PowerAqua: Fishing the Semantic Web

Vanessa Lopez, Enrico Motta, and Victoria Uren

Knowledge Media Institute & Centre for Research in Computing, The Open University.
Walton Hall, Milton Keynes,MK7 6AA, United Kingdom.
{v.lopez, e.motta, v.s.uren}@open.ac.uk

Abstract. The Semantic Web (SW) offers an opportunity to develop novel, sophisticated forms of question answering (QA). Specifically, the availability of distributed semantic markup on a large scale opens the way to QA systems which can make use of such semantic information to provide precise, formally derived answers to questions. At the same time the distributed, heterogeneous, large-scale nature of the semantic information introduces significant challenges. In this paper we describe the design of a QA system, PowerAqua, designed to exploit semantic markup on the web to provide answers to questions posed in natural language. PowerAqua does not assume that the user has any prior information about the semantic resources. The system takes as input a natural language query, translates it into a set of logical queries, which are then answered by consulting and aggregating information derived from multiple heterogeneous semantic sources.

1 Introduction

The development of a semantic layer on top of web contents and services, the *Semantic Web* [1], has been recognized as the next step in the evolution of the World Wide Web as a distributed knowledge resource. The Semantic Web brings to the web the idea of having data formally defined and linked in a way that they can be used for effective information discovery, integration, reuse across various applications, and for service automation.

Ontologies play a crucial role on the SW: they provide the conceptual infrastructure supporting semantic interoperability, addressing data heterogeneity [2] and opening up opportunities for automated information processing [3]. However, because of the SW's distributed nature, data will inevitably be associated with different ontologies and therefore ontologies themselves will introduce heterogeneity. Different ontologies may describe similar domains, but using different terminologies, while others may have overlapping domains: i.e. given two ontologies, the same entity can be given different names or simply be defined in different ways.

Our goal is to design and develop a Question Answering (QA) system, able to exploit the availability of distributed, ontology-based semantic markup on the web to answer questions posed in natural language (NL). A user must be able to pose NL queries without being aware of which information sources exist, the details associated with interacting with each source, or the particular vocabulary used by the sources. We call this system *PowerAqua*.

PowerAqua follows from an earlier system, AquaLog [4], and addresses its main limitation, as discussed in the next section.

2 The AquaLog question answering system

AquaLog [4] is a fully implemented ontology-driven QA system, which takes an ontology and a NL query as an input and returns answers drawn from semantic markup compliant with the input ontology. In contrast with much existing work on ontology-driven QA, which tends to focus on the use of ontologies to support query expansion in information retrieval [5], AquaLog exploits the availability of semantic statements to provide precise answers to complex queries expressed in NL.

An important feature of AquaLog is its ability to make use of generic lexical resources, such as WordNet, as well as the structure of the input ontology, to make sense of the terms and relations expressed in the input query. Naturally, these terms and relations normally match the terminology and concepts familiar to the user rather than those used in the ontology.

Another important feature of AquaLog is that it is *portable with respect to ontologies*. In other words, the time required to configure AquaLog for a particular ontology is negligible. The reason for this is that the architecture of the system and the reasoning methods are completely domain-independent, relying on an understanding of general-purpose knowledge representation languages, such as OWL[1], and the use of generic lexical resources, such as WordNet. AquaLog also includes a learning mechanism, which ensures that, for a given ontology and community of users, its performance improves over time, as the users can easily correct mistakes and allow AquaLog to learn novel associations between the relations used by users, which are expressed in natural language, and the ontology structure.

AquaLog uses a sequential process model (see Figure 1), in which NL input is first translated into a set of intermediate representations – these are called *query triples*, by the Linguistic Component. The Linguistic Component uses the GATE infrastructure and resources [6] to obtain a set of syntactic annotations associated with the input query. The set of annotations is extended by the use of *JAPE* grammars to identify terms, relations, question indicators (who, what, etc.), features (voice and tense) and to classify the query into a category. Knowing the category of the query and having the GATE annotations for the query, it becomes straightforward for the Linguistic Component to automatically create the Query-Triples. Then, these query triples are further processed and interpreted by the Relation Similarity Service Component, which uses the available lexical resources and the structure and vocabulary of the ontology to map them to ontology-compliant semantic markup or triples.

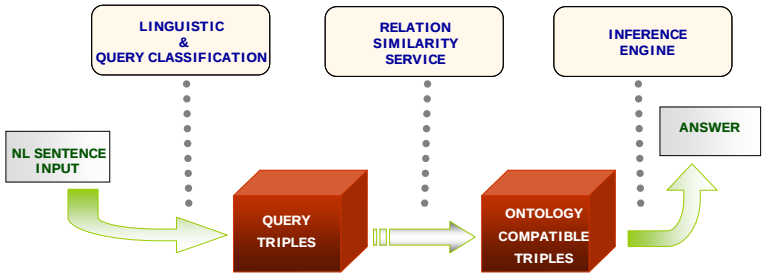

Figure 1. The AquaLog Data Model

However AquaLog suffers from a key limitation: at any time it can only be used for one particular ontology. This of course works well in many scenarios, e.g. in company intranets where a shared organizational ontology is used to describe resources. However, if we consider the

[1] A plug-in mechanism and a generic API ensure that different Knowledge Representation languages can be used.

SW *in the large*, this assumption no longer holds. As already pointed out, the semantic web is heterogeneous in nature and it is not possible to determine in advance which ontologies will be relevant to a particular query. Moreover, it is often the case that queries can only be solved by composing heterogeneous information derived from multiple information sources that are autonomously created and maintained. Hence, to perform effective QA on the semantic web, we need a system which is able to locate and aggregate information, without any pre-formulated assumption about the ontological structure of the relevant information.

