PowerAqua: supporting users in querying and exploring the Semantic Web

Editor: Philipp Cimiano, Universität Bielefeld, Germany
Solicited reviews: Danica Damljanovic, University of Sheffield, UK; Jorge Gracia del Río, Universidad Politécnica de Madrid, Spain; Haofen Wang, Shanghai Jiao Tong University, China

Vanessa Lopez, Miriam Fernández, Enrico Motta, Nico Stieler
{v.lopez, m.fernandez, e.motta, n.stieler}@open.ac.uk
KMi. The Open University.
Walton Hall, Milton Keynes MK76AA, United Kingdom.

Abstract. With the continued growth of online semantic information, the processes of searching and managing this massive scale and heterogeneous content have become increasingly challenging. In this work, we present PowerAqua, an ontology-based Question Answering system that is able to answer queries by locating and integrating information, which can be distributed across heterogeneous semantic resources. We provide a complete overview of the system including: the research challenges that it addresses, its architecture, the evaluations that have been conducted to test it, and an in-depth discussion showing how PowerAqua effectively supports users in querying and exploring Semantic Web content.

Keywords: Question Answering, Linked Data, Semantic Web, Natural Language

1. Introduction

With the emergence of initiatives like the Linked Open Data (LOD) [2] and the current interest of the major commercial search engines, Yahoo! SearchMonkey[1] or Google Rich Snippets[2], in the exploitation of Semantic Web (SW) content, the amount of metadata available on the Web has significantly increased in the last few years. This metadata has been generated by means of rich semantic resources, such as FreeBase[3] or DBpedia [1], by opening up large datasets previously hidden under backend databases, like the one released by the data.gov[4] initiative, or by encouraging publishers to annotate their own Web content using RDFa[5], or Microformats[6]. In a recent publication[7], Google declared that currently only 5% of Web pages have some semantic markup, however they predict this number will rise soon to 50%.

Although this data growth opens new opportunities for SW applications, the diversity and massive volume currently reached by the publicly available semantic information introduces a new research question: *how can we support end users in querying and exploring this novel, massive and heterogeneous, structured information space?.*

The current approaches that have attempted to address this question suffer from one or more of the following limitations:

a) Limited support for expressing queries, usually at the level of keyword-based search. For example, popular SW gateways like Swoogle[8], Watson[9], or Sindice[10] can find ontology items representing *actors*

[1] http://developer.yahoo.com/searchmonkey/
[2] http://www.google.com/webmasters/tools/richsnippets
[3] FreeBase: http://www.freebase.com/
[4] http://data.gov.uk/
[5] http://http://www.w3.org/TR/xhtml-rdfa-primer/
[6] http://microformats.org/

[7] Google Semantic Technologies invited talk, San Francisco, 2010: http://www.readwriteweb.com/archives/google_semantic_web_push_rich_snippets_usage_grow.php
[8] Swoogle: http://swoogle.umbc.edu
[9] Watson: http://watson.kmi.open.ac.uk
[10] Sindice: http://sindice.com

or *titanic*, but cannot answer the query "which British actors act in Titanic?".

b) A narrow search scope. In particular, *closed-domain* approaches [8-11] assume that the knowledge is encoded in one, or a subset of, pre-selected homogeneous Knowledge Bases (KBs).

c) Limited ability to cope with the ambiguity inherent in user queries. As a result such systems require the users to disambiguate between different interpretations of their input or alternatively suffer from low levels of precision, relying on the user to filter out incorrect answers [6-7].

In this paper, we present PowerAqua [5], an ontology-based Question Answering (QA) system that, in contrast to the previously mentioned approaches: 1) offers a NL query interface that balances usability and higher expressivity - usability studies [3] have demonstrated that casual users prefer the use of Natural Language (NL) queries over keywords when querying a semantic information space, 2) is able to answer queries by locating and integrating information, which can be distributed across heterogeneous semantic resources. To this purpose, PowerAqua supports query disambiguation, knowledge fusion (to aggregate similar or partial answers), and ranking mechanisms, to identify the most accurate answers to queries.

The rest of the paper is organized as follows. Section 2 describes related work. Section 3 describes the research challenges addressed by PowerAqua. Section 4 introduces the global architecture of the system. The conducted experiments and results are reported in Section 5. Conclusions and future work are presented in Section 6.

2. Related Work

An overview of related work shows a wide range of approaches and tools, with different affordances, that have attempted to support end users in querying and exploring the publicly available SW information. Here we give a brief overview of these approaches, their limitations, and how PowerAqua has attempted to overcome them.

One main dimension by which these approaches can be classified is their scope, i.e., up to which level (partial or total) they are able to exploit the publicly available SW content.

At the first level we can distinguish the so-called **closed-domain** approaches, whose scope is limited to one a-priori selected domain or ontology at a time. A representative subset of these approaches is the ontology-based QA systems [8-11], [24], which exploit the semantic information of an underlying ontology to drive and/or to give meaning to the queries expressed by a user. While these approaches have proved to work well in semantic intranets, where a pre-defined domain ontology (or a set of them) is often used to provide a homogeneous encoding of organizational data, there are no reported results, that we are aware of, concerning the feasibility or use of these systems in an open domain scenario, where a massive, heterogeneous set of semantic information should be covered. A new layer of complexity arises because of the "openness" of the scenario, as mentioned in the challenges in Section 3.

It is important to highlight that ontology-based QA systems emerged as a combination of ideas, and as an attempt to enhance the limitations of two different research areas: Natural Language Interfaces to Data Bases (NLIDB) and QA over free text. NLIDB approaches, as well as ontology-based QA systems, are focused on the exploitation of structured data in closed-domain scenarios. However, ontology-based QA systems are able to handle a much more expressive and structured search space, where, as opposed to a DB, the information is highly interconnected by the use of explicit relations. Thus, the knowledge and semantics encoded in an ontology, together with the use of domain-independent linguistic and lexical resources, are the primary sources for understanding the user query, as such, these systems are practically ontology independent [3]. On the other hand, QA over free text, which is a strong and well-founded research area stimulated since 1999 by the TREC QA track, is able to perform QA in open domain environments. However, as stated in [12], the pitfalls of QA over free text, with respect to ontology-based QA approaches arise when a correct answer is unlikely to be available in one document, but must be assembled by aggregating answers from multiple ones, and when the questions are not trivial to interpret.

At the second level, and enhancing the scope embraced by closed-domain models, we can distinguish those approaches **restricted to their own semantic resources**. Currently popular examples of these systems are: Powerset, Wolfram Alpha, or TrueKnowledge[11]. Although these approaches obtain structured answers in an open domain scenario they are restricted to the use of their own semi-automatically built KBs. For example, the Wolfram Alpha knowledge inference engine builds and queries its own large KB about the world (storing more than 10TBs),

[11] http://www.powerset.com/, http://www.wolframalpha.com/ and http://www.trueknowledge.com/ respectively.

while True Knowledge relies on users to add and curate its KB, and PowerSet relies on Freebase.

At the third level, we can highlight the **Open Linked Data (LOD) search approaches**. These systems are not limited by closed-domain or homogeneous scenarios, neither by their own resources, but provide a much wider scope, attempting to cover the majority of publicly available semantic knowledge. Examples of these approaches are: a) the work of Meij et al. [22], which uses DBpedia as a source for a query completion component on the Yahoo search engine, b) the second prize winner of the billion triple challenge (BTC) in 2008, SearchWebDB [6], which offers a keyword-based query interface to data sources available in the BTC datasets, c) the eRDF infrastructure [7], which explores online semantic knowledge by querying live SPARQL endpoints and, d) The mash-up Sig.ma (http://sig.ma), which is able to aggregate heterogeneous data obtained from the search engine Sindice about a given keyword.

While these applications present a much wider scope, scaling to the large amounts of available semantic data, they perform a shallow exploitation of this information: a) they do not perform semantic disambiguation, but do need users to select among possible query interpretations [6,7], b) they do not discover mappings between data sources on the fly, but need to pre-compute them beforehand [6] and c) they do not generally provide knowledge fusion and ranking mechanisms to improve the accuracy of the information retrieved to the users.

