the first comprehensive attempt at supporting open domain QA on the Semantic Web.

1. Introduction

The development of a semantic layer on top of web contents and services, the Semantic Web (SW) [12], has been recognized as the next step in the evolution of the World Wide Web from its original characterization as a web of documents to a large scale web of formally characterized data.

The emergence of such large scale semantics has re-ignited interest in natural language front ends, which make it possible for users to formulate precise queries using their own terminology. In particular, consistently with the crucial role played by ontologies in structuring semantic information on the web, a new trend in the Question Answering (QA) area has emerged, *ontology based QA,* which directly exploits the power of ontologies during query analysis and translation (Aqualog [7], ORAKEL [4], GINO [2]). However, all these systems are essentially designed to support QA in corporate databases or *semantic intranets*, where a shared organizational ontology is typically used to annotate resources in an organizational intranet. Thus, all of these systems can only make use of a single ontology at a time, and moreover they require either the user to provide domain specific grammars (ORAKEL), or the use of guided user interfaces (GINO), which generate a dynamic grammar rule for every ontology element, or they ask the user for assistance every time an ambiguity arises (AquaLog). In a nutshell all of them require a degree of customization or interactivity, which may make sense in a closed domain, but it is not suitable to support QA in the open domain of the semantic web, which is already characterized by thousands of ontologies and millions of documents and is expected to grow by at least another order of magnitude in the next 3-5 years.

To address the aforementioned gap, we have developed a novel system, PowerAqua [6], which extends the capabilities provided by AquaLog, to support QA in the open domain of the Semantic Web. Specifically, PowerAqua takes as input a question expressed in natural language and returns all the answers to the question that can be found anywhere on the Semantic Web.

2. Retrieving answers in open multi-ontology environments

2.1 Issues in open domain QA

PowerAqua is able to use the available information on the SW to produce an answer to a query, where the system has to deal not only with the heterogeneity introduced by the ontologies themselves but, in addition, it cannot assume that the relevant ontologies actually refer to the same domain. In general, ontologies can have overlapping or disjoint domains and can use similar or completely different terminologies.

As highlighted in [6], several new challenges have to be solved in the open domain of the SW, in order to interpret a query by means of different ontologies.

- First of all, in a heterogeneous, open domain scenario it is not possible to determine in advance which ontologies will be relevant to a particular query. Hence it is crucial to have efficient and intelligent techniques for real time ontology selection and ranking.
- Secondly, user terminology has to be translated into several ontology-centric terminologies, as several ontologies may in principle provide alternative answers, or parts of a composite answer. Here, mapping and Word Sense Disambiguation techniques have to be applied to avoid potentially incoherent constructions (e.g., "a conference chair with four legs") and ensure that the concepts that are shared by statements derived from different ontologies (e.g., "conference chair" and "chair") have the same sense.
- Finally, the answer to a query may require the integration of information from multiple sources. Among other things, this requires the ability to recognize that individuals drawn from different sources may actually refer to the same entity (co-reference of instances).

Hence, the major challenge in developing PowerAqua arises from the combined issues of heterogeneity and scale characterizing the Semantic Web, which require new, efficient solutions for ontology selection, mapping and for word sense disambiguation (WSD), which must be applicable to real time query answering.

2.2 Limitations of existing approaches to ontology mapping, selection and WSD

While a lot of research has traditionally been carried out in the areas of ontology mapping and selection and WSD, the scenario of interest to us is very different from the traditional applications in which these techniques have been applied and therefore we found that existing approaches do not necessarily provide the level of support required by PowerAqua.

For instance, according to a study presented in [11], most ontology search systems use a set of ontology structure based metrics (compactness, richness, coverage) but don't look for synonymic information and cannot find ontologies where relevant concepts exhibit a syntactical dissimilar structure.

Existing ontology and schema based matchers have been primarily designed for design time alignment of ontologies known to cover the same domain, and therefore, they tend to perform badly when there is little overlap between the labels of the ontology entities, or when the ontologies have weak or dissimilar structures. Other approaches require axiomatized domain ontologies as background knowledge, and therefore do not work well in open domains.

Finally the general approach to designing WSD techniques[1] (see [5] for a state of the art) has been to map the local terms of distinct ontologies into a single shared ontology, and then semantic similarity is determined as a function of the path distance between terms in the hierarchy of the single ontology. However, further work is needed to extrapolate these techniques for cross-ontology comparisons.

In sum, tools like PowerAqua require solutions that can tackle the heterogeneity and large scale characterizing the online available semantic data, while at the same time proving themselves suitable for use at run time. In [8] we presented some of the requirements that have to be addressed by such novel techniques and we described PowerMap [8], a knowledge-based matcher that, unlike traditional mapping algorithms, is focused towards dealing with several, heterogeneous ontologies, which are not given a priori, but rather discovered depending on the content of the user's query. The novelty of PowerMap is that the mapping process is driven by the task that has to be performed, more concretely by the query that is asked by the user. Indeed, this is novel in comparison with traditional approaches where mappings are done prior to the ontology being used for a specific task. Furthermore, in contrast with traditional mapping approaches, PowerMap does not assume that the ontologies to be matched will be similar in complexity or structure and describe more or less the same domain, given that such similarity assumptions do not apply in our scenario (open QA on the SW). Indeed, PowerMap is able to reason about ontologies, which may only have very few concepts in common and describe different domains.

3. PowerAqua architecture

The overall QA processing is illustrated in Figure1. In a first step, the linguistic component analyzes the NL query and translates it into its linguistic triple form. E.g a query "What are the cities of Spain?" has the linguistic triple (<*what-is, cities, Spain*>). The role of the Query-Triples is simply to provide an easy way to manipulate the input. The AquaLog linguistic component [7] is appropriate for the linguistic analysis thanks to its ontology independent nature, and therefore, it is reused for PowerAqua. The notation used to represent a triple is: <*term, relation, term* >.