3 QA for the Semantic Web: multiple-ontology scenario

In the previous sections we have sketched our vision for a QA system suitable for the semantic web, PowerAqua, and we have also explained why AquaLog does not quite fit the bill. In this section we address the problem in more detail and we examine the specific issues which need to be tackled in order to develop PowerAqua. It should be noted that here we only focus on the issues which are specific to PowerAqua and are not tackled already by AquaLog. For instance, we will not be looking at the problem of translating from NL into triples: the AquaLog solution, which is based on GATE, can be simply reused for PowerAqua.

Resource discovery and information focusing
PowerAqua aims to support QA on the open, heterogeneous Semantic Web. In principle, any markup associated with any ontology can be potentially relevant. Hence, in contrast with AquaLog, which simply needs to retrieve all semantic resources which are based on a given ontology, PowerAqua has to automatically identify the relevant semantic markup from a large and heterogeneous semantic web[2]. In this paper we do not address the problem of scalability or efficiency in determining the relevance of the ontologies, in respect to a query. Currently, there are ontology search engines, such as Swoogle [7] and different RDF ontology storage technologies suitable for processing SW information [8], e.g. 3store and Sesame servers.
Mapping user terminology into ontology terminology
A key design criterion for both AquaLog and PowerAqua is that the user is free to use his / her own terminology when posing a query. So, while this is an issue also for AquaLog, a critical problem for PowerAqua, not applicable to AquaLog, is that of different vocabularies used by different ontologies to describe similar information across domains [9].
Integrating information from different semantic sources
Queries posed by end-users may need to be answered not by a single knowledge source but by consulting multiple sources, and therefore, combining the relevant *information* from different repositories. On other occasions more than one source contains a satisfactory answer to the same query. Thus, if there is a complete translation into one or more ontologies or if the current partial translation, in conjunction with previously generated partial translations, is equivalent to the original query, the data must be retrieved from the relevant ontologies and appropriately combined to give the final answer. Interestingly, the problem of integrating information from multiple sources in the first instance can be reduced to the problem of identifying multiple occurrences of individuals in the sources in question.

[2] Here we do not need to worry about the precise mechanism used to index and locate an ontology and the relevant semantic markup. Various solutions are in principle possible depending on the SW evolution, here we can simply assume that the semantic web will provide the appropriate indexing mechanisms, much like the cluster architecture used by Google provides indexing mechanisms for the web as a whole.

4 Methodology: Query-driven semantic mapping algorithm step by step

The algorithm presented here covers the design of the whole PowerAqua system. However the AquaLog components reusable for PowerAqua have already been described in detail in [4], so here they will be described only briefly. In this paper, we focus primarily on the issues of mapping user terminology into ontology terminology in a semantic web multi-ontology scenario, and the information integration problem.

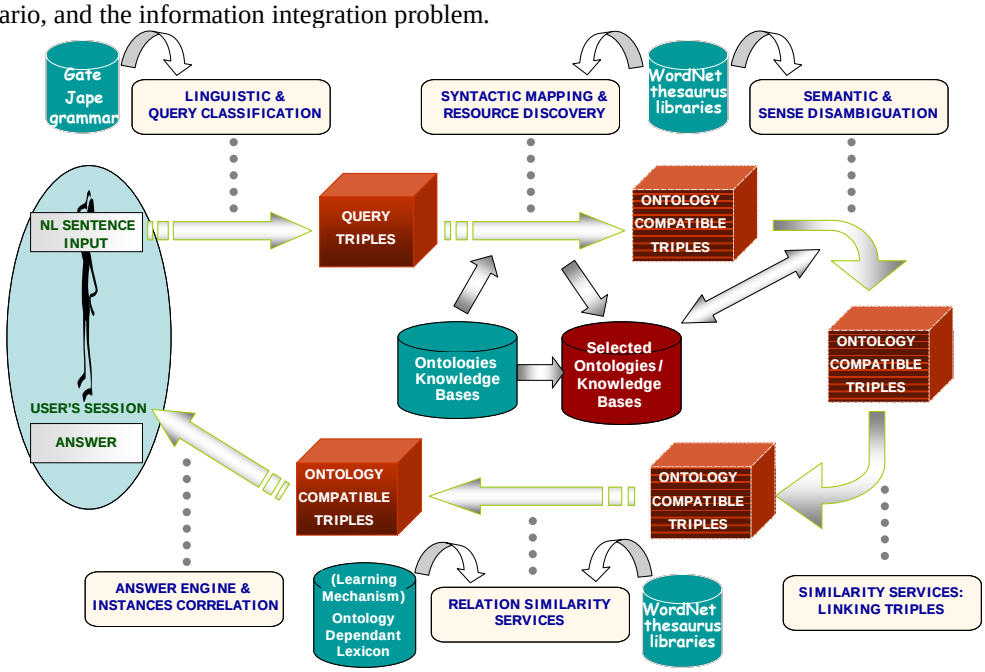

Figure 2 . Algorithm step by step

To help the reader make sense of the algorithm shown in Figure 2, we will use the query "What is the capital of Spain?" as a running example throughout. This query is particularly useful to present the issues introduced in section 3, especially when describing the different ways in which the PowerAqua algorithm interprets the above query and the query "Was *Capital*[3] written in Spain?".

4.1 STEP 1: LINGUISTIC & QUERY CLASSIFICATION ANALYSIS.

The Linguistic Component's task is to map the NL input query into Query-Triples. The role of the Query-Triples is simply to provide an easy way to manipulate the input. AquaLog linguistic component [4] is appropriate for the linguistic analysis thanks to its ontology portable and independent nature, and therefore, it is reused for PowerAqua.