Aiming to go one step beyond the state of the art, unlike the previously presented closed-domain applications and approaches that rely on their own semantic resources, PowerAqua is not limited by the single-ontology assumption, it does not impose any pre-selection or pre-construction of semantic knowledge, but rather explores and scales to the increasing number of multiple, heterogeneous sources autonomously created on the Web[12]. In addition, and attempting to overcome the limitations of LOD search approaches, PowerAqua has developed sophisticated, syntactic, semantic and contextual information processing mechanisms that allow a deep exploitation of the available semantic information space. Thus, PowerAqua can answer queries by composing information from multiple heterogeneous semantic sources of varying quality across domains.

In the next sections we will introduce PowerAqua's research challenges and architecture to explain in more detail how this system attempts to overcome the limitations of current approaches.

3. PowerAqua: The Research Challenges

PowerAqua evolved from AquaLog [16], an ontology-based QA system for intranets, limited to the use of one ontology at a time. It was first envisioned in 2006 [17] in the context of a paradigm shift from the first generation of closed-domain semantic systems, to the next generation of open SW applications, able to exploit the increasing amounts of semantic data. Opening up to a multi-ontology scenario brought several important research challenges:

1. *Finding the relevant ontologies to answer the user's query*. In an open domain scenario it is not possible to determine in advance which ontologies will be relevant to answer the user's information needs.
2. *Identifying semantically sound mappings*. User queries can be mapped over several ontologies. In the case of ambiguity, the correct interpretation of the given term in the context of the user query should be returned.
3. *Composing heterogeneous information*. Answering queries may require fusing information from multiple sources. Composite translations and partial answers from different ontologies need to be combined and ranked to retrieve accurate results. Among other things, this requires the ability to recognize whether two instances from different information sources may refer to the same individual.

In addition, the emergence of the LOD initiatives has increased the number of large datasets available on the SW, at the same time bringing additional challenges [5] that PowerAqua needs to address to effectively support users in querying and exploring the current SW:

4. *Scalability.* As a result of the LOD initiative, scale is not only related to the number of ontologies on the SW, but also to their size. These large datasets can potentially cover a wide range of user queries, thus making it more difficult for PowerAqua to focus quickly on a few ontologies with high discriminatory power.

[12] Given that PowerAqua accesses the SW through the Watson SW Gateway, in practice PowerAqua will only retrieve information if this has been crawled and indexed by Watson or in specified online repositories.

5. *Higher Heterogeneity*. The LOD initiative has also caused a shift from the exploitation of small domain ontologies to the exploitation of large generic ontologies covering a variety of domains. As a result, heterogeneity is not only arising from the use of different ontologies, but also within the same ontology. As argued in [23], ontology-based QA systems in restricted domains can tackle the answer-retrieval problem by means of an internal unambiguous knowledge representation. However, in open-domain scenarios, or when using open-domain large ontologies, as is the case of DBpedia, systems face the problem of polysemous words (and multiple interpretations), which are usually unambiguous in restricted domains.
6. *Dealing with noisy and incomplete data*, including: modelling errors, lack of domain and range information for properties, undefined entity types, complex semantic entity labels, redundant entities within the same dataset (e.g., birthplace and placeofbirth), etc.

The really challenging aspects of these Linked Datasets appears to be the combination of scale with heterogeneity and noise, which can lead to many alternative translations of a query, from which the most accurate answers need to be extracted at run time. For example let us consider the query "Give me English actors that act in Titanic". DBpedia contains a huge number of potential ontological hits for one or more of the terms in the user query (even in those cases where the answer to the user query is not actually contained in DBpedia). In particular, although DBpedia contains several mappings for *English actors*, *act* and *Titanic* an ontological translation for the user query can only be found by splitting the compound *English actors* in two[13]. The keyword *English* alone produces more than a thousand mappings in DBpedia, which have to be filtered or analyzed to determine or not their relevance (e.g., English language, English people, English channel, English football, England, etc.). Thus, *English actors* is translated into 26 ontological triples formed with various relations (e.g.: residence, ethnicity, location, hometown, etc.), some of them duplicated (birthplace, birthPlace and born) between the class *Actor* and the instances *England* and *English_people*. There are also 25 resultant ontological triples linking the class *Actor* to various instances of *Titanic* (S.O.S Titanic, Titanic 1943 film, Titanic 1953 film, Titanic 1997 film, etc.) through several relations (starring, director, producer, etc.), because the matches for the linguistic relation *act* (the ontological property *act* and various instances, such as *The act* or *Sister act*) turn out not to be relevant when considering the arguments of the query. PowerAqua combines the partial answers to extract the final set of answers (the English actors: Bernard Hill, Ian Holm and Kate Winslet starring in Titanic 1997, Brian Aherne and Ian Holm starring in Titanic 1953 and S.O.S. Titanic respectively), as presented in Figure 1.

[Figure 1 about here]

PowerAqua aims to address the aforementioned challenges, providing a step towards the realization of scalable and effective SW applications, able to deal with the new layers of complexity introduced by the continuous growth of the semantic data.

4. PowerAqua: The architecture

To support users in querying and exploring the SW content, PowerAqua accepts users' queries expressed in NL and retrieves precise answers by dynamically selecting and combining information massively distributed across highly heterogeneous semantic resources. To do so, PowerAqua follows the pipeline architecture described in Figure 2. The set of components and the overall retrieval process can be briefly summarised in the following steps:

[Figure 2 about here]

1. The *Linguistic Component* performs a linguistic processing of the query to identify the associations between the query terms. The output of this module is a set of Query-Triples <subject, predicate, object>, which map the user's request into a linguistic triple-based representation (Section 4.2).
2. *The Element Mapping Component (PowerMap)* is composed by two sub-components. The *Ontology discovery component* identifies those semantic resources that may be relevant to the user query. This initial match is performed by means of syntactic techniques that, in many cases, generate several possible candidate semantic entities, which may provide potential alternative interpretations for a query term. To address this problem, *the Semantic Validation Component* builds on Word

[13] "English actor" is the exact label for several DBpedia instances of actors, none of them starring in Titanic.

Sense Disambiguation techniques to disambiguate between different possible interpretations of the same query term across ontologies (Section 4.3).

3. *The Triple Mapping Component (Triple Similarity Service -TSS)* makes use of the query context, formalized in step 1, and the ontological context surrounding the entity candidates, obtained in step 2, to determine the most likely interpretation of a user query as whole. This is done by extracting the set of ontology triples that better match, partially or completely, the set of linguistic triples that represent the user information needs. Several filtering heuristics are integrated in this component to limit the set of candidates for computational expensive queries (Section 4.4).
4. *The Merging and Ranking Component* composes precise answers by integrating the set of ontological facts (triple patterns), recovered in the previous step from multiple semantic resources. Once the different facts are merged and the answers (the list of semantic entities that comply with the facts) are composed, this component applies a set of ranking criteria to sort the list of results (Section 4.5).

Each of the aforementioned components can be considered a research contribution on its own. In this section we first describe the technical infrastructure of PowerAqua, which includes plugins for several semantic storage platforms to collect and provide fast access to the semantic information (Section 4.1). Secondly, in the next Sections we detail each of the query processing components and associate them with the research challenges that each of them aims to address. To illustrate the functionality of each component and to give a comprehensive account on the way the system returns answers to queries we will follow the illustrative query example: "*Give me actors starring in movies directed by Clint Eastwood*".

4.1. Semantic Storage Platform

PowerAqua provides a plugin specification mechanism that supports a common API to manipulate content independently of the storage platform, knowledge representation language, and location. As a result, PowerAqua provides unified access to multiple distributed semantic repositories and keeps the query processing and algorithms independent of the underlying infrastructure.

Plugins are loaded on demand. Given a user query, each potentially relevant ontology is dynamically associated to an instantiation of a plug-in containing all the connection information needed to access the online ontology (ontology identifier, language, its corresponding framework and location). This flexible infrastructure has allowed us over time to integrate more efficient query platforms into PowerAqua, without being restricted to using just one given search engine or platform. Currently we have plugins for a) the Watson SW gateway, which provides an API to query data at run time[14], b) Virtuoso[15] and, c) Sesame, versions 1 and 2[16]. While the Watson SW gateway is used as the main window to access the online semantic information, Virtuoso and Sesame are used to store and access selected datasets that are not currently available in Watson due to their size and format, e.g., some of the datasets offered by the LOD community[17]. Different extensions of the implementations are done to encapsulate the different ontology frameworks: the functionality of the plug-in is implemented through SeRQL queries in the case of Sesame and SPARQL queries for a Virtuoso SPARQL end-point, while in the case of Watson the plug-in functionality is implemented on the top of calls to the Watson API.