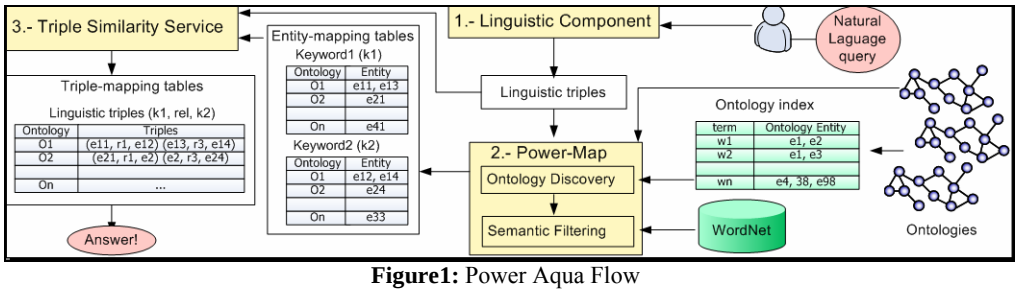


Figure1: Power Aqua Flow

In a second step the Ontology Discovery sub module of PowerMap [8], identifies the set of ontologies likely to provide the information requested by the user. To do so, it searches for approximate syntactic matches within the ontology indexes, using not just the linguistic triple terms, but also lexically related words obtained from WordNet and from the ontologies, used as background knowledge sources. For instance, the term *cities* match with the concepts *city, metropolis, etc*. Once the set of possible syntactic mappings has been identified, the PowerMap Semantic Filtering sub module checks its validity using a WordNet-based filtering methodology. For instance, the ontological concept *Game*

[1] WSD are classified in: (1) Ontology-based; (2) Information theory based, and (3) Vector space and string based

obtained as a synonym of the query term *Sport* will be discarded if its ontological parent is *HuntedAnimals.*

After this process, PowerMap generates a set of Entity Mapping Tables where each table links a query term with a set of concepts mapped in the different domain ontologies.

In a third step the Triple Similarity Service module takes as input the previously retrieved Entity Mapping Tables and the initial Linguistic triples and extract, by analyzing the ontology relationships, a small set of ontologies that jointly cover the user query. The output of this module is a set of Triple Mapping Tables, where each table relates a linguistic triple with all the equivalent ontological triples. Using these triples the information drawn from the relevant semantic sources is analysed to generate the final answer.

3.1 The PowerMap Algorithm

PowerMap is a hybrid knowledge-based matching algorithm comprising terminological and structural scheme matching techniques with the assistance of large scale ontological and lexical resources.

PowerMap as presented in [8] is the solution adopted by PowerAqua to translate user terminology into several ontology-compliant terminologies, while at the same time performing effectively in the run time scenario of open QA on the Semantic Web. In this section we provide more details on the PowerMap algorithm, describing the various phases of the algorithm. It is important to note that, in order to optimize performance, the complexity of these phases increases both with respect to the type of ontology entities that they consider and because of the techniques they use. Hence the most time-consuming techniques are executed last, when the search has been narrowed down to a smaller set of ontologies.

(1) Indexing Ontologies

We envision a scenario where a system may need to deal with millions of semantic documents structured according to hundreds of ontologies. To successfully manage such amounts of information in real time, the semantic sources are previously analyzed and stored into one or several inverted indexes using Lucene[2].

The semantic entities are indexed (different indexes for classes and properties, and for instances and literals) based on a mapping between each entity and a set of keywords extracted, by default, from their local name and their *rdfs:label* meta property. These mappings allow the generation of an inverted index where each keyword may be associated to several semantic entities from different ontologies. To search the semantic information stored in the indexes we make use of the advantages that Lucene provides for approximate searches combined with the capabilities of WordNet. A second index level is also generated with taxonomical information about each semantic entity using a database. PowerAqua makes use of both levels of indexing to increase the mapping speed of semantically sound entities, managing the distributed semantic information in real time.

(2) PowerMap: ontology selection and discovery

This phase is responsible for bridging the gap between user terminology and the multiple heterogeneous ontologies. This component identifies a set of ontologies that are likely to provide the information requested by the user's query. To broaden the search space and bridge the gap between the user and ontology terminology it uses approximate mappings and WordNet.

The novelty here is that, each mapped entity has associated metadata (consisting of its equivalent entities, i.e. *owl:sameAs*, and sub/superclasses) that can be used as a source of

[2] http://lucene.apache.org/java/docs/

information (relevant ontologies are used as *background knowledge sources*) to find new relevant entities with dissimilar labels. For instance, WordNet does not provide "academics" as a synonym for "researchers", but "AcademicStaff" can be found as a superclass (hypernym) of "researcher" in the Ka2 ontology. Moreover, this metadata is also used to select the most informative (up in the hierarchy) or exact (i.e. equivalent vs. hypernyms) mappings within the same ontological taxonomy.

(3) PowerMap: semantic relevance analysis and filtering at element level

A semantic mapping component that considers the content of an information item as its intended meaning is needed to: (1) help on the disambiguation to narrow down the search or right mappings based on the meaning; (2) to answer a query when the system needs to combine partial answers from more than one ontology. Therefore, a query term can be mapped to different ontologies, and as long as they provide similar interpretations of the query term, they are valid solutions. In other words, two concepts are semantically equivalent if their instance information can be correlated or merged.