4.1.1 Running Example.

The example query "What is the capital of Spain?" is classified as a *wh-query*[4] that represents a binary relationship where there is not any information about the type of the *query term (fo-*

[3] Book written by Karl Marx (1867)

[4] The set of "wh-queries" are the ones starting with: what, when, where, are there any, does anybody, how many, and also imperative commands like list, give, tell, name, show. "wh-queries" like "who" can be interchanged into "which person/organization", "where" into "which location" and so on.

cus), and generates the linguistic triple: <?, capital, Spain>. However, the relation "is the capital of" contains the noun "capital", therefore, we need to take into account that the triple may be restructured as a triple with an implicit relation between the term "capital" and "Spain".

The second example query "Was Capital written in Spain?" is translated into a basic affirmative/negative query that generates the triple: <capital, written, Spain>.

4.2 STEP 2: SYNTACTIC TERM MAPPING AND RESOURCE DISCOVERY.

The initial selection of candidate ontologies, which may have the potential to answer the query, is entirely done by syntax driven techniques (SDT). Note that we use the same terminology as [10] referring to syntactic matching when the matching between two nodes is computed using the labels of the nodes. SDT looks for similarities between labels by means of string-based metrics[5], taking into account abbreviations, acronyms and domain and lexical knowledge.

4.2.1 Phase a: Extending the query vocabulary with lexical and domain knowledge.

To maximize recall, with respect to other ontology search systems that only looks for classes or instances that have labels matching a search term either exactly or partially [11], each term in the query, or noun in the relation if any (relations may be formed by a noun plus verbs and prepositions) is extended with its synonyms, hypernyms and hyponyms.

The current version of WordNet provides *a priori* lexical and domain knowledge. As Ide and Veronis state [12], WordNet is the most used lexical resource at present for disambiguation in English. Most of the research methods in the literature are limited to WordNet [13]. Nouns, verbs, adjectives, and adverbs are each organized into networks of synonyms sets (*synsets).* Each synset has a gloss to define it. There are nine types of semantic relations defined on the noun subnetwork: hyponymy (IS-A) relation, and its inverse hypernymy; six meronymic (PART-OF) relations – COMPONENT-OF, MEMBER-OF, SUBSTANCE-OF and their inverses; and the COMPLEMENT-OF relation.

4.2.2 Phase b: Syntactic matching of ontology terms.

Depending on the query category, the system will look for ontology instances, classes or both to map a term or its lexical variations. The system looks for ontology classes, which can be handled in the client memory, through the use of string distance metrics, also used in AquaLog.

SDT are used in AquaLog, however, the weakness of these techniques becomes more evident when applied to PowerAqua (see example in section 4.2.4). Firstly, the discovery of user terms in the ontology by the use of SDT becomes increasingly computationally expensive as the number of ontologies increases. Secondly, many of the discovered ontology terms syntactically related with the query terms, obtained as a result of applying SDT, may be similarly spelled words (labels) that do not have precisely matched meanings. As already indicated in section 3, in this paper we will not address the issues to do with the efficiency and scalability of the algorithm in determining the relevance of the ontology and terms by use of SDT, but we will focus on the issue of disambiguating among the possible interpretations of a query.

4.2.3 Phase c: Complete coverage of the triple by candidate ontologies.

A criterion for filtering candidate ontologies is to select the ones that present potential candidates mappings for all the terms within a triple, if any. In other words, if ontology 1 presents a possible complete translation of a query triple, while ontology 2 only presents a partial translation of the same triple, the later will be discarded. Similarly, the coverage of an ontology given the search terms is used as a measure in the ontology ranking approach on AKTiveRank [11].

---

[5] http://secondstring.sourceforge.net/

Consider the query “Which wine is appropriate with chicken?”. The term “wine” has a syntactic mapping with the term “wine” belonging to an ontology of colors, and with the term “wines” related to an ontology of food and wines. Similarly, the term “chicken” maps to an ontology of farming and to the same food and wine ontology. Since the food and wine ontology presents a complete potential translation for the triple we retain it, and we discard both the farming and color ontologies, which only present partial translations.

However, we may find the case in which none of the available ontologies contains a whole translation of the triple. 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>.

In some cases, it may happen that no candidate terms are found due to the vocabulary used in some ontologies, e.g. labels with multiple words. In this case, if there is a possible mapping for one of the two query terms on the triple, we can identify a set of possible candidate terms that can complete the triple through the ontology relationships valid for this mapped term.

4.2.4 Running example.

Through WordNet we get the synonyms, hypernyms and hyponyms presented in Table 1.

Table 1. Lexical related Words obtained in WordNet

Capital (glosses)	Synonyms	Hypernyms	Hyponyms
#1: assets available for use in the production of further assets	working capital	assets	stock, venture capital, risk capital, operating capital
#2: wealth in the form of money or property	-	assets	endowment, endowment fund, means, substance, principal, corpus, sum
#3: a seat of government	-	seat	Camelot, national / provincial / state capital
#4: one of the large alphabetic characters used as the first letter	capital letter, uppercase, majuscule	character, grapheme, graphic symbol	small capital, small cap
#5: a book written by Karl Marx	Das Kapital, Capital	book (*instance-of*)	-
#6: the upper part of a column that supports the entablature	capital, chapiter, cap	top	-

As said in 4.1.1, the relation in the query example “What is the capital of Spain?” is the noun “capital”, and therefore it can be understood as a) an ontology relation or as a b) query term that should be mapped into an ontology class. After running phases b and c, the system obtains the following ontologies:

- Ontology 1: Geographical information. Contains the terms “capital-city” as a candidate mapping for “capital” and “Spain” as an instance of “country”. There is a direct relation that connects “capital-city” and “country”.
- Ontology 2: Financial ontology. Contains the terms “capital” and “Spain” as an instance of “country”. The classes “capital” and “country” are related through the concept “company”.
- Ontology 3: Country statistics. Contains the term “Spain”.
- Ontology 4: flights information. Contains the term “Logrono” (a Spanish city), where “Logrono” is a WordNet hyponym of the only synset of “Spain”.