In a scenario where the user may need to interact with thousands of semantic documents structured according to hundreds of ontologies, full text index searches are required to manage such amounts of information in real time. However, unlike the Watson and Virtuoso platforms, which implement their own indexing mechanisms, in the case of Sesame, the plugin is extended with an offline ontology indexing module based on Lucene[18].

4.2. The Linguistic Component

The purpose of the linguistic component is to perform a NL processing of the query and map the user terminology into a triple-based representation where the interdependencies between query terms are identified and formalized. This purely linguistic representation of the query, with no correspondence with any

[14] Other search engines, like Swoogle or Sindice, adopt a web view of the SW. They support keyword search but fail to exploit the semantic nature of the content they store and therefore, are still rather limited to dynamically exploit online ontologies.

[15] Virtuoso: http://virtuoso.openlinksw.com

[16] Sesame: http://www.openrdf.org

[17] http://esw.w3.org/TaskForces/CommunityProjects/LinkingOpenData/DataSets

[18] http://lucene.apache.org/java/docs/index.html

ontology, facilitates the exploitation of the query context by the subsequent components.

The triple-based representation, also called linguistic Query Triple (QT) representation, is defined by the following structure: <subject, predicate, object>, which formalizes a relationship between the query terms where they can adopt the roles of subject, predicate or object. In the case of ambiguity, i.e., when more than one query term can fulfil the same role, the ambiguous role is represented in its QT, splitting the candidate query terms by the symbol "/", e.g., <subject$_1$ / subject$_2$, predicate, object>. If the query term of a specific role is unknown, this situation is specified in the QT by means of the symbol, "?", e.g., <subject, ?, object>.

Following our example, the NL query "*Give me actors starring in movies directed by Clint Eastwood?*" is transformed into the following set of QTs: {QT$_1$: <actors, starring, movies> & QT$_{2:}$ <actors/movies, directed, Clint Eastwood>}. As we can see, an ambiguity is specified for the role subject of QT$_2$, since both query terms, actors and movies, can fulfil the role.

To process the query and extract this representation, PowerAqua's linguistic component makes use of the Gate NL processing tool [18]. Using this tool, the component is able to identify factual queries formed with wh-terms (which, what, who, when, where) or commands (give, list, show, tell, etc.) which vary in length and complexity: from simple queries, with adjunct structures or modifiers, to complex queries with relative sentences and conjunctions /disjunctions. Negations, comparatives, superlatives existential, or queries involving circumstantial (why?) or temporal reasoning (last week, in the 80's, between the year 94 and 95) are currently out of the scope of the linguistic component. More details about how these query types are identified and processed can be found in [16].

4.3. The Element Mapping Component: PowerMap

PowerMap's main goal is to discover those resources that may contain a complete or partial answer to the user's query. To do so, PowerMap makes sequential use of two components: the *Ontology Discovery Component* and the *Semantic Validation Component.*

Addressing the first of the challenges identified in Section 3, the *Ontology Discovery Component* makes use of the semantic storage platforms coupled with PowerAqua (Section 4.1) to extract a rough set of semantic resources which may contain the information requested by the user. To do so, it performs an initial syntactic matching at element level, between the query terms, expanded with their lexically related words obtained from WordNet (WN)[19] (synonyms, hypernyms), and the SW itself as a source of background knowledge (using the *owl: sameAs* property of the matched semantic entities).

In many cases this initial syntactic match will generate several possible candidates (i.e., semantic entities), which may provide potential alternative interpretations for a query term. For instance, let's consider the query "*groups that play rock*". Here PowerAqua will find entity mappings in ontologies describing rock as stone or aggregated minerals (rock#1) and ontologies describing rock as a music genre (rock#2).

To identify the different semantic interpretations for the same query term, addressing the second identified challenge (Section 3), PowerAqua makes use of the *Semantic Validation Component.* This component builds on techniques developed in the Word Sense Disambiguation community [27], and exploits the background knowledge provided by WN and the context surrounding the candidate entities in their ontologies to disambiguate between different possible interpretations of the same query term within an ontology or across several ones.

Following our example, to compute the meaning of the ambiguous term (*rock*) in each of the ontologies that previously produced a mapping during the ontology discovery phase, PowerAqua computes a WN-based semantic similarity measure between the term and its ascendants in the ontology (e.g., in the Stanford TAP ontology, the ascendant or direct superclass of *rock* is the concept *genre*). This semantic similarity is computed considering: a function of the path distance between the synsets associated with the ambiguous term (e.g., *rock*) and the synsets associated to its ascendant in the ontology (e.g., *genre*) and, to which extent these synsets share information in common in the is-a hierarchy of WN. E.g., if we compute the similarity measure for the TAP ontology, we will see that the highest value for this measure is obtained by the path of distance two: *rock*#1 *>popular_music_genre >genre*, where the rock#1 synset and the genre synset share in common five elements in the is-a hierarchy of WN: *genre >music >auditory_comunication >abstraction >entity*. Therefore the assigned meaning of the term *rock* for the TAP ontology refers to music genre (rock#1). While in the ATO ontology, where the ontology ancestor of *rock* is *substance,* the highest value of this measure is ob-

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

tained using the synset of *rock* associated to the WN meaning stone (rock#2), whose shorter path is *rock#2 >material >substance and* the common elements that subsume them both are: *substance >matter >physical_entity >entity*. Our empirical tests so far validate our approach as being able to cope with the ambiguity inherent to multiple ontologies with a good degree of accuracy. The reader is referred to [15][17] for concrete details of the disambiguation algorithms and its evaluation.

The output of PowerMap is represented as a set of tables, where each table contains its corresponding QT term and its corresponding set of mapped semantic entities (and their synsets) found on the SW. These tables are known as *Entity Mapping Tables* (ETMs). In our example, the EMT for the query term "actors" contains, among several others, exact candidate matches in DBpedia, the movie database[20] and TAP ontologies, and hypernym matches for "person" in SWETO and the SW conference ontology.

4.4. The Triple Mapping Component: TSS

Once the most suitable subset of semantic resources from which a query can be answered has been identified, the *TSS* (triple similarity service) explores the context of the user query, formalized as QTs (Section 4.2), and the ontological context surrounding the semantic entity candidates (Section 4.3), to determine the most likely interpretation of a user query as a whole and the individual terms in the query. By exploring the ontological relationships of the candidate entity mappings, the TSS assembles the element level matches, obtained by the *Element Mapping Component*, and identifies the Ontology Triples (OTs) that better match, partially or completely, with the set of QTs that represent the user information needs.

Following our example "*Give me actors starring in movies directed by Clint Eastwood"*, the QT_1 <actors, starring, movies> is mapped by the TSS with the OTs <actor, starring, film> and <actor, starring, American_movie> found in DBpedia, and <Actor, participates, Movie> or <Actor, plays, Movie> in the movie database ontology. The QT_2 <actors/movies, directed, Clint Eastwood > is mapped to the OT <film, director, Clint Eastwood>, and the OT chain (assigned to a lower rank position, see Section 4.5) [<actor, occupation, place> <place, birthplace, Clint Eastwood>], both found in DBpedia[21].

PowerAqua's set of answers is then obtained by extracting the list of semantic entities that fulfil the OT patterns. These answers are later composed and ranked by the *Merging and Ranking Component* (Section 4.5) to obtain the final list of results. More details about the TSS can be found in [19].

Obtaining the set of OTs from which answers can be derived is a complex and costly procedure. In order to maintain real time performance in a scenario of perpetual change and growth (challenge 4, Section 3), the TSS explores the simplest techniques in the first place and progressively makes use of more expensive mechanisms to retain a good level of recall when the simple techniques do not provide answers. Specifically:

- The TSS selects first the ontologies that better cover the user query, i.e., ontologies providing entity candidates for two or more terms of a given QT are preferred over ontologies providing only one candidate.
- The TSS selects first entity mappings covering a compound QT, e.g., *Clint Eastwood*, over entity mappings covering just individual parts of the compound QT.
- The TSS searches first for direct ontological relationships between the candidate entities. Indirect relationships, i.e., relationships that require two OTs to be joined in one mediating concept, are only explored if no direct relations are found.

