# Is Question Answering fit for the Semantic Web?: a Survey.

Vanessa Lopez[a,*], Victoria Uren[b], Marta Sabou[c] and Enrico Motta[b]
[a]*Knowledge Media Institute. The Open University. Walton Hall, Milton Keynes, MK17 6AA, United Kingdom.*
[b]*The University of Sheffield, S14DP, United Kingdom.*
[c]*MODUL University of Vienna, Austria.*

**Abstract.** With the recent rapid growth of the Semantic Web (SW), the processes of searching and querying content that is both massive in scale and heterogeneous have become increasingly challenging. User-friendly interfaces, which can support end users in querying and exploring this novel and diverse, structured information space, are needed to make the vision of the SW a reality. We present a survey on ontology-based Question Answering (QA), which has emerged in recent years to exploit the opportunities offered by structured semantic information on the Web. First, we provide a comprehensive perspective by analyzing the general background and history of the QA research field, from influential works from the artificial intelligence and database communities developed in the 70s and later decades, through open domain QA stimulated by the QA track in TREC since 1999, to the latest commercial semantic QA solutions, before tacking the current state of the art in open user-friendly interfaces for the SW. Second, we examine the potential of this technology to go beyond the current state of the art to support end-users in reusing and querying the SW content. We conclude our review with an outlook for this novel research area, focusing in particular on the R&D directions that need to be pursued to realize the goal of efficient and competent retrieval and integration of answers from large scale, heterogeneous, and continuously evolving semantic sources.

Keywords: Question Answering survey, Natural Language, Semantic Web, ontology.

## 1. Introduction

The emerging Semantic Web (SW) (Berners-Lee et al., 2001) offers a wealth of semantic data about a wide range of topics, representing real community agreement. We are quickly reaching the critical mass required to enable a true vision of large scale, distributed SW with real-world datasets, beyond prototypes and proof of concepts, leading to new research possibilities that can benefit from exploiting and reusing these freely-available data, unprecedented in the history of computer science. As such, there is now a renewed interest in the search engine market towards the introduction of semantics in order to enhance keyword search technologies (Fazzinga et al., 2010) (Hendler, 2010) (Baeza et al., 2010). The notion of introducing semantics to search on the Web is not understood in a unique way, according to (Fazzinga et al., 2010) the two most common uses of SW technology are: (1) to interpret Web queries and resources annotated with respect to some background knowledge described by underlying ontologies, and (2) since the extraction of semantic knowledge from Web sources is encoded in a Knowledge Base (KB), a closely related use is the search in the structured large datasets of the SW as a future alternative of the current web.

Apart from the benefits that can be obtained as more semantic data is published on the Web, the emergence and continued growth of a large scale SW poses some challenges and drawbacks:

- There is a gap between users and the SW: it is difficult for end-users to understand the complexity of the logic-based SW. Paradigms that allow the typical Web user to profit from the expressive power of SW data-models, while hiding the complexity behind, are of crucial importance.

[*] Corresponding author. E-mail: v.lopez@open.ac.uk

- The processes of searching and querying content that is massive in scale and highly heterogeneous have become increasingly challenging: current ontology-based approaches have difficulties to scale their models successfully and exploit the increasing amount of online available, distributed semantic metadata.

Hence, user-friendly interfaces that can handle the real-world Web of data, to support end users in querying and exploring this novel and diverse, structured information space, are an important contribution to make the vision of the SW a reality.

Consistent with the role played by ontologies in structuring semantic information on the Web, recent years have witnessed the rise of ontology-based Question Answering (QA) as a new paradigm of research, to exploit the expressive power of ontologies, beyond the highly ambiguous and impoverished representation of user information needs in keyword-based queries. QA systems have been investigated by several communities (Hirschman et al., 2001). Traditionally, QA approaches have largely been focused on retrieving answers from raw text, most emphasis being made on using ontologies to mark-up and make the retrieval smarter by using query expansion (McGuinness, 2004). The novelty of this new trend of ontology-based QA is to exploit the SW information for making sense of, and answering, user queries, leading to a new twist on the old issues associated with Natural Language Interfaces to Databases (NLIDB), which has long been an area of research in the artificial intelligence and database communities (Androutsopoulos et al., 1995).

In this paper, we present a survey of ontology-based QA systems and relevant related work. We look at the promises of this novel research area from two perspectives. First, its contributions to the context of QA systems in general; and second, its potential to go beyond the current state of the art in SW interfaces that support end-users to reuse and querying the massive and heterogeneous SW content, and thus, despite its open issues, to go one step forward in bridging the gap between the user and the SW.