This phase operates on the reduced set of ontologies identified in the previous phase by *syntax driven techniques* (SDT). Here, the goal is to verify the syntactic mappings identified previously and exclude those that do not make sense from a semantic perspective (e.g., the intended meaning of the query term differs from the intended meaning of the concept that was proposed as a candidate match). For example, if the term "capital" is matched to concepts with identical labels in a geographical ontology and a financial ontology, these two meanings are not semantically equivalent.

To check the semantic validity of the mappings, WordNet based methods are used to elicit the sense of a candidate concept by looking at the ontology hierarchy, and to compute the similarity between the query term and the concepts from distinct ontologies (see [8] for an example). Formally, semantic similarity is determined as a function of the path distance between the terms and of the extent to which they share information in common [9] in the IS_A hierarchy of WN, as given by the Wu and Palmer's formula [13]:

$$\text{Similarity}(t, c) = t \sim c = (2 \times \text{depth}(\text{C.P.I}(t, c))) \;/\; (\text{depth}(t, c) + 2 \times \text{depth}(\text{C.P.I}(t, c)))$$

Notation: the uppercase letters T, C,… denote terms (words) and lowercase letters t, c, synsets in WN, we write depth(t, c) for the path between t and c, and depth (CPI (t,c)) for the depth between the common parent of t and c and the root of the IS-A hierarchy. The maximum depth to be considered is 10. We also use > (and <) to express the hierarchical order relation in an ontology.

Let S_T and S_C be the synsets of a query term T and its mapped term C, respectively. We define the set of shared senses (synsets) of C with T to be:

$$S_{C,T} = \{c \in S_C \mid \exists t \in S_T \text{ such that } t \sim c\}$$

That is, $S_{C,T}$ is the set of those synsets c of C for which there exists a synset t of T such that t and c are semantically similar. If $S_{C,T}$ is empty, the mapping C is discarded because the intended meaning of the term T is not the same as that of the concept C. Finally, the true senses of C are determined by its place in the hierarchy of the ontology:

$$S^H_C = \{c \in S_C \mid \forall\ R\ ((R{>}C \vee R{<}C) \rightarrow (\exists r \in S_R\ (c \sim r)))\}\ .$$

That is, S^H_C consists only of those synsets of C that are similar to some synset of any of the ancestors and descendants of C in the ontology. We then intersect these senses, S^H_C, with the senses obtained in our previous step, $S_{C,T}$. Obviously, if this intersection is empty it means that the sense of the concept in the hierarchy is different from the sense that we thought it might have in the previous step, and therefore that mapping should be discarded

For instance, *bird* is mapped to *fowl_cholera* in UN FAO's AGROVOC[3], using syntactic techniques over its WordNet synonym *fowl*. However, after this semantic analysis and considering that its ontology parent is *bacteriosis* the mapping is not longer valid.

The drawback is that we fully rely on sense information provided by WordNet to compute semantic similarity, which in some cases may affect recall.

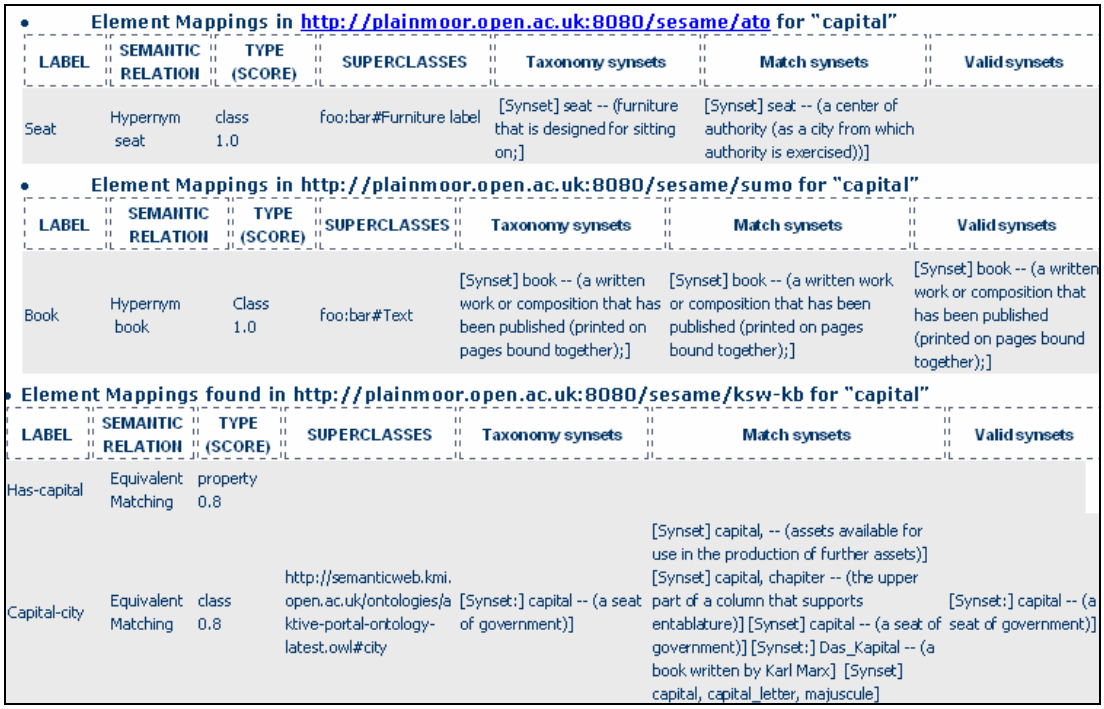
3.2 An illustrative example

Consider an ambiguous keyword T = "capital", which in WordNet has the senses represented in Table 1. Consider also the mappings, shown in Figure 2, for "capital" as classes in the ATO, SUMO and ksw-kb[4] ontology: C1 = "seat", C2 = "book" and C3 ="capital-city" respectively, where the first two mappings correspond to WordNet hypernyms. The possible synsets for the mapped terms when considering the query term are reduced to:

$S_{C1,\ T}$ = {Synset#c: seat -- (a center of authority (as a city from which authority is exercised))}

$S_{C2,\ T}$ = {Synset#e: book -- (a written work or composition that has been published)}

$S_{C3,\ T}$ = {Synset#all} Note that in principle they share all the synsets in common



Element Mappings in http://plainmoor.open.ac.uk:8080/sesame/ato for "capital"

LABEL	SEMANTIC RELATION	TYPE (SCORE)	SUPERCLASSES	Taxonomy synsets	Match synsets	Valid synsets
Seat	Hypernym seat	class 1.0	foo:bar#Furniture label	[Synset] seat -- (furniture that is designed for sitting on;]	[Synset] seat -- (a center of authority (as a city from which authority is exercised))]	

Element Mappings in http://plainmoor.open.ac.uk:8080/sesame/sumo for "capital"

LABEL	SEMANTIC RELATION	TYPE (SCORE)	SUPERCLASSES	Taxonomy synsets	Match synsets	Valid synsets
Book	Hypernym book	Class 1.0	foo:bar#Text	[Synset] book -- (a written work or composition that has been published (printed on pages bound together);]	[Synset] book -- (a written work or composition that has been published (printed on pages bound together);]	[Synset] book -- (a written work or composition that has been published (printed on pages bound together);]

Element Mappings found in http://plainmoor.open.ac.uk:8080/sesame/ksw-kb for "capital"

LABEL	SEMANTIC RELATION	TYPE (SCORE)	SUPERCLASSES	Taxonomy synsets	Match synsets	Valid synsets
Has-capital	Equivalent Matching	property 0.8				
Capital-city	Equivalent Matching	class 0.8	http://semanticweb.kmi.open.ac.uk/ontologies/aktive-portal-ontology-latest.owl#city	[Synset:] capital -- (a seat of government)]	[Synset] capital, -- (assets available for use in the production of further assets)] [Synset] capital, chapiter -- (the upper part of a column that supports entablature)] [Synset] capital -- (a seat of government)] [Synset:] Das_Kapital -- (a book written by Karl Marx] [Synset] capital, capital_letter, majuscule]	[Synset:] capital -- (a seat of government)]

Figure 2: Some of the element level mappings for the keyword "capital" and its synsets

[3] http://www.fao.org/agrovoc

[4] The AKT ontology populated with information about KMi

	City#1: large and densely populated urban area.., metropolis	**City#2:** an incorporated administrative district.	**City#3:** people living in large municipality
Capital#a (assests ..)	*Not an allowable path or depth is too long to be considered relevant*		
Capital#b (wealth ..)			
Capital#c (seat of government)	**Depth** = 8, **CPI** = region, **score**=0.42 **CPI_depth**= 3 (region, location, entity)	**Depth** = 7, **CPI** = region, **score**=0.46 **CPI_depth**= 3 (entity, location, region)	
Capital#d (capital letter)			
Capital#e (book by Karl Marx)			
Capital#f (upper part column)	**Depth** = 8**, CPI** = location, **score**=0.33 **CPI_depth** = 2 (entity, location)	**Depth** = 7, **CPI** = location, **score**=0.36 **CPI_depth** =2 (entity, location)	

Table 1. Similarity between "capital" and its ontology ancestor "city"

Then, the sense of the mapped term, in the context of the ontology it belongs to, is obtained by looking at its ontology ancestor. For instance, the results of computing similarity for the mapped term C3 "capital-city", whose lemma is "capital", when considering its ontology ancestor "city" in the "ksw-kb", are presented in Table 2 (please note that blank means that either there is not an allowable IS-A path between the senses or the depth is too long to considered relevant). Analyzing these results we can quickly select capital#c as the correct meaning in the ontology. Moreover, the hypernym of *capital#c* is *"seat#5"*, defined as "seat –centre of authority (*city* from which authority is exercised)". Note that the word "city" is used as part of its definition, another indication that capital#c is strongly related to "*city"*. The same semantic similarity is computed for C2 "book", whose superclass is "text", and the obtained ontological sense is the same as the previously obtained synset: Synset#e. However, for C1, "seat", whose superclass is "furniture", its meaning in the ontology class refers to S^H_C ={Synset: seat#d – furniture that is designed for sitting on}, so the intersection between its meaning in the ontology and its intended meaning on the mapping is empty ($S^H_C \cap S_{C1,T} = \emptyset$) and the mapping should be discarded.

During the next phases, a deeper analysis of the ontology semantics (relationships and triples) is needed to distinguish further between the two valid non-equivalent interpretations of "capital" in C2 and C3. The relevant techniques are described in the next section.

3.3 The Similarity Services

Having worked at the level of individual term mappings so far, the mappings produced by the previous phase are spread over several ontologies. The goal of this final phase is to identify out the meaningful mappings that better represent the query domain to create the ontology compliance triples equivalent to the user query by (a) determining those ontologies that cover entire triples and not just individual terms of the triples and by (b) studying the *ontology relatedness* to determine the valid semantic interpretation (e.g. to decide which ontological interpretation of "capital" is valid for the sense of the query term). In this phase we employ triple and relation centered similarity services to match between the predicates of the triples and the relations in the identified ontologies. This step will return a small set of ontologies that jointly cover all terms and hopefully contain enough information to generate the answer to the question.