In addition to time performance optimization, the TSS has to confront the high levels of heterogeneity and noise present in some of the available semantic resources (challenges 5 and 6, Section 3). E.g., for the query term movie, DBpedia provides more than a thousand mappings, including: the class *film*, the properties: *film, movies, show,* and various instances: *Sky_Movies, MTV_Movie_Awards, AmericanMovie, FilmFestival, FilmAward*, etc. To limit the space of solutions provided by these resources, PowerAqua has implemented a set of filtering heuristics [5], among which we can highlight:

- Within an ontology, the TSS selects the exact mappings, if any, over approximate mappings for the same term and entity type, e.g., the exact class *Film* is preferred over the class *FilmFestival* as a matching for the synonym term *Film*. Furthermore, if there is not a valid mapped relation in the

[20] http://data.linkedmdb.org/

[21] OTs contain actual namespaces such as dbpedia:actor, while QTs would be just 'actor' as referring to the word which is not yet linked to any ontology resource.

ontology for the candidate arguments, the TSS search for relations among entity pairs in this order: pairs formed with at least one exact mapping over pairs formed using only approximate, synonym or hypernym mappings. In our example, the TSS selects the DBpedia mapping for the term *actor*, over the DBpedia mapping obtained with its hypernym *person* to look for relations between *actor* (exact) and *movie* (synonym).

- The TSS eliminates ambiguous mappings by exploiting the query context. For example, for the QT <*rock, ?, musician*> extracted from the query "Rock musicians in Britain", the TSS is able to discard the ontologies providing element mappings of the term rock as a stone, since these ontologies are not likely to provide any triple mappings with the term rock (as a stone) associated with the QT term musician.

These heuristics cannot be based on very specific assumptions about the semantic resources, since some of these resources are noisy or incomplete (e.g., the exact mapping for *movies* does not lead to any answer, while the synonym *film* does).

4.5. The Merging and Ranking Component

A major challenge faced by PowerAqua (challenge 3, Section 3) is that answers to a query may need to be derived from different ontological facts and even different semantic sources. Depending on the complexity of the query, i.e., the number of its corresponding QTs, as well as the way each QT is matched to OTs, these individual answers can: a) be redundant, b) be part of a composite answer to the entire query (*intersection-based queries*) or c) be alternative answers derived from different ontological interpretations of the QTs, (*union-based queries*). Hence, different merging scenarios may arise depending on how the terms are linked across OTs.

Following our example in Section 4.4, to answer QT_1, <actors, starring, movies>, a set of actors and the movies in which they starred are obtained using the two OTs in the DBpedia and movie database ontologies. To answer QT_2, <actors/movies, directed, Clint Eastwood>, a set of movies directed by Clint Eastwood is extracted from DBpedia, using the first OT formed with the WN synonym class *Film,* the WN derived property *director* and the instance *ClintEastwood.* This OT is ranked higher (on the basis of a confidence ranking algorithm which sorts the ontological facts) than the less accurate DBpedia indirect OT chain, formed with ad-hoc relationships between the matched concepts *actor*, *place* (a synonym of *directed*) and the instance *ClintEastwood* . To obtain the final set of responses, in a first step, the equivalent entities of actors and movies returned by DBpedia and movie database are identified and merged (union of answers) to avoid redundancy. In a second step, the answers of each QT are intersected based on the common movies. This intersection leads to a final response of 35 actors, including: John Cusack, Laura Linney, Kevin Bacon.

Once the different ontological facts are merged and the answers are composed, this component applies a set of ranking criteria to sort the list of results. These ranking criteria are based on: a) the confidence of the mapping algorithm on the ontological facts from which the answer is derived, b) the confidence of the disambiguation algorithm about the interpretation of the answer, and c) the confidence of the merging algorithm, also called *popularity* of the answer.

The confidence of the mapping algorithm is based on the fuzziness of the mapping at element level, i.e., if the mapping has been extracted using the original query term or by means of any of its synonyms or hypernyms, (see Section 4.3, *Ontology Discovery Component*) and the fuzziness of the mapping at triple level, i.e., how well the OT from which the answer is extracted covers the information specified in the QT, (see Section 4.4, *The Triple Mapping Component*), as previously shown for our example query.

The confidence of the disambiguation algorithm is based on the popularity of the different interpretations of a query term. For those cases in which alternative interpretations of the same query term are identified, e.g., rock as stone and rock as music genre, PowerAqua computes how many ontologies with answers contain each particular interpretation (see Section 4.3, the *Semantic Validation Component*). Answers obtained by means of the most popular semantic meaning, i.e., the one appearing in a higher number of ontologies, are ranked first.

The popularity of the answer, computed by the merging algorithm, refers to the number of ontologies from which the answer has been extracted. Popular answers are prioritised over non-popular ones.

While for testing purposes, the ranking criteria can be individually selected and configured, these criteria can also be subsequently combined using the following order: a), b), c). More details about the merging and ranking algorithms can be found in [4].

5. Evaluation and Results

In contrast with the Information Retrieval (IR) community, where evaluation using standardized techniques, such as those used for the annual TREC competitions, has been common for decades, the SW community is still a long way from defining standard evaluation benchmarks to evaluate the quality of semantic technologies [20]. Important efforts have been made in the last few years towards the establishment of common datasets, methodologies and metrics to evaluate semantic technologies, e.g., the SEALS project [26]. However, the diversity of semantic technologies and the lack of uniformity in the construction and exploitation of the data sources are some of the main reasons why there is still not a general adoption of evaluation methods. Evaluating PowerAqua, constitutes a major research challenge, not just because of this lack of standard evaluation datasets, but because of the inherent difficulties in defining evaluation methodologies able to assess the quality of its cross-ontology search capabilities.

The **aim of our evaluations** is to probe the feasibility of performing QA in an open SW scenario, even in its current form, defined by multiple heterogeneous semantic sources and Linked Data datasets. Scalability is still a major open issue; although we have experimented with multiple semantic storage platforms, more work needs to be done on the back end infrastructure to cover not just a subset but all semantic data available. Nonetheless, regarding the querying process we believe that the results obtained from our experiments can be extrapolated to a large proportion of semantic tools that wish to retrieve, use and combine large, multi-domain semantic data on the fly.

In this section we present the six main evaluations conducted to test PowerAqua. For each evaluation we report: a) the context in which the evaluation was conducted, b) the evaluation set up and measures used and, c) the lessons learned. Among them, the latest PowerAqua's evaluations focus on assessing the performance of its algorithms using different semantic storage platforms and on usability. All of the mentioned datasets, evaluation results and an online demo can be found at:

http://technologies.kmi.open.ac.uk/poweraqua/.

5.1. Evaluating PowerAqua by reusing standard IR evaluation benchmarks

Evaluation Context: This evaluation study [16] was performed at a stage when the SW had expanded, offering a wealth of semantic data that could be used for experimental purposes, therefore allowing the first testing of PowerAqua's capabilities in answering questions across multiple ontologies on the SW. Aiming to conduct a large scale and formal evaluation with standard datasets, PowerAqua was evaluated not as a stand-alone system but as a query expansion component of a more complex IR system [21].

Evaluation Setup: The evaluation focused on assessing whether the exploitation of PowerAqua as a query expansion module, provided an improvement in precision over a keyword-based retrieval baseline. A practical advantage in this case was that the evaluation could be conducted using: a) a gold standard from the IR community, the TREC WT10G document collection, b) the queries and judgments from the TREC 9 and TREC 2001 Web track competitions, and c) the standard IR TREC evaluation metrics for search engines: precision and recall. To represent the semantic information space we collected and indexed in the PowerAqua's storage platforms (Section 4.1) around 2GB of metadata comprising different domains.

Lessons Learned: The approach was proposed as a first step aimed to bridge the gap between the SW and the Web. PowerAqua successfully generated a query expansion for 20% of the queries where semantic information was available to cover the queries (degrading gracefully for the queries were semantic data was not available or incomplete), leading to important improvements over the purely keyword-based baseline approach in 85% of the evaluated queries. Although from an IR perspective, the experiment was only able to cover 20% of the queries, in the context of the growth of the SW, this experiment can be judged as a real milestone. For the first time the input semantic data is heterogeneous and representative, while the queries and success criteria are externally sourced and independently built in the context of an international benchmark.