In Ontology 1 and 2, the query triple “capital” is understood as an ontology class, and therefore, the resultant triple will be an unknown relation between “capital-city / capital” and “Spain”. For the ontology 3 and 4 “capital” is understood as an ontology relation, therefore the ontologies contains only a mapping for the term “Spain”, as relations are not addressed until the step in section 4.4..

At this stage we have selected the candidate ontology terms that potentially will be part of the equivalent ontology semantic query by a simple lexical analysis of the labels (SDT). In the next phase the system performs sense disambiguation using the ontology semantics and WordNet to analyze the meaning and discard non-related ontology terms mapped in this phase.

Also, it is worth mentioning that in the question "Was Capital written in Spain?", where the triple is <*capital, written, Spain*>, the system should only obtain the following ontology:

- Ontology 5: Bibliographic information. Contains the terms "Das-Kapital" as an instance of "book", also "Spain" as an instance of a "country" (e.g. where a book is published, at this stage we do not know if "published" is the same as "written").

This is because the category of the query (affirmative-negative) is telling us that the term "capital" should be mapped into the instance "Das-Kapital", while in "What is the capital of Spain?" "capital" should be mapped into a class, and thanks to WordNet we know that "book" is related to "capital" by an "instance-of" kind of relationship not by an "hypernym".

4.3 STEP 3: SEMANTIC MAPPING FROM USER TERMINOLOGY INTO ONTOLOGY TERMINOLOGY.

The mapping between user and ontology terms becomes increasingly complicated as the number of ontologies increases. SDT (string metrics, lexicon, synonyms) used to select the candidate terms and ontologies are obviously not enough to identify relevant terms in the heterogeneous scenario introduced by multiple ontologies. A semantic mapping component that considers the content of an information item and its intended meaning is needed because:

– Calling the user to disambiguate between possible ontology candidate terms is not feasible because of the broad space of syntactically obtained distributed terms[6]: spelled words (labels) may have not precisely matched meanings. Relationships between word senses, not words, are needed. If we know the possible *senses* for the user's query we can filter the candidate results without the user's feedback.

– To answer a query the system may need to combine partial answers from more than one ontology, or two ontologies may provide compatible answers, e.g. answers which can be merged, to the same query. Semantic interoperability between two concepts is only possible if they are semantically equivalent, or in other words, instance information from different ontology classes can be correlated / integrated only if the ontology classes are semantically equivalent. We make the assumption that two ontology classes may be semantically equivalent, and denote compatible information, if the WordNet *senses* associated with the labels of the classes, in the context of their position in the ontology taxonomy, share some similarity. Otherwise they are just classes that share lexically-related labels but they refer to different domains and therefore their information is not compatible.

In this step the semantic equivalence of the candidate ontology terms obtained in step 2 is studied. As a consequence, ontology terms that are syntactically related to the terms in the query, but are not semantically equivalent, are discarded as potential mappings. The semantic equivalence, and therefore the word sense disambiguation (WSD), is measured through the notion of *similarity*. Many reasonable similarity measures and strategies exist in the literature for WSD (see [12] for a state of the art). Hence, to maximize our system applicability we propose a sense-based similarity matcher algorithm in section 4.3.1. This algorithm applied to PowerAqua is described in the steps 4.3.2 and 4.3.3.

---

[6] Interactivity should be the last resort for the *Similarity Services* (section 4.4) where, after a deep analysis of the ontology, domain knowledge does not further help to automatically perform disambiguation.

4.3.1 Semantic equivalence between two terms: sense-based similarity algorithm

To study similarity between terms the meaning of each term should be made explicit by an interpretation of its label and position in the ontology taxonomy (see 4.3.3). Note that similarity is a more specialized notion than association or relatedness. Similar entities are semantically related by virtue of their similarity (bank-trust company). Dissimilar entities may also be semantically related by lexical relationships such as meronym (*car-wheel)* and antonymy (*hot-cold)*, or just by any kind of functional relationship or frequent association (*pencil-paper, penguin-Antarctica*) [13]. Taking the example in [14] doctors are minimally similar to medicines and hospitals, since these things are all instances of "something having concrete existence, living or nonliving" (although they may be highly associated), but they are much more similar to lawyers, since both are kinds of professional people, and even more similar to nurses, since both are professional people within the health professions.

In *Hierarchy distance based matchers* [15] the relatedness between words is measured by the distance between two concepts/senses in a given input hierarchy. In particular, similarity between words is measured by looking at the shortest path between two given concepts/senses in the WordNet "IS-A" taxonomy of concepts.

Two words are similar if any of the following holds:

1. They have a synset in common (e.g. "human" and "person")
2. A word is a hypernym/hyponym in the taxonomy of the other word.
3. If there exists an allowable "is-a" path connecting a synset associated with each word –in the WordNet taxonomy-.
4. Additionally, if any of the previous cases is true and the definition (gloss) of one of the synsets of the word (or its direct hypernyms/hyponyms) includes the other word as one of its synonyms, we said that they are strongly similar.

For evaluating points 2 and 3 we make use of two WordNet indexes: the *depth* and the *common parent index (C.P.I)*. At the top of WordNet hierarchy are 11 abstract concepts or *unique beginners* (e.g. "entity"), the maximum depth in the noun hierarchy is 16 nodes. The shorter the path between two terms [14] the more similar they are, e.g. depth=1 represents case 3 ("is-a" path). However, a widely acknowledged problem is that the approach typically "relies on the notion that links in the taxonomy represent uniform distances", but typically this is not true and there is a wide variability in the "distance" covered by a single taxonomic link [13]. Resnik [14] established that one criterion of similarity between two concepts is the extent to which they share information in common, which, in an IS-A taxonomy, can be determined by inspecting the relative position of the most-specific concept that subsumes them both. With the use of the C.P.I we can immediately identify this lowest super-ordinate concept (lso) between two terms, or the most specific common subsumer. The number of links (depth) is still important to distinguish between any two pairs of concepts having the same lso. Apart from point 1 of the algorithm, in which the words have a synset in common, the most immediate case occurs in point 2 (C.P.I = 1, Depth = 1), e.g. while comparing "poultry" and "chicken" we notice that "poultry#2" is the common subsumer (hypernym) of "chicken#1".