The ***Triple Similarity Service*** is invoked after all linguistic terminology has been meaningfully mapped at the element level. It takes as input the linguistic triples and the *Entity Mapping Tables*, produced by the PowerMap element level techniques. The outputs are the *Triple Mapping Tables* that relate each linguistic triple to all the equivalent ontological triples obtained in different ontologies at the schema-level (see Fig. 3). From

them, all the ontology triples that make sense with respect to the relevant semantic sources, and therefore can be used to generate an answer, are selected.

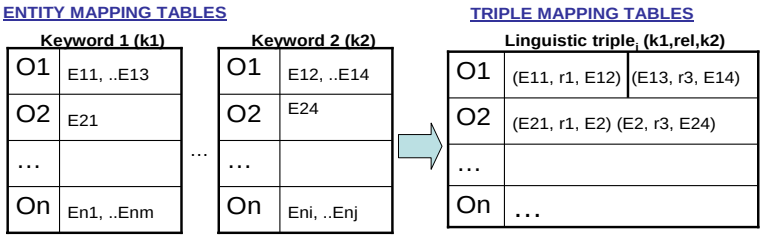


Figure 3. From element level mappings to triple level mappings

The *Triple Similarity Service* is responsible for creating the ontology compliant triples by a) linking the mapped ontology terms to create the *Onto Triples* and b) linking the triples between themselves. For the step a) to create the triples, a pair of ontology terms is linked by relationships within the same ontology to which the terms belong, through the ***Relation Similarity Services (RSS)***. For step b) while different triples may belong or not to different ontologies, if they are partial translations they have to be linked between them, by at least one common term, to create a complete translation. There is not a single strategy here; basically it depends on the type of query and ontology structure.

a) The Triple Similarity Service

Specifically, the process of creation of the *Triple Mapping Tables*, for a basic query, is as follows. First, the *Entity Mapping Tables* are obtained for each different term on the *Linguistic triples*. At this stage, the *Triple Similarity Service* can use the domain information to modify the *Linguistic triples* accordingly to represent compound terms that do not have any ontology mapping, until they are decomposed in different components that match ontology terms (within the same ontologies). Basically this involves the creation of a new *Linguistic triple* for each compound. For instance in a query like "List me all the Spanish researchers living in UK", whose *Linguistic triple* is of the form <*what-is, Spanish researchers, UK*>, the "Spanish researchers" term does not have any ontology mapping covering the whole compound, while there may be different mappings for the term "Spanish" and the term "researchers". In such a case the *Linguistic triple* will be modified to <*Spanish, ?, researcher*> <*researcher, living, UK*>. Naturally, in those cases where there is indeed a mapping for the compound in a particular ontology, then we do not go looking for separate mappings. While this approach may in some cases miss relevant mappings, it simplifies the mapping process and in most cases avoids the generation of noise. For instance, the term "Semantic Web" discomposed in "Semantic" and "Web" would produce many irrelevant mappings.

In the second stage, once the *Entity Mapping Tables* are obtained, the RSS is invoked for each linguistic triple and each ontology. As input, the RSS gets the mappings, within the same given ontology, for each term in the linguistic triple. As a result, the RSS obtains the set of *Onto Triples* for that given ontology. These results are used by the *Triple Similarity Service* to generate the *Triple Mapping Tables*.

During the third stage, using the category and features[5] of the triples, the *Triple Similarity Service* selects the set of *Onto Triples* that together best represent a satisfactory

[5] There are some lexical features which help in the translation or put a restriction on the answer, presented in both the linguistic triple and onto triples like if the relation is passive, or is an IS-A relation.

translation for each linguistic triple and the query as a whole. The filtering of the right *Onto Triples* on the *Triple Mapping Table* is based on:

- The similarity and meaning of the mappings that link the triples between themselves. Moreover, as we can have more than one satisfactory solution, different answers can be merged, i.e., by identifying common instances.
- The nature of the retrieved ontology relations (e.g., direct relations are preferred) and the level of coverage of the query (i.e., ontologies that cover the query completely are preferred).
- The nature of the taxonomic relationship in question. Specifically, *Onto Triples* formed with equivalent mappings are preferred, if possible, rather than *Onto Triples* created with mappings that are related to the query terms through a hypernym relationship.

Going back to the "capital" example, we have narrowed down to three valid non-equivalent ontology interpretations for the term "capital", in the linguistic triple: *<capital, ?, Spain>*, in three different ontologies: (1) geographical ontology (*capital#c)*; (2) financial ontology (*capital#a,b)*; (3) ontology about books ("book" as hypernym of *capital#e*). However, only ontologies (1) and (2) present also mappings for "Spain" (cover the whole triple), and therefore, following the coverage criterion, the book ontology is discarded. A deeper analysis of the ontology relationships will find that the direct relation "is-capital-of" connects any country, e.g. "Spain", with the class "capital" in the geographical ontology. However, in the financial ontology there is not a direct relation between countries and capital because there is a mediating concept that represents a company, that has a series of capital goods and is based in a country. We take this as an indication that the geographical ontology is more related to our query and should be selected to create the *Onto Triples*.

Finally, by definition, all the elements on an ontology compliance triple belong to the same ontology. If no ontology compliance triples can be found to map a linguistic triple and the elements in the linguistic triple map to different ontologies, then this indicates that the linguistic triple may need to be decomposed into more than one triple. For instance, consider the query "Which researchers play football?", where we can find an ontology about researchers and an ontology about footballers. In this case, the linguistic triple *<researchers, play, football>* should be restructured and translated into two triples solved by different ontologies: *<?, is-a, researcher>* and *<?, is-a, footballer>*.

b) The Relation Similarity Service (RSS)

Essentially, for each linguistic triple and ontology with relevant mappings for its terms, the RSS is called to try to make sense of the input linguistic triple and obtain all the possible ontology compliance triple combinations that represent it. By analyzing the taxonomy and the relationships between the mapped entities in a given ontology, a linguistic triple can be mapped to more than one *Onto Triple*, each one being a complete alternative translation of the linguistic triple, or to partial translations that combined together cover the whole linguistic triple.