We seek a comprehensive perspective on this novel area by analyzing the key dimensions in the formulations of the QA problem in Section 2. We classify a QA system, or any approach to query the SW content, according to four dimensions based on the type of questions (input), the sources (unstructured data such as documents, or structured data in a semantic or non-semantic space), the scope (domain-specific, open-domain), and the traditional intrinsic problems derived from the search environment and scope of the system. To start with, we introduce in Section 3 the general background and history of the QA research field, from the influential works in the early days of research on architectures for NLIDB in the 70s (Section 3.1), through the strong and well-founded current approaches for open domain QA over text stimulated by the QA track in TREC since 1999 (Section 3.2), to the latest proprietary (commercial) semantic QA systems, based on data that is by and large manually coded and homogeneous (Section 3.3). Then, in Section 4 we discuss the achievements of state of the art ontology-based QA systems (Section 4.1) and their drawbacks (restricted domain) when considering the SW in the large (Section 4.2). We then review the latest trends in open domain QA interfaces for the SW (Section 4.3) and look at the performance of the reviewed ontology-based QA systems based on the evaluations that have been conducted to test them (Section 4.4). We finish this Section with a discussion on the competences and achievement of these systems in the QA scenario (Section 4.5), highlighting the open issues on QA for the SW (Section 4.6). In Section 5, we give some basic notions, lessons and open research issues of various relevant approaches, developed in the last decade, that have attempted to support end users in querying and exploring the massive publicly available heterogeneous SW information, from early global-view information systems (Section 5.1) and restricted domain Semantic Search (Section 5.2), through the latest works on open domain large scale Semantic Search and Linked Data (Bizer, Heath, et al., 2009) interfaces (Section 5.3). In Section 6, we argue how this new ontology-based search paradigm based on natural language QA, which combines ideas from traditional QA, is a promising direction towards the realization of user-friendly interfaces for the SW and may one day lead to overcoming the research gaps identified for each of the analyzed dimensions, as they allow users to express arbitrarily complex information needs in an intuitive fashion. We conclude in Section 7 with an outlook for this research area, in particular, our view on the potential directions ahead to realize its ultimate goal: to retrieve and combine answers from multiple, heterogeneous and automatically discovered semantic sources.

## 2. Goals and dimensions of Question Answering

The goal of QA systems, as defined by (Hirschman et al., 2001), is to allow users to ask questions in

Natural Language (NL), using their own terminology, and receive a concise answer. In this Section, we give an overview of the multiple dimensions in the QA process. These dimensions can be extended beyond NL QA systems to any approach to help the casual users locating and querying the massive and often heterogeneous structured data on the Web.

We can classify a QA system, and any semantic approach for searching and querying SW content, according to four interlinked dimensions (see Figure 2.1): (1) the input or type of questions it is able to accept (facts, dialogs, etc); (2) the sources it is based on for deriving the answers (structured vs. unstructured data); (3) the scope (domain specific vs. domain independent), and (4) how it copes with the traditional intrinsic problems that the search environment imposes in any non-trivial search system (e.g.: adaptability and ambiguity)

At the **input** level, the issue is balancing usability and higher expressivity at the level of the query, hiding the complexity of ontology-based SQL-like languages to query the semantic data (the user doesn't need to learn complex query languages), while involving a representation of the information need that grasps and exploits the conceptualizations and content meanings involved in user needs. Different kinds of search inputs provide complementary affordances to support the ordinary user to query the semantic data. The best feature of keyword-based searches is its simplicity, nevertheless, in this simplicity lies its main limitation: the lack of expressivity, e.g., to express relationships between words and the lack of context to disambiguate between different interpretations of the words. In (Moldovan et al., 2003), QA systems are classified according to five increasingly sophisticated types of questions they can accept as input: systems based on factoids, systems with reasoning mechanisms, systems that deal with temporal reasoning, systems that fuse answers from different sources, interactive (dialog) systems and systems capable of analogical reasoning. Most research in QA focuses on factual QA, where we can distinguish between *Wh-queries* (who, what, how many, etc.), or commands (name all, give me, etc.) requiring an element or list of elements as an answer, or an affirmation / negation type of questions. As pointed out in (Hunter, 2000) more difficult kinds of factual questions include those which ask for opinion, like *Why* or *How* questions, which require understanding of causality or instrumental relations, and *What* questions which provide little constraint in the answer type, or definition questions. In this survey we focus on factual QA, including open-domain definition questions, i.e., *What*-queries about arbitrary concepts, as the foundations and "warm up" for future research on more ambitious forms of QA on the SW. In the SW context factual QA means that answers are ground facts as typically found in KBs.