4.3.2 Phase a: Filtering non-semantically equivalent candidate ontology terms with respect to a query by the use of similarity

SDT (string algorithms, synonyms) were used in the previous phases to select the first set of candidate terms and ontologies to map a query. Because of the use of SDT, the ontology mapped term and the query term do not necessarily share the same meaning. However, they must share some similarity in common; otherwise the candidate ontology term is discarded.

For instance, for a query like "What investigators work in the akt project?" the system, using string algorithms over WordNet synonyms, discovers the following terms as possible candidate mappings for "investigators": "researcher", "KMi-researchers", "research-worker",

"research-area". Using the WordNet "IS-A" taxonomy we must find at least one synset in common with the mapped ontology term and the query term or a short/relevant path in the IS-A WordNet taxonomy that relates them together. Otherwise it is discarded as a solution.

Here, "researcher" and "investigator" have a synset in common, namely "research-worker, researcher, investigator – a scientist who devotes himself to doing research". We get the same for "research-worker" and "KMi-researchers" (nominal compound which lemma is "researcher"). However "research-area" will be discarded (even if they may be highly associated) because not only do they not share any sense in common but also there is not a relevant "IS-A" path that connects "researcher" with "research-area"; "researcher" is connected to the root through the path "scientist/man of science" and "person", while "research-area" is connected through "investigation" which is connected to "work".

4.3.3 Phase b: Analysis of the semantic interoperability between candidate ontology terms by means of similarity measures.

Different ontology mappings for the same query term may represent different meanings of the query term, and therefore they are not necessarily semantically equivalent. Two classes are semantically interoperable or two instances are semantically equivalent if they are similar, following the algorithm in 4.3.1, for any of its possible WordNet *synsets*. The meaning of an ontology term is determined not only by its label but by its position in the ontology taxonomy (ancestors and descendants) and by the meaning of the rest of the concepts in the same taxonomy path (the context where the class or instance occurs).

The algorithm used to obtain the set of possible WordNet synsets valid for an ontology term as part of an ontology taxonomy is inspired by the algorithm described in [16] to make explicit the semantics hidden in schema models: Let L be a generic label for a concept and L1 either an ancestor label or a descendant label of L and let s* and s1* be respectively the sets of WordNet senses of a word in L and a word in L1. If one of the senses belonging to s* is either a synonym, hypernym, holonym, hyponym or a meronym of one of the senses belonging to s1*, these two senses are retained and all the other senses are discarded. As an example, imagine *Apple* (which can denote either a fruit or a tree) and *Food* as its ancestor; since there exits a hyponymy relation between *apple#1* (denoting a fruit) and *food#1*, we retain *apple#1* and discard *apple#2* (denoting a tree). Note this phase works better when the ontology term is a class instead of an instance, as WordNet may not have the correct sense for a proper name. This phase is further described in the running example.

4.3.4 Running example.

Going back to the example "What is the capital of Spain?" the mappings for "capital" for the geographical and financial ontologies are "capital-city" and "capital" respectively. After execution of *phase a* both interpretations remain, as the lemma for both terms is the same as the query term "capital" and therefore, in principle, they have all the synsets in common. In *phase b* the system will study whether both interpretations are semantically equivalent by obtaining the *sense* of the mapped term in the context of the ontology it belongs to.

For instance, we run the algorithm of similarity presented in 4.3.1 to obtain the *synset* of the term "capital" in the geographical ontology. We obtain the results presented in table 3 when trying to find an allowable path between all the senses of the candidate ontology word "capital" and all the senses of its ancestor "city" (please note that blank means that either there is not an allowable path or the depth is too long to be considered as relevant).

Analyzing the results of table 3 we can quickly filter *capital#c*, *capital#f*, *city#1*, *city#2* and discard the others. A deeper study will show that *capital#c* is more likely than *capital#f* because there are only 2 common subsumers in the latter (entity and location), both of them representing abstracts top elements of the WordNet taxonomy, while in the former we have 3

common subsumers. We can not study the descendants of "capital" in the ontology because none exist. The study of the next direct ascendant of "city" ("geographical-unit") does not offer additional information (the fine-grainedness of WordNet sense distinctions, e.g. in this case city#1 and city#2, is a frequently cited problem). Moreover, the hypernym of *capital#c* is *"seat#5"*, defined as "seat –centre of authority (*city* from which authority is exercised)". The word "city" is used as part of its definition. Therefore *capital#c* is strongly related to "*city"*.

Table 3. Similarity between "capital" and its ontology ancestor "city" using WordNet "IS-A" taxonomy

	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 ..)			
Capital#b (wealth ..)			
Capital#c (seat of government)	**Depth** = 8, **lso** = region **Num_so** (common subsumers) = 3 (region, location, entity)	**Depth** = 7, **lso** = region **Num_so** = 3 (entity, location, region)	
Capital#d (capital letter)			
Capital#e (book by Karl Marx)			
Capital#f (upper part column)	**Depth** = 8**, lso** = location **Num_so** = 2 (entity, location)	**Depth** = 7, **lso** = location **Num_so** = 2 (entity, location)	

After *phase b* it is clear that in the financial ontology "capital" is referred to senses #1 and #2, while in geographical ontology "capital" is referred to sense *#3*. Therefore both terms in different ontologies are not semantically equivalent and their information cannot be correlated (even if they share the same label) which means that the system must select one of them using ontology semantics or query relatedness in the following steps.