We can distinguish three different generic cases the RSS has to deal with to create the *Onto Triples*:

First case: The simplest case is when an *Onto Triple* can be directly created from the mapped entities. For instance, given the *Linguistic triple <person, works, akt>* generated from a query such as "Who works in akt?", where "akt" is an instance of a "project", a given ontology can provide one or more valid or alternative solutions to be disambiguated: (a) *<person, works-for, project>;* (b) *<project, has-project-member, person>;* (c) *<project, has-project-leader, person>*.

Second case: A linguistic triple may need to be mapped to a combination of *Onto Triples* within the same ontology. For instance, the query "who has publications on iswc?", whose *Linguistic triple* is*: <person/organization, have publications, iswc>*, may be mapped into: *<person, have-publication, publication>* & *<publication, in-proceedings, iswc>*

Third case: For instance, the Wine Ontology, which has been used to illustrate the specification of the OWL W3C recommendation, has no direct relations between wines and food. Instead, a mediating concept "mealcourse" is used. Hence, to address a question like "which wines are recommended with cakes?" two *Onto Triples* should be generated to show the indirect relation: *<MealCourse, hasFood, food >* & *<MealCourse, hasDrink, PotableLiquid,>*

3.4 The RSS algorithm

The RSS finds candidate ontology compliance triples by looking for relations between two arguments at the schema level. If one of the arguments is an instance it looks for relations in which the class of the instance is the domain or range of the relation. These schema relations might or might not be instantiated for a particular instance but this information is used at a later stage to generate an answer or for selecting out the best ontological triples.

The input to the algorithm includes all the candidate entities and relation mappings for the terms in a linguistic triple in a given ontology. However, the set of candidate mappings for one term in the *Linguistic triple* may be empty (1) because of PowerMap element level techniques could not find any ontology mapping for that term; or (2) because of the type of query. The former case is very common when mapping relations, because ontology relations have complicated labels difficult to detect by purely syntactic techniques. The latter case can be seen for example in *what-is* queries where there is not information about the type of the *wh-query* term, or in queries where the relation is implicit and therefore there are not syntactic mappings for it, or when the type of the relation is "IS_A", i.e. "is Dali a painter?". If both candidate ontology mappings for both arguments are empty then the process is aborted and it will return an empty set of *Onto Triples* for that ontology.

The detailed description of the algorithm is as follows:

Case 1: If the set of candidate ontology entities for a linguistic relation is empty, either because the ontology relation has a label that is difficult to detect by syntactic techniques, or because the linguistic relation is implicit, the algorithm proceeds as follows.

(Step 1) whenever there are successful matches for both arguments, the problem becomes one of finding *ad-hoc* relations which link the *wh-query* term to the second term or any of its superclasses. Superclasses and subclasses are considered due to the inheritance of relations through the subsumption hierarchy. For instance, in "who works in akt?", the possible relations could be defined only for researchers, students or academics, rather than people in general. The ontology can provide one or more alternative solutions like (a) *<person, has-project-member, project>* and (b) *<person, has-project-leader, project>*.

(Step 2) If no ad-hoc relations are found, it looks for IS-A relations between both arguments (i.e. "AKT" IS-A "project").

(Step 3) If still no ontological relations are found, it looks for indirect relations through one mediating concept between both arguments. For instance in "which wines are recommended with cakes?" two *Onto-Triples* need to be generated to match the ontology: *<MealCourse, hasFood, food >* & *<MealCourse, hasDrink, PotableLiquid,>*

Case 2: If we have a set of candidate relations, the procedure to find *Onto Triples* for each mapped ontology relation is as follows:

(Step 1) *if* the candidate ontology entity for a relation is an ontology property, it creates ontology triples with the mappings for the arguments corresponding to the domain and range of the relation. Moreover, in *what-is* queries, where there is no information about the type of the *wh-query* term, through the ontology relationships that are valid for the ontology mapped term, we can identify a set of candidate values or ontological terms for the *wh-query term* that can complete the triple;

(Step 2) If the relation is presented as an ontology class, the RSS generates ontology triples that links the first argument, the ontology class and the second argument together, i.e., the query "who has publications at iswc?" generates the ontology triples: <*person, have-publication, publication*> & <*publication, in-proceedings, iswc*>;

(Step 3) If there are candidate mappings for the arguments of the triple and the relation, but there are no valid ontology triples that link them together, then the RSS ignores the relation name and looks for ontology triples between the arguments only. The rationale behind this is that the meaning in a relation is given by the type of its domain and its range, rather than by its name. For example, in "List the researchers that work in the OpenK project", the relation "work" can have been mapped to the ontology relation "occupation (hypernym)" or to the ontology class "learning", however no ontology triple is found for those when considering the arguments "researcher" and "Open Knowledge project". However, considering only the arguments, the following *Onto Triples* are obtained: <*research-staff-member, has-project-member, OpenK*> and <*research-staff-member, has-project-leader, OpenK*>.

3.5 An illustrative examples of the Similarity Services.

As said earlier, the number of *Linguistic Triples* obtained by the Linguistic Component is not fixed a priori and can be increased when nominal compounds that match more than one ontology term are present, or when there is no ontology that covers the whole triple. Analogously the number of resulting *Onto Triples* also depend on the way the ontology schema is organized. In fact, a typical situation is when the structure of triples in the ontology does not match the way the information was represented in the *Linguistic triples* and more *Onto Triples* are created at run-time to generate an equivalent representation according to the ontologies. Here, we explore this situation with some examples.