5.2. Evaluating PowerAqua's individual components by means of user-centric methodologies

Evaluation Context: The complexity and scale of the evaluation study reported in Section 5.1 meant that it was not viable to analyze in detail specific PowerAqua limitations. For this purpose, and aiming to test PowerAqua's competence to answer queries in real time from multiple distributed information sources, we conducted a user-centric evaluation [19].

Evaluation Setup: To make the experiments reproducible and to simplify the task of the question designers (to generate NL queries from the semantic sources in a given collection), we collected 700 semantic documents distributed in 130 repositories, which provided around 3GBs of metadata. A total of 69 queries were generated by 7 users, familiar with the SW, who were asked to generate factual questions that were covered by at least one ontology of the semantic information space. We measure overall accuracy, which is the percentage of questions that are answered correctly. As we ensured that there was at least one ontology covering each query, if PowerAqua was not able to find any answer or the answer was incorrect, it was considered a failure. The set of failures were then analyzed and divided into four categories according to the component that led to the error: a) the Linguistic Component, b) PowerMap and, c) the TSS. The merging and ranking component was under development when this experiment was conducted. The time to provide an answer for each query was also computed.

Lessons Learned: PowerAqua successfully answered 48 (69.5% of accuracy) out of 69 questions. The failures included: a) performing an incorrect linguistic analysis of the query (7.2.%), b) not finding element mappings, or discarding valid element mappings (18.8%) and, c) incorrectly locating the ontology triples to answer the query (4.3%). The average query answering time was 15.39 seconds, with queries ranging from 0.5 to 79.2 secs. This is because we do not always have enough ontological context to focus on precision when, because of heterogeneity, there are many alternative translations (see the example "Give me English actors that act in Titanic" in Section 3). As main lessons learned, this evaluation highlighted an illustrative sample of problems for any generic algorithm wishing to explore SW sources without making any a priori assumptions about them:

- Firstly, such algorithms are not only challenged because of the scale of the SW but more importantly because of its considerable heterogeneity, as entities are modelled at different levels of granularity and with different degrees of richness.
- Secondly, while the distinctive feature of PowerAqua is its openness to unlimited domains, its potential is overshadowed by the sparseness of the knowledge on the SW. To counter this sparseness, the PowerAqua algorithms maximize recall (e.g., by using lexically related words), which may lead to a decrease in accuracy and an increase in execution time.
- Thirdly, in addition to the sparseness, most of the identified ontologies were barely populated with instance data. This caused PowerAqua's failure to retrieve a concrete answer in some cases even when a correct mapping of the query was found in an ontology.
- A fourth aspect that hampered our system was the existence of many low quality ontologies which contained redundant, unrelated terms, causing the selection of incorrect mappings, discarding relevant ones, or being unable to fill in the missing information in order to fully understand a query due to the lack of range and domain information.
- Finally, as the fifth aspect, we note the yet suboptimal performance of ontology repositories and semantic search platforms to query large datasets. This limits the amount of information PowerAqua can explore in a reasonable amount of time, e.g. searching for indirect relations between entities.

5.3. Evaluating PowerAqua's merging and ranking

Evaluation Context: The merging and ranking capabilities of PowerAqua were still work in progress when carrying out the previous evaluation. For this purpose, an evaluation was conducted and reported in [4], to assess the quality results obtained after the application of the merging and ranking module.

Evaluation Setup: To represent the information space with the purpose of obtaining a representative set of queries, which could be correctly mapped by PowerAqua into several ontological facts, preferably across different ontologies, additional metadata was collected with respect to the previous experiment, up to 4GB, including large ontologies, such as the DBpedia infoboxes[22]. We collected a total of 40 questions, selected from the example queries in the PowerAqua demo website[23] and from the previous PowerAqua evaluations [19], which were complex enough to require merging or ranking in order to obtain accurate and complete answers. As judgments to evaluate the merging and ranking algorithms, two ontology engineers provided a True/False manual evaluation of answers for each query. Precision and recall were selected as evaluation metrics, where precision is the number of correct answers from the total of retrieved answers after applying merging and ranking; and recall is the number of correct answers, after applying the merging and ranking, with respect

[22] A subset of DBpedia which in the 2008 version was only 1GB.

[23] http://poweraqua.open.ac.uk:8080/poweraqualinked/examplestopic.html

the number of correct answers in an scenario were merging and ranking algorithms are not applied.

Lessons and open issues: The results obtained from this evaluation indicated improvements in the quality of the answers with respect to a scenario where the merging and ranking algorithms were not applied. The merging algorithm was able to filter out a significant subset of irrelevant results, and all the ranking algorithms were able to increase the precision of the final set of answers, without significant loss in recall, thus showing a deeper semantic "understanding" of the intent of the question. More specifically, the fusion algorithm (a co-reference algorithm to identify similar instances from different ontologies) exhibited a 94% precision and 93% recall. The merging algorithm was able to filter out up to 91% (32% on average) for union-based queries, and up to 99% (93% on average) for intersection based queries of irrelevant results. Even with the different behavior of these ranking methods (Section 4.5), the combined algorithm is over-performed by the confidence on the mapping ranking in terms of precision, but it is able to improve the precision and recall ratio. The semantic similarity ranking depends on being able to calculate the semantic interpretation of each OT, but that's not the case if the OT entities are not covered in WN, or the taxonomical information is not significant enough to elicit the meaning of the entity in the ontology. The popularity ranking requires fused answers to be obtained from at least two ontologies. We believe that any future growth in the availability of online semantic data will result in direct improvements for both popularity (hampered by knowledge sparseness) and semantic similarity ranking measures (hampered by low quality data). The best ranking algorithm (by confidence of the mapping) was able to obtain an average of 96% precision for union queries and 99% for intersection queries.

An interesting side effect was that answers to some questions that were distributed across ontologies could only be obtained if the partial results were merged. Therefore, the introduction of the merging algorithm provides PowerAqua with the capability to answer queries that cannot be answered when considering a single knowledge source. For example, "which languages are spoken in South American countries?" is answered by combining partial results across two ontologies: languages spoken in any country by DBpedia and countries in South America by the TAP ontology.

5.4. Evaluating PowerAqua's performance when dealing with the scale and heterogeneity of the LOD

Evaluation Context: As mentioned in Section 3, the LOD has defined a turning point in the evolution of the SW and its applications, giving a step towards the exploitation of real-world, massive, heterogeneous and distributed semantic information. The evaluation reported in [5] investigated the feasibility of PowerAqua to scale to this new semantic information space, by introducing one of the largest and most heterogeneous LOD datasets, DBpedia.

Evaluation Setup: The same evaluation set up used for the previous evaluation was used here, in which the biggest source, SWETO[24] is not more than 1GB (over 3 million triples). The only change was the addition of more than 13 GBs of semantic data from DBpedia (in a Sesame repository) to the semantic search space, as a representative LOD dataset.

Lessons and open issues: The average number of valid answers obtained after applying the fusion algorithm, which has a precision of 94% [4], increased from 64 to 370 when the DBpedia dataset was used (as many questions were also answered in DBpedia). In addition the average time to answer a query increased from a total of 32 to 48 secs in average for the same set of queries [5] (54.3 secs if more complex queries, answered only in DBpedia, were added). This increase in the response time is due to two main reasons: a) because of the higher heterogeneity introduced by the new dataset, more complexity is added to the mapping algorithms and, b) because of the suboptimal performance of the semantic storages, where the response time to calls increases for large datasets. This suboptimal behaviour is detected when: searching for relationships between instances of highly populated classes, searching for indirect relations between element mappings and, searching for relations involving literals.