4.3.5 Selection of candidate ontology terms using the notion of Relatedness

After the execution of previous steps, we have narrowed down to two the valid mappings for the linguistic triple: *?(capital, Spain)*, one in the geographical ontology and the other one in the financial ontology. We also know that there is not semantic interoperability or equivalence between the class "capital" represented in both ontologies, therefore only one mapping will be valid to create the final ontology compliant triple.

The next step (section 4.4) is the study of the ontology taxonomy and relationships to analyze the *relatedness* between ontology terms to choose a correct mapping for the query. However, it is worth mentioning that we also consider the study of the sense of term "capital" in the user's query by using the idea of relatedness found in the computational linguistics literature. Most approaches assume that words that appear together in a sentence can be disambiguated by assigning to them the senses that are most closely related to their neighboring words [17]. For instance, in "What is the capital of Spain?", for a human user it is obvious that *capital#c*, should be adopted when considering only Spain as the neighborhood term. Pendersen and his colleagues [17] have made available a Perl implementation of six WordNet measures evaluated in [13] plus their own sense disambiguation algorithm based on glosses [17] to assign a meaning to every content word in a text. Basically, these measures look for a path connecting a synset associated with each word, e.g. in Hirst and St-Onge measure the intuition behind is "the longer the path and the more changes of direction (upward for hypernym and meronym; downward for hyponymy and holonymy and horizontal for antonymy) the lower the weight". In [17] *extended semantic gloss matchers* measure semantic relatedness between concepts (and its ancestors/descendants according to the *is-a* WordNet hierarchy) that is based on the number of shared words in their definitions (glosses).

SDT based on text is not mature enough because there are useful computational methods in the literature only for *quantifying semantic distances for non-ad hoc relationships.* However, *r*elatedness includes not just the WordNet relationships but also *associative* and *ad hoc* relationships. These can include just about any kind of functional relation or frequent association in the world (i,e bed-sleep), sometimes constructed in the context, and cannot always be determined purely from *a priori* lexical resources such as WordNet.

We believe that in our PowerAqua scenario we can take advantage of the relatedness expressed in the ontology semantics to filter the correct candidate ontology triples equivalent to the user query triples, without the need to apply techniques for text relatedness. This is explained in section 4.4.

4.4 RELATION SIMILARITY SERVICES AND LINKING TRIPLES.

Essentially, the relation similarity service (RSS) tries to make sense of the input query and express it in the form of ontology relationships between ontology terms. The RSS is invoked after all the linguistic terminology is mapped into ontology terms (classes or instances). The RSS is responsible of creating the ontology compliant triples by a) linking the mapped ontology terms to create 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. For step b) while different triples may belong or not to different ontologies they have to be also linked by at least one common term.

AquaLog mechanisms for step a) and b) can be reused. Briefly, for step a) AquaLog looks for a set of possible ontology relationships between two terms by looking at the structure in the ontology. This set is further disambiguated by the use of distance metrics, or as the vocabulary of the user may have a number of discrepancies with the vocabulary of the ontology it also uses WordNet and a learning mechanism. For step b) sentences that are structurally ambiguous, in the way they are linked, can be disambiguated using domain knowledge or in the last instance by calling the user to choose between alternative readings.

There is not a single strategy here; basically it depends on the query category and ontology structure. A typical situation is when the structure of triples in the ontology do not match the way the information was represented in the query triples. We explore this situation with the following example: consider the query "which KMi researchers working in the Semantic Web have publications in the ESWC conference?" and the subset of ontologies in figure 3. The resultant semantically equivalent mappings or ontology-compliant-triples are presented in table 4. Note that the first query triple <*KMi researchers, working, Semantic Web*> has a translation in both ontologies, while the second query triple <*KMi researchers, have publications, eswc conference*> can only be resolved by the second ontology.

The number of query triples is fixed *a priori* for each query category, however the final number of ontology triples is not obvious at the first stage and it is dependent on the ontology semantics. Therefore, triples must be created at run-time to generate an equivalent representation according to the ontologies. Linguistic terms can be mapped into ontology classes (i.e., "Kmi-researchers"), instances ("Semantic-web-area", "ISWC conference), or even a new triple (like the nominal compound "KMi researchers" into the triple <academics, Belongs-to, KMi>).

Different situations can be found by the similarity services when looking for a proper relation mapping. For instance, the simple case is when a linguistic relation is mapped into a ontology relation like "working" into "has-interest-on" in the case of the first triple. In other cases, to map a relation a new triple must be created, for instance, the relation "have publications" is mapped in the ontology B though the mediating concept "papers", and a new triple is created to represent the indirect relationship (<*academics, wrote, papers*> <*papers, accepted-in, european semantic web conference*>). Other mapping situations can be found in [4].

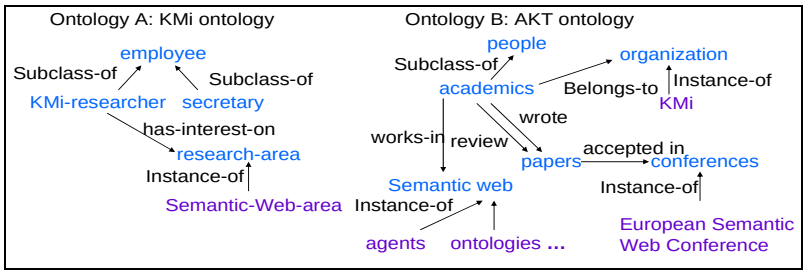

Figure 3. Ontology scenario example

Table 4. 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]	*Ontology 2:* [academics, belongs-to, kmi] [academics, works-in, semantic-web]
<kmi researchers, have publications, eswc conference>	*Ontology 2:* [academics, wrote, papers] [papers, accepted-in, european semantic web conference]	

4.4.1 Running example.

As said before, through the use of WordNet and the ontology we have narrowed down to two valid non-equivalent mappings for the linguistic triple: <*capital, ?, Spain*>, one in the geographical ontology and the other one in the financial ontology. A deeper analysis of the ontology relationships will find a direct relation that connects any country, e.g. Spain, with its capital for the geographical ontology. However, in the financial ontology there is not a direct relation between countries and capital. There is a mediating concept that represents a company, that has a series of capital goods and it is based in a country. This is a strong indication that the geographical ontology is more related to our query and should be selected.