Consider the query "which KMi researchers working in the Semantic Web have publications in the iswc conference?" and the subset of ontologies in figure 4. The resultant semantically equivalent *Onto Triples* are presented in table 2. Note that the first *Linguistic triple* <*KMi researchers, working, Semantic Web*> has a translation (mappings) in both ontologies, while the second linguistic triple <*KMi researchers, have publications, iswc conference*> can only be resolved by the second ontology.

Linguistic terms can be mapped into (1) ontology classes (i.e., "KMi-researchers"), (2) instances (i.e. "Semantic-web-area" and "iswc conference" where "iswc" is an instance of "conference"[6]), or (3) a new triple (like the nominal compound "KMi researchers" into the triple <*academics, Belongs-to, KMi*>). However, as said before, to minimize noise and avoid increasing unnecessarily the number of mapping outcomes, option (3) is only used if option (1) and (2) does not produce any mappings, which is not the case here.

For linguistic relations, the simple case is when they are mapped into ontology relations, like "working" into "has-interest-on" in the case of the first triple. In other cases a linguistic triple may need to be mapped to more than one ontology triple within the same ontology.

[6] This can also be represented as a new "IS-A" triple: <*iscw, IS-A, conference*> but in any case, the meaning is the same.

For instance, the relation "have publications" is mapped in the ontology B through the mediating concept "papers", and a new triple is created to represent the indirect relationship (<*academics, wrote, papers*> <*papers, accepted-in, international semantic web conference*>). Moreover, as seen in Table 2, we can have more than one valid candidate interpretation (or set of *Onto Triples*), from the same or different ontologies, for each *Linguistic triple*.

While different triples may belong to the same or different ontologies, they have to be also linked by at least one common term. For instance, in the previous example the "linking" term is "KMi-researchers" in ontology 1 and "academics" in ontology 2, these two terms are semantically equivalent and their instances can be merged to generate an answer.

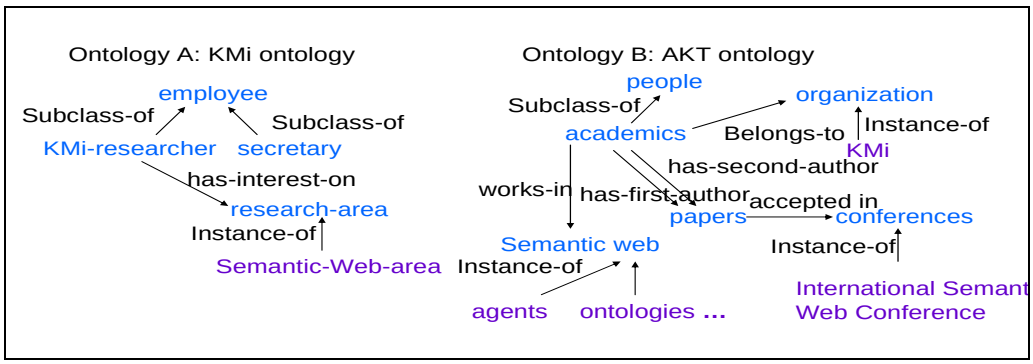


Figure 4. **Ontology scenario example**

Table 2. Triples representation

Query-triples (linguistic triples)	Onto-triples (ontology compatible triples)
<kmi researchers, working, semantic web>	*Ontology 1:* [kmi-researchers, has-interest-on, semantic-web-area]
<kmi researchers, have publications, iswc conference>	*Ontology 2:* [academics, has-first-author] [papers, accepted-in, iswc]
	Ontology 2: [academics, has-other-authors, papers] [papers, accepted-in, iswc]

4. Power Aqua in Action

A demo of the first PowerAqua prototype can be found at http://kmi.open.ac.uk/technologies/aqualog/okdeliverable.

We have tested our prototype on a collection of ontologies saved into online repositories and indexed by PowerMap but in the meantime we are working on adapting it to directly fetch relevant ontologies through a plug-in for the search engine WATSON[7], which has currently crawled around 7K ontologies. Our collection of ontologies includes high level ontologies, like ATO, TAP, SUMO, DOLCE, and very large ontologies like SWETO_DBLP or SWETO [1] with around 800.000 entities and 1.600.000 relationships.

Consider the simple query "What are the cities of Spain?" (<*what-is, cities, Spain*>), where both the *sweto*[8] and the *agrovoc*[9] ontologies are selected as relevant by PowerMap, as they have candidate matches for both arguments ("Spain" and "cities"), so they

[7] http://watson.kmi.open.ac.uk

[8] http://lsdis.cs.uga.edu/projects/semdis/sweto/

[9] http://www.few.vu.nl/wrvhage/oaei2006/

potentially cover the linguistic triple. Then, the similarity services are called to try to make sense of each linguistic triple, and its candidate element mappings, by analyzing the ontology taxonomy and relationships. Essentially, the triple similarity services are responsible for mapping each linguistic triple into one or more ontology triples within each relevant ontology, if possible.

In this example, both ontologies represent the linguistic relation "cities" as the ontology class "city". Therefore, the RSS generates ontology triples that link the first argument, the class "city", and the second argument together. In the case of *agrovoc*, the second argument is the instance "Spain", therefore, the problem becomes one of finding *ad-hoc* relations which link the *query* term "city" with the instance "Spain" (superclasses and subclasses are considered due to the inheritance of relations through the subsumption hierarchy). In the case that no ontological relations were found, it looks for indirect relations through mediating concepts between both arguments. In this case, the answer would then be the instances of "city" which have a relationship with "Spain".