The first problem has been partially addressed by the implementation of filtering heuristics that balance precision and recall (to limit and keep to a reasonable size the space of solutions to be analyzed) in the TSS (see Section 4.4), which reduced the number of SeRQL calls to the repositories by more than 40% (from 587 to 352 average). These heuristics, as well as PowerAqua's iterative approach, which explores the simple solutions first, augmenting the complexity in each re-iteration until an answer is found or all possibilities have been analyzed, aim to keep a good level of recall, while maintaining an acceptable re-

[24] http://knoesis.wright.edu/library/ontologies/sweto/

sponse time. As argued in [23] open-domain QA can benefit from the size of the corpus, as the size increases it becomes more likely that the answer to a specific question can be found. In our scenario, as more semantic data becomes available and the quality of the data improves, it will become easier to find more precise mappings with answers, without requiring a complex mapping algorithms, as long as the back end can efficiently handle the increased scale.

To address the suboptimal performance of the semantic storages, the Virtuoso semantic storage platform has been integrated as a plugin for PowerAqua. The evaluation of this solution is reported in the following section.

5.5. Evaluating PowerAqua's response time when using different semantic storage platforms

5.5.1. Using Virtuoso as semantic storage platform

Evaluation context and set up: Aiming to assess the time performance of PowerAqua when introducing Virtuoso as a new semantic storage platform, we have re-run the evaluation presented in Section 5.4.

[Table 1 about here]

The results of this evaluation can be seen in Table 1. The first column shows a subset of 16 queries used to test the system. The second column shows the performance of PowerAqua before DBpedia was integrated within the semantic search space and using Sesame as the main semantic storage platform. The third column shows the performance of PowerAqua when integrating DBpedia as part of the search space and, the last column shows the performance of PowerAqua when integrating Virtuoso as the main semantic storage platform. As we can see in the table, the average query response time has diminished considerably, from 54 seconds to 20 seconds, a 63%.

Lessons and open issues: This huge decrease in the query response time obtained thanks to the use of Virtuoso is a very positive sign, indicating that the latest solutions for semantic storage can efficiently handle the growth of semantic resources, thus increasing the potential of applications that rely on them, such as PowerAqua. As public sparql end points[25] are also based in Virtuoso, we also implemented a plugin to query them, the Virtuoso plugin we have already described could not be used because currently sparql end points do not expose the SQL port to the public. As a result they have to be accessed by HTTP services[26], rather than through the SQL interface (JDBC) which provided better performance. In addition, network delays and tighter constrains on the web services (e.g., query timeouts, number of users) make the QA process too slow.

5.5.2. Using the Watson SW gateway

Evaluation context and set up: An issue remains nonetheless open in all previous evaluations: the use of our own collected datasets to perform the experiments. The SW community has yet to propose standardized benchmarks to evaluate cross-ontology open-domain QA systems. Despite this fact we have tested our algorithms with a significant amount of distributed semantic metadata of varying levels of quality and trust and different domains. However, we also report in here: (1) a small-scale test to measure the performance of PowerAqua using Watson to access online semantic data, and (2) the lessons learnt from our experiments as part of the *Billion Triple Challenge* (BTC) contest.

In (1) the performance was tested with a set of 27 ad-hoc queries answered by one or more ontologies in Watson. Total recall cannot be measured due to the large size and openness of the scenario, therefore, we obtained the following averages in the aggregated results: 0.77 for precision (the number of correct results from all retrieved results), 0.83 for precision@1 (the number of correct results from the results ranked in first position) and 0.69 for recall@1 (the number of correct results ranked in first position with respect to the total of correct results retrieved, independently of their ranked value). The raking criterion used in here is the combination ranking. It took an average of 27.7 secs to translate the NL query into the OTs and 5.43 secs for fusion, a total 33.1 secs. Thus, we achieved similar response times with Watson in comparison to the previous experiments with online repositories.

In (2) an instance of Watson was produced relying on indexes generated on top of the BTC dataset. We performed optimizations in PowerAqua to obtain a tighter interaction and better performance with Watson, which made it possible for us to compete in the first BTC in 2008. The main modification has been done by reducing the number of candidate mappings returned by Watson using a functionality provided by the Watson API. This functionality allows PowerAqua to restrict the ontological mappings for a given term to the ontologies that also contain mappings for

[25] http://dbpedia.org/sparql

[26] Accessed using Jena arq libraries: http://jena.sourceforge.net/ARQ/cmds.html#arq.remote

another given term (e.g., looking for the subject, or property, of a query only in ontologies which contains also mappings for the object of the query, considering any of their lexical variations). More details are given in [25].

Lessons and open issues: PowerAqua selects the ontologies relevant for a user query on the fly as part of the querying process. The main advantage of Watson is that it provides an infrastructure to automatically discovering ontologies in the SW with zero cost (PowerAqua can find answers from any of the datasets crawled by Watson without being previously aware of them). However, semantic sources in the open Web appear to have many quality issues. The size and quality of ontologies found in Watson, which includes a large number of small, lightweight ontologies (often not populated and not fit for QA purposes) and foaf files, is lower than those added in our repositories. The semantic data is often duplicated, noisy, or it does not have a schema associated to it (an ontology split into different files that are not recognized as part of the same graph). Quality issues added to the scale of the BTC corpus hampers the performance of the system to find answers.

PowerAqua algorithms were originally developed to work on a sparsely populated SW and designed to maximize recall (by augmenting the search space). However, as the number of SW sources increases (like when using Watson to provide access to the billion triple data), this approach is not longer effective, and heuristics that balance precision and recall are used to be able to prune the search.

5.6. SEALS campaign usability study

Evaluation context and set up: While the previous evaluations focused on accuracy and performance, we present here the first usability results of PowerAqua as a NL interface to semantic repositories. The evaluation was carried out following the formal benchmark proposed for the SEALS 2010 semantic search evaluation campaign, and focused on the interface and usability aspects of each different search tool (in particular keyword-based, form-based and NL). This evaluation cannot assess PowerAqua's ability to query multiple ontologies and fuse answers across sources, but it can compare different interfaces within a user-based study in a controlled scenario. 10 human subjects were given 20 tasks (questions) to solve using the *Mooney* geography dataset, a range of user-centric metrics, such as the time and number of attempts required to obtain an answer of a system, were collected. In addition, data regarding the user's impression of the tool is gathered using the System Usability Scale (SUS) questionnaire [28]. More details are given in [26].

Lessons and open issues: The results are positive, giving an important insight on the usability of PowerAqua. PowerAqua SUS score was 72.25, which is consistent with the number of attempts the users required to formulate the query (2 attempts in average) and their degree of satisfaction. SUS scores have a range of 0 to 100, a score of around 60 and above is generally considered as an indicator of good usability. PowerAqua was the system with the highest SUS score and where the users found the highest number of satisfactory answers and with the best precision (0.57) and recall (0.68)[27]. Still precision and recall values do not give the full picture, because of the presence of complex queries that enclose multiple concepts, modifiers and conjunctions, including comparatives, superlatives and negations, which are out of PowerAqua coverage. For each task (question) the users could formulate the questions themselves. Since a number of the tested tasks (questions) had a high complexity level, the cognitive process of the user for this kind of tasks, which require them to formulate various questions in order to get an answer, cannot be captured in terms of precision and recall. Also, a limitation of this evaluation is that it can only elicit usability measures of a system in a controlled scenario where a number of users (10 in our case) are given a number of tasks (relatively complex queries from the linguistic point of view but formulated according to the structure and vocabulary use in the ontology) to solve using the Mooney geography dataset, rather than measuring the system's ability to solve open-ended real user queries.

6. Conclusions

PowerAqua tackles the problem of supporting users in querying and exploring information across multiple and heterogeneous SW sources. PowerAqua's main contribution with respect to the state of the art is to effectively exploit and combine large amounts of distributed and heterogeneous SW resources to drive, interpret and answer the users' requests. This represents a considerable advance with respect to other systems, which either restrict their scope to an ontology-specific or homogeneous fraction of all the

[27] Campaign 2010 results at: http://www.seals-project.eu/seals-evaluation-campaigns/semantic-search-tools/results-2010

publicly available SW content, or perform a shallow exploitation of it.