For the linguistic triple <?, capital, Spain> where capital is considered a relation, a relationship analysis will uncover the relation "is-capital-of" between "country" and "city" in ontology 3 (country statistics), while in the ontology 4 (flight information) there are not any relations similar to "is-capital-of". Therefore ontology 3 is selected.

Note that both triple representations are valid representations of the query and semantically equivalent to each other (they refer to "city" as the ascendant of "capital" in one ontology or as the type of the relation "is-capital-of" in the other ontology). In the next phases of the algorithm, an answer can be generated by correlating both results, e.g. identifying the common instance "Madrid" as an answer, or by selecting one representation.

4.5 GENERATING AN ANSWER.

A key issue when generating an answer is to identify if semantically equivalent concepts in the ontology triples have overlapping information and, in such a case, perform the fusion of instances. For example, in the *KMi semantic web portal ontology*, the instance "Nigel Shadbolt" from the class "researcher" has some basic information, but an instance about the same person has also been defined in the *AKT web portal ontology* under the class "AKT-researcher".

4.5.1 Phase a: Operational Combination of triples.

AquaLog provides two mechanisms (depending on the triple categories) for operationally integrating the triples information to generate an answer. These mechanisms are: (1) and/or link: e.g., in "who has an interest in ontologies or in knowledge reuse?", the result will be a fusion of the instances of people who have an interest in ontologies and the people who are

interested in knowledge reuse; (2) conditional link, in which we can differentiate between: a) conditional link to a term: e.g. in ”which KMi academics work in the akt project sponsored by eprsc?” the second triple <*akt project, sponsored, eprsc*> must be resolved and the instance representing the “akt project sponsored by eprsc” identified to get the list of academics required for the first triple <*KMi academics, work, akt project*>; and b) conditional link to a triple: e.g. in ”What are the homepages of the researchers working on the semantic web?” the second triple <*researchers, working, semantic web*> must be resolved and the list of researchers obtained prior to generating an answer for the first triple <*?, homepage, researchers*>.

4.5.2 Information Correlation: identify common instances.

It is common to get semantically equivalent triples from different ontologies, as a translation of one query triple. The challenge is to identify the instances in common between the two equivalent terms in each triple. For example, the query “Who are the academics working on the Semantic Web?” might have a complete translation in the ontology X about researchers in KMi, ontology Y about academics in the University of Trento and ontology Z about the AKT consortium. Ontologies X and Y have no instances in common. However, ontologies X and Z contain overlapping information, as many of the academics in KMi belong to the AKT project. Common instances must be identified to give a complete non-redundant answer.

Furthermore, for queries represented by partial translations from different ontologies the identification of common instances is also a key issue. For instance, the query “What are the citations for the publications of Enrico Motta?” is solved by an ontology about citations and an ontology about academics in which the instance “Enrico Motta” is related to his publications. The publications from the academics ontology must be identified in the citations ontology.

Identifying whether two instances from semantically equivalent concepts are the same is not an easy task. Instances may not have the same name, and information about the same instance can have different purposes, e.g. the description of a car for sale or for an environmental study. We can use the OWL mechanism which identifies the attributes that provide sufficient evidence that two instances are the same. However, further mechanisms need to be adopted, e.g., use of joint probability approaches similar to GLUE[3] over the instance full name (from the taxonomy root) and its textual content (word frequency over attributes and values)

7 Related Work

The AquaLog linguistic component, reused for PowerAqua, in combination with the SW scenario provides a new twist on the old issues associated to asking natural language queries to databases (NLDB). See [4] for comparisons between AquaLog and previous work in NLDB and open-domain NL QA systems. Here, we look at the solutions proposed in the literature to address semantic heterogeneity in information systems.

The Semantic Knowledge Articulation Tool (SKAT) [18] uses a first order logic notation to specify declarative matching rules between ontology terms. SKAT initially attempts to match nodes in the two graphs based on their labels and their structural similarity. The idea of presenting a conceptually unified view of the information space to the user, the *world-view*, is studied in [19]. The user can pose declarative queries in terms of the objects and the relations in the *world-view*. Given a query to the *world-view*, the query processor in the global information system poses subqueries to the external sources that contain the information relevant to answer a query. In order to do that, the semantic of the contents of the external sites is related to the *world-view* through the use of a description language. These solutions have an intrinsic limitation to be applied to the open-world domain introduced by the SW scenario, where the

distributed sources are constantly growing. And therefore, it is not possible to apply any closed-domain solution for environments with well-defined boundaries, like corporate intranets, in which the problem can be addressed by the specification of shared models like mapping rules, global ontologies/vocabularies, and definitions of conversion libraries or functions between semantic data/values, among others. The manual effort needed to maintain any kind of centralized/global shared approach for semantic mapping (i.e. to implement the previous solution) in the SW is not only very costly, in terms of maintaining the mappings for such a highly dynamic environment that evolves quickly, but also has the added difficulty of "negotiating" a shared model that suits the needs of all the parties involved [20].