For the *sweto* ontology, "Spain" corresponds to a literal, therefore it looks for all the instances of "city" that have "Spain" as the value of one of its attributes. The resultant list of instances from both ontologies should be merged to generate a more complete answer.

Furthermore, in order to generate an ontological interpretation of a query, and therefore an answer, nominal compounds terms, like "rock albums", which are translated into two ontology terms, are represented by a new ontology triple that links them together. For instance the resultant ontology triples for "show me rock albums" are: <album, has-albums, group> <group, has genre, rock> in an ontology about music, as seen in Figure 5

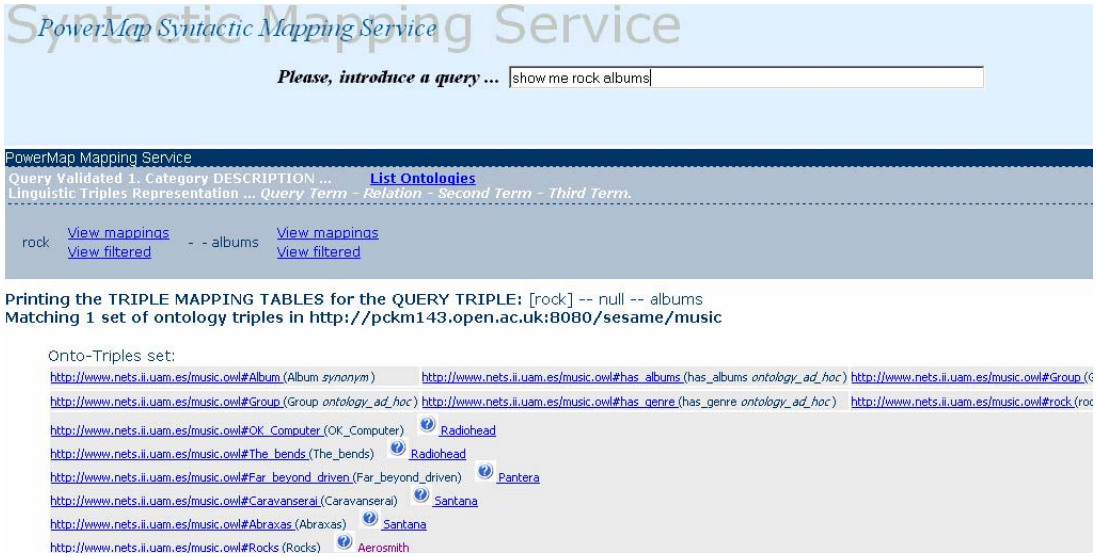


Figure 5. Screenshot of the example "Show me rock albums"

5. Conclusions

Exploiting the large heterogeneous Semantic Web is essentially about discovering interesting connections between items in a meaningful way. PowerAqua provides a natural language front end, which makes it possible to perform Question Answering on the Semantic Web, hence supporting such discovery process. This contrasts sharply with formal query languages for the semantic web, such as RDQL or SPARQL, which not only can be used solely by experts, but in addition are unable to perform queries across

ontologies. Hence they cannot be used to support such process of discovering and linking information spread across multiple sources.

In this first prototype, almost all the components of the final version of PowerAqua are already present, even though the current system can only answer basic linguistic queries (such as, assertions requiring an affirmation/negation as an answer, wh-queries, or imperative commands like list, give, tell, name, etc, represented by only one linguistic triple that relates two terms together).

Currently we are working on extending the range of queries the system is able to handle, in particular finalizing the implementation of the techniques needed for decomposing queries into multiple linguistic queries, mapping these to several relevant ontologies and then integrating the results. In addition, we also plan to achieve a tight integration with Watson, the aforementioned ontology search engine. Finally, more work is needed on the user interface, to improve the presentation of results to the users.

References

[1] Aleman-Meza, B., Halaschek, C., Sheth, A., Arpinar, I. B., Sannapareddy, G. (2004). "SWETO: Large-Scale Semantic Web Test-bed. *In Proc. of the16th Intl. Conf. on Software Engineering & Knowledge Engineering: Intl. Workshop on Ontology in Action*

[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] Buitelaar, P., Declerck, T., Calzolari, N., Lenci, A. (2003). Language Resources and the Semantic Web. *In Proc. of the ELSNET/ENABLER Workshop.*

[4] 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 Int Conf on Intelligent User Interfaces*.

[5] Ide N. and Veronis J. Word Sense Disambiguation: The State of the Art. *Computational Linguistics*, 24(1):1-40. (1998).

[6] Lopez, V., Motta, E. and Uren, V. (2006). PowerAqua: Fishing the Semantic Web, European Semantic Web Conference 2006, Montenegro.

[7] 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*, 5, 2, pp. 72-105, Elsevier.

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

[9] Resnik P. Disambiguating noun grouping with respect to WordNet senses. *In* Proc. *of the 3rd Workshop on very Large Corpora*. MIT (1995).

[10] Sabou, M., d'Aquin, M., and Motta, E. (2006). *In Proc. of the International Workshop on Ontology Matching.*

[11] Sabou, M., Lopez, V., Motta, E. (2006). Ontology Selection on the Real Semantic Web: How to Cover the Queens Birthday Dinner?. *In Proc. of the EKAW*.

[12] The Semantic Web. Berners-Lee, T., Hendler, J. and Lassila, O. *Scientific American, 284(5):* 33-43 (2001)

[13] Wu Z., and Palmer, M. (2004). Verb Semantics and Lexical Selection. *In Proc of the 32nd Annual Meeting of the Associations for Computational Linguistics.*