Despite the challenges emerged with the development of the SW and LOD (scalability, heterogeneity, inaccurate semantic information, etc.), the conducted experiments have shown the capabilities of PowerAqua, to provide accurate responses to answer the users' requests from massively distributed SW content. Each of the six evaluations presented here allow us to extract useful lessons and open issues for developers in the wider SW community:

1) PowerAqua evaluation as a query expansion module, reusing an IR evaluation benchmark, highlighted the potential of using SW information to enhance searches on the Web, but also the sparseness and incompleteness of the SW when compared to the Web [13].
2) In the evaluation of PowerAqua's ability to map a NL query into several ontologies, we found that PowerAqua was able to answer correctly more than half of the queries (a positive result considering the openness of the scenario). The evaluation highlighted that most of the failures were due to lexical level issues originated as consequence of the high levels of heterogeneity combined with poorly modeled and incomplete or barely populated ontologies.
3) The merging and ranking evaluation showed an improvement in the quality of answers. Besides obtaining more accurate integrated answers to questions by exploiting the increasing amount of collectively authored, highly heterogeneous, semantic data, it allows PowerAqua to answer user's requests that extend beyond the coverage of single datasets and build across ontological statements from different sources. The confidence of the mapping algorithm was the best ranking measure; semantic similarity and popularity ranking measures were hampered by the sparseness and incompleteness of data on the SW.
4) The evaluation on scalability shows that PowerAqua's response time increases when a large semantic source such as DBpedia is added. The reasons behind the decrease in speed are not so much because of the increase in the number of resultant hits, obtained when querying more and larger heterogeneous repositories. Heuristics that balance precision and recall keep the number of mappings and queries to the semantic sources more or less constant, even when adding large semantic sources or a large number of them (although this heuristics cannot be too strict due to the higher heterogeneity and noise of the datasets). The main reason is because of the increase of query response times in the semantic storage for large datasets.
5) We experimented with different storage platforms to show the increase in PowerAqua's performance with the evolution of semantic storage platforms. Balancing the complexity of the querying process and the amount of semantic data is still an open problem in general. However, since PowerAqua is based on external semantic storage platforms, its scalability is also conditioned by the evolution of these platforms towards the efficient response to the growth of the semantic sources.
6) The usability study showed that despite PowerAqua's still limited linguistic coverage and the habitability problem typical of NL interfaces [3] (the user requires a bit of familiarization with the system to know what is possible to ask: the coverage of the system and of the underlying data), users like the flexibility of being able to pose NL queries.

Performance and scalability issues remain nonetheless open. As future work, basing our premises in the continuous growth of semantic data, we aim to focus on the development of algorithms that help to improve the precision of answers retrieved by PowerAqua, leaving recall as a secondary goal since, as indicated in our experiments (Section 5.4), we expect recall to grow in line with the growth of available semantic content. Of course, as the size of the SW increases, additional experimental evaluations will be needed to locate the optimal trade-off between recall and precision. Finally, we also aim to carry out further experiments in integrating PowerAqua with standard IR approaches, thus using the answers retrieved from the SW as a way to improve standard search tasks.

Acknowledgements. This research was partially funded by the SmartProducts project (EC-231204).

References

[1] Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., Hellman, S. (2009) DBpedia. A Crystallization Point for the Web of Data. Journal of Web Semantics: Science, Services and Agents on the World Wide Web, 7(3), p.154-156.

[2] Bizer, C., Heath, T., Berners-Lee, T. (2009) Linked Data – The Story So Far. Journal on Semantic Web and Information Systems, 5(3), p.1-22.

[3] Kaufmann, E., Bernstein, A. (2007) How Useful Are Natural Language Interfaces to the Semantic Web for Casual End-Users?. In Proc. of the International Semantic Web Conference, p.281-294.

[4] Lopez, V., Nikolov, A., Fernandez, M., Sabou, M, Uren, V. and Motta, E. (2009) Merging and Ranking answers in the Semantic Web: The Wisdom of Crowds. In Proc. of the Asian Semantic Web Conference, p.135-152.

[5] Lopez, V., Nikolov, A., Sabou, M, Uren, V., Motta, E., and d'Aquin, M. (2010) Scaling up Question-Answering to Linked Data. In Proc. of the Int. Conference on Knowledge Engineering and Knowledge Management by the Masses, p.193-210.

[6] Wang, H., Tran, T., Haase, P., Penin, T, Liu, Q., Fu, L., Yu, Y. (2008) SearchWebDB: Searching the Billion Triples!. Billion Triple Challenge at the International Semantic Web Conference.

[7] Gueret, C., Groth, P., and Schlobach. S. (2009) eRDF: Live Discovery for the Web of Data. Billion Triple Challenge at International Semantic Web Conference.

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

[9] 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. Conference on Intelligent User Interfaces, p.180-189.

[10] Tablan, V, Damljanovic, D., and Bontcheva, K. (2008) A Natural Language Query Interface to Structured Information. In Proc. of the European Semantic Web Conference, p.361-375.

[11] Wang, C, Xiong, M., Zhou, Q., Yu, Y. (2007) PANTO: A portable Natural Language Interface to Ontologies. In Proc. of the European Semantic Web Conference, p.473-487.

[12] Hallett, C., Scott, D. and Power, R. (2007) Composing Questions through Conceptual Authoring. Computational Linguistics 33 (1), p.105-133.

[13] Polleres, A., Hogan, A., Harth, A., Decker, S. (2010) Can we ever catch up with the Web? The Journal of Semantic Web - Interoperability, Usability, Applicability,1(1-2), p.45-52.

[14] Fernandez, M., Lopez, V., Motta, E., Sabou, M., Uren, V., Vallet, D., Castells, P. (2008) Semantic Search meets the Web. In the Int. Conference on Semantic Computing, p.253-260.

[15] Gracia, J., Lopez, V., d'Aquin, M., Sabou, M., Motta, E., Mena, E. (2007) Solving Semantic Ambiguity to Improve Semantic Web based Ontology Matching. In the Ontology Matching Workshop at the International Semantic Web Conference, p.1-12.

[16] Lopez, V., Motta, E., Uren, V. and Pasin, M. (2007) AquaLog: An ontology-driven Question Answering System for Semantic intranets, Journal of Web Semantics, 5(2), p.72-105.

[17] Lopez, V., Sabou, M. and Motta, E. (2006) PowerMap: Mapping the Semantic Web on the Fly. In Proc. of International Semantic Web Conference, p.414-427.

[18] Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V. (2002) 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.

[19] Lopez, V., Sabou, M., Uren, V., Motta, E. (2009) Cross-Ontology Question Answering on the Semantic Web –an initial evaluation. In Proc. of the Knowledge Capture Conference.

[20] Uren, V., Sabou, M., Motta, E., Fernandez, M., Lopez, V., Lei, Y. (2011) Reflections on five years of evaluating semantic search systems. International Journal of Metadata, Semantics and Ontologies, 5(2), p.87-98.

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

[22] Meij, E., Bron, M., Hollink, L., Huurnink, B., Maarten de Rijke (2009) Learning Semantic Query Suggestions. In Proc. of the International Semantic Web Conference.

[23] Mollá, D, Vicedo, J. L. (2007) Question Answering in Restricted Domains: An Overview. Computational Linguistics, 33 (1), pp. 41-61.