In Query Processing in Global Information Systems [9] user queries are rewritten by using inter-ontology relationships to obtain semantic translations across ontologies. There are two restrictions: firstly the user must subscribe to the terminology and model captured by a chosen ontology. Secondly, the solution to the vocabulary problem is obtained through the declarative representation of synonym relationships relating ontology terms. The disadvantages are: 1) synonym relationship mappings must be maintained between terms in the user ontology and the underlying repositories. 2) Every time there is a change in the structure of underlying repositories the mappings of the component ontology must be change. 3) Such synonym relationships should be defined when a new ontology is added to the system (its centralized nature may affect the efficiency of the system). The advantage is that different partial answers can be easily correlated since all of them are expressed in the language of the user ontology.

CUPID [21] analyzes the factors that affect effectiveness of algorithms for automatic semantic reconciliations; however, this is a complementary goal to ours: our system matches terms and relations in an user's query with distributed ontologies while they match data repositories and ontologies. In GLUE [3] the probability of matching two concepts is studied by analyzing the available ontologies using a relaxation labeling methods; however, this approach is not very adaptable because it analyzes all the ontology concepts. Finally, In our QA-driven scenario there is no need for obtaining mappings for each pair of concepts belonging to different ontologies, in which the level of effort is at least linear in the number of matches to be performed [22] (see algorithms for the Match operator [22]). In our run-time scenario only relevant concepts to the user's query are analyzed (on-demand driven approach).

6 Summary

We have presented the design of PowerAqua, a novel QA system which provides answers drawn from multiple, heterogeneous and distributed ontologies on the Web. PowerAqua evolved from AquaLog, an implemented ontology-based QA system limited to one ontology at a time. The issues derived from opening the system with respect to the SW have been addressed here. A prototype based on the algorithm presented here will be implemented in the following months.

Acknowledgements

This work was partially supported by the AKT project sponsored by UK EPSRC and by the EU OpenKnowledge project (FP6-027253). Thanks to Yuangui Lei and Marta Sabou for useful input.

References

1. The Semantic Web. Berners-Lee, T., Hendler, J. and Lassila, O. *Scientific American, 284(5):* 33-43 (2001)
2. Semantic Integration of Heterogeneous Information Sources Using a Knowledge-Based System. Adams T., Dullea J., Clark P., Sripada S. and Barrett T. *In Proc. of the 5th International Conference on Computer Science and Informatics*, (2000).
3. Learning to Map between Ontologies on the Semantic Web. Doan A., Madhavan J., Domingos P., and Halevy A. *In Proc. of the World-Wide Web Conference* (2002).
4. AquaLog: An Ontology-portable Question Answering System for the Semantic Web. Lopez V., Pasin M. and Motta E. *In Proc. of the 2nd European Semantic Web Conference* (2005)
5. Question Answering on the SW. Mc Guinness,D. *IEEE Intelligent Systems,19(1)*82-85 (2004)
6. Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V.: GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. *In Proc. of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02).* Philadelphia (2002).
7. Swoogle: A semantic web search and metadata engine. X L. Ding et al. *In Proc. 13th ACM Conf. on Information And Knowledge Management* (2004)
8. An Evaluation of Knowledge Base Systems for Large OWL Datasets. Guo, Y., Pan, Z., Heflin, J. *International Semantic Web Conference* 274-288 (2004)
9. OBSERVER: An approach for Query Processing in Global Information Systems based on Interoperation across Pre-existing Ontologies. Mena E., Kashyap V., Sheth A. and Illarramendi A. *Distributed and Parallel Databases* 8(2): 223-271 (2000)
10. S-Match: an algorithm and an implementation of semantic matching. Giunchiglia F., Shvaiko P and Yatskevich M. *In Proc. of the 1t European Semantic Web Symposium* (2004).
11. Ontology Ranking based on the Analysis of Concept Structures. Alani, H. and Brewster, C. *In Proc, of the 3th International Conference on Knowledge Capture* (2005).
12. Word Sense Disambiguation: The State of the Art. Ide N. and Veronis J. *Computational Linguistics*, 24(1):1-40. (1998).
13. Evaluating WordNet-based measures of semantic distance. Budanitsky, A. and Hirst, G. *Computational Linguistics* (2006).
14. Disambiguating noun grouping with respect to WordNet senses. Resnik P. *In Proc. of the 3rd Workshop on very Large Corpora*. MIT (1995).
15. Element Level Semantic Matching. Giunchiglia F. and Yatskevich M. *Meaning Coordination and Negotiation Workshop, ISWC* (2004).
16. Making Explicit the Semantics Hidden in Schema Models. Magnini B., Serafín L., and Speranza M. *In Proc. of the Workshop on Human Language Technology for the Semantic Web and Web Services*, held at *ISWC-2003*, Sanibel Island, Florida, (2003).
17. Extended Gloss Overlaps as a Measure of Semantic Relatedness. Banerjee S., and Pedersen T. *International Joint Conference on Artificial Intelligence* (2003).
18. Semi-automatic Integration of Knowledge Sources. Mitra P., Wiederhold G., Jannink J. *In Proc. of the 2nd International Conf.erence on Information Fusion.* (1999).
19. Data Model and Query Evaluation in Global Information Systems. Levy A., Y., Srivastava D. and Kirk T. *Journal of Intelligence Information Systems.* 5(2): 121-143 (1995).
20. Semantic coordination: a new approach and an application. Bouquet P., Serafini L. and Zanobini S. *International Semantic Web Conference* 130-145 (2003).
21. Generic schema matching with cupid. Madhavan, J., Bernstein, P.A. and Rahm, E. *The Very Large Databases Journal*: 49-58 (2001)
22. A survey of approaches to automatic schema matching. Rahm E. and Bernstein P. A. *The VLDB Journal — The International Journal on Very Large Data Bases* 10(4): 334-350, (2001).