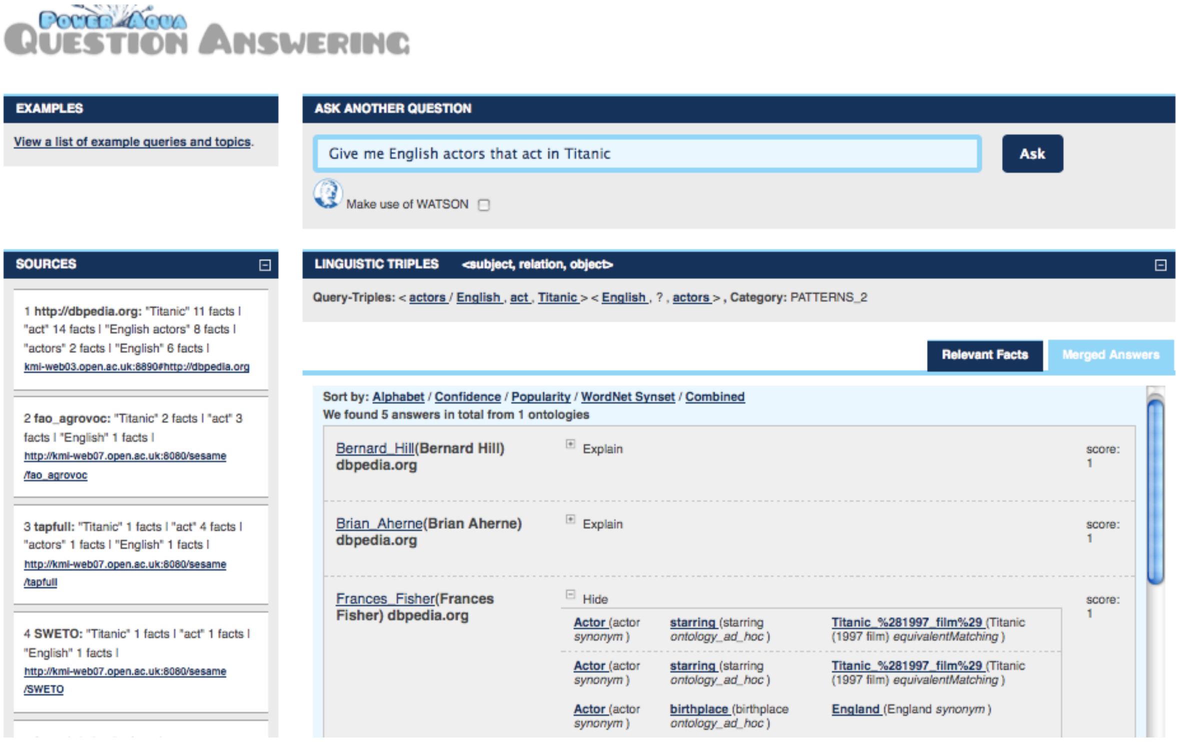
[24] Damljanovic, D., Agatonovic, M., Cunningham, H. (2010) Natural Language interface to ontologies: Combining syntactic analysis and ontology-based lookup through the user interaction. In Proc. of the European Semantic Web Conference, p.106-120.

[25] D'Aquin M., Lopez V., Motta, E.(2008) FABilT – Finding Answers in a Billion Triples. Billion triple challenge at the International Semantic Web Conference.

[26] Wrigley, S. N., Elbedweihy, K., Reinhard, D., Bernstein, A., and Ciravegna, F. (2010) Evaluating semantic search tools using the SEALS Platform. In Proc. of the Workshop on Evaluation of Semantic Technologies at International Semantic Web Conference.

[27] Resnik P. (1995) Disambiguating noun grouping with respect to WordNet senses. In Proc. of the 3rd Workshop on very Large Corpora. MIT, p.54-68.

[28] Brooke, J. (1996) SUS: a quick and dirty usability scale. In P. W. Jordan, B. Thomas, B. A.Weerdmeester, and I. L. McClelland, editors, Usability Evaluation in Industry, p. 189-194.



EXAMPLES

View a list of example queries and topics.

ASK ANOTHER QUESTION

Give me English actors that act in Titanic

Ask

Make use of WATSON □

SOURCES

1 **http://dbpedia.org:** "Titanic" 11 facts | "act" 14 facts | "English actors" 8 facts | "actors" 2 facts | "English" 6 facts |
kmi-web03.open.ac.uk:8890#http://dbpedia.org

2 **fao_agrovoc:** "Titanic" 2 facts | "act" 3 facts | "English" 1 facts |
http://kmi-web07.open.ac.uk:8080/sesame/fao_agrovoc

3 **tapfull:** "Titanic" 1 facts | "act" 4 facts | "actors" 1 facts | "English" 1 facts |
http://kmi-web07.open.ac.uk:8080/sesame/tapfull

4 **SWETO:** "Titanic" 1 facts | "act" 1 facts | "English" 1 facts |
http://kmi-web07.open.ac.uk:8080/sesame/SWETO

LINGUISTIC TRIPLES <subject, relation, object>

Query-Triples: < actors / English , act , Titanic > < English , ? , actors > , Category: PATTERNS_2

Relevant Facts | Merged Answers

Sort by: Alphabet / Confidence / Popularity / WordNet Synset / Combined
We found 5 answers in total from 1 ontologies

Bernard_Hill(Bernard Hill) dbpedia.org ⊞ Explain score: 1

Brian_Aherne(Brian Aherne) dbpedia.org ⊞ Explain score: 1

Frances_Fisher(Frances Fisher) dbpedia.org ⊟ Hide score: 1

Actor (actor *synonym*)	starring (starring *ontology_ad_hoc*)	Titanic_%281997_film%29 (Titanic (1997 film) *equivalentMatching*)
Actor (actor *synonym*)	starring (starring *ontology_ad_hoc*)	Titanic_%281997_film%29 (Titanic (1997 film) *equivalentMatching*)
Actor (actor *synonym*)	birthplace (birthplace *ontology_ad_hoc*)	England (England *synonym*)

Figure 1. Screenshot of PowerAqua for the query "Give me English actors that act in Titanic": in the top right part of the figure, under the title "ask another question" the user introduces her NL query in the text box. To support users in their initial interaction with the system, PowerAqua provides in the top left part of the interface a set of NL query examples. Once the user formulates a query, the system retrieves on the left hand side the list of semantic resources that are relevant to the user's query. On the right hand side of the interface, PowerAqua displays the set of linguistic triples in which the query have been translated and the final ranked list of answers obtained for the query, PowerAqua interface contains mechanisms to allow the user to see the answers before and after the merging (in the figure, Relevant Facts / Merged Answers), as well as to sort the responses according to the different ranking criteria, (in the figure Alphabet/ Confidence/ Popularity, etc.). Moreover, every item on the Onto-Triple and answers are links to ontology entries, giving the user the possibility to navigate through the ontological information.

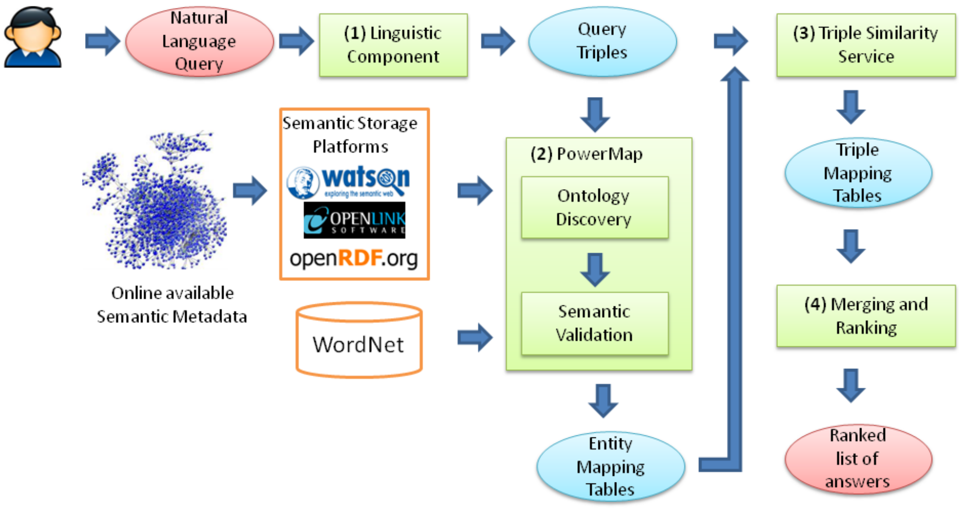


Figure 2. PowerAqua architecture and components

Table 1. Different performance times before and after adding DBpedia and filtering heuristics, and after the Virtuoso integration

NL Query:	**Before DBpedia**	**After DBpedia. (sesame)**	**After DBpedia (virtuoso)**
How many languages are used in Islamic countries?	34.5	95.2	30
Which Russian rivers end in the Black Sea	27.3	41.3	13
Who lives in the white house	13.7	17.9	36
Give me airports in Canada	14.22	23	16
List me Asian countries	15.3	67.4	25
Give me the main companies in India	43.9	17.4	17
Give me movies starring Jennifer Aniston	4.5	10.7	5
Which animals are reptiles?	7.1	42.8	16
Which islands belong to Spain	104	206	33
Find all the lakes in California	12.9	13.6	16
Tell me actors starring in films directed by Francis Ford Coppola	120	173	22
Find me university cities in Japan	-	68	35
Show me Spanish films with Carmen Maura	-	30.5	10
Give me English actors that act in Titanic	-	144	26
Give me tennis players in France	-	14.7	11
Television shows created by Walt Disney	-	9.4	10
Average response time	**32**	**54.3**	**20.06**