QA systems can also be classified according to the different **sources** used to generate an answer as follows:

- Natural Language interfaces to structured data on databases (NLIDB traced back to the late sixties (Androutsopoulos et al., 1995)).
- QA over semi-structured data (e.g.: health records, yellow pages, wikipedia infoboxes).
- Open QA over unstructured documents or free text, fostered by the open-domain QA track introduced by TREC in 1999 (TREC-8).
- QA over structured semantic data, where the semantics contained in ontologies provide the context needed to solve ambiguities, interpret and answer the user query.

Another distinction between QA systems is whether they are domain-specific (closed domain) or domain-independent (open domain). The **scope** of a QA system, up to which level (partial or total) they are able to exploit the publicly available SW heterogeneous content (obtained from sources that are autonomously created and maintained), is closely related to the type of sources used to generate an answer. Traditionally, NLIDBs are closed or domain-specific while open QA systems are domain-independent and open. Ontology-based QA emerged as a combination of ideas of two different research areas - it enhances the scope of NLIDB over structured data, by being agnostic to the domain of the ontology that it exploits; and presents complementary affordances to open QA over free text (TREC), the advantage being that it can help with answering questions requiring situation-specific answers, where multiple pieces of information (from one or several sources) need to be assembled to infer the answers at run time. Nonetheless, most ontology-based QA systems are akin to NLIDB in the sense that they are able to extract precise answers from structured data in a specific domain scenario at a time, instead of retrieving relevant paragraphs of text in an open scenario. Latest proprietary QA systems over structured data, such as True-Knowledge, Powerset (detailed in Section 3.3), are open domain but restricted to their own proprietary sources.

Domain-independent systems are challenged by a **search environment** that can be characterized by its large scale, heterogeneity, openness and multilinguality. The search environment influences to what level semantic systems perform a deep exploitation of the semantic data. In order to take full advantage of the inherent characteristics of the semantic information space to extract the most accurate answers for the users, QA systems need to tackle various **traditional intrinsic** problems derived from the search environment, such as:

- Mapping the terminology and information needs of the user into the terminology used by the sources, in such a form that: (1) it can be evaluated using standard query processing and inferencing techniques, (2) it does not affect portability or adaptability of the systems to new domains, and (3) it leads to the correct answer.
- Disambiguation between all possible interpretations of a user query: independently of the type of query, any non-trivial NL QA system has to deal with ambiguity. Furthermore, in an open scenario ambiguity cannot be solved by means of an internal unambiguous knowledge representation as in domain-restricted scenarios. In open-domain scenarios, systems face the problem of polysemous words, with different meanings according to different domains.
- Because answers may come from different sources, and different sources have varying levels of quality and trust, knowledge fusion and ranking measures should be applied to select the better sources, fuse similar answers together and rank the answers across sources.
- With regards to scalability, in general terms, there is a trade-off between the complexity of the querying process and the amount of data systems can use in response to a user demand in a reasonable time.

Multilinguality issues, the ability to answer a question posed in one language using an answer corpus in another language, fostered by the Multilingual Question Answering Track at the cross language evaluation forum (CLEF)[1] since 2002 (Forner et al., 2010), are not reviewed in this survey. This is because in the context of QA in the open SW, challenges such as scalability and heterogeneity need to be tackled first to obtain answer across sources.

NL interfaces are an often-proposed solution in the literature for the casual users (Kauffman and Bernstein, 2007), being particularly appropriate in domains for which there are authoritative and comprehensive databases or resources, and are complex enough to warrant the need of a QA system (Mollá and Vicedo, 2007). However, their success has been typically overshadowed by both the brittleness and habitability problems (Thompson et al., 2005), defined as the mismatch between the user expectations and the capabilities of the system with respect to its NL understanding and what it knows about (users do not know what it is possible to ask). As stated in (Uren et al., 2007) iterative and exploratory search modes are important to the **usability** of all search systems, to support the user to know what is the knowledge of the system or what subset of NL is possible to ask about, as well as the ability to present and provide justifications for an answer in an intuitive way (NL generation), suggest the presence of unrequested but related information, actively help the user to refine or recommend searches or propose alternate paths of exploration. For example, view based search and forms can help the user explore the search space better than keyword-based or NL querying systems, but they become tedious to use in large spaces and impossible in heterogeneous ones.

Usability of NL interfaces is not covered in this review so for extended information see (Uren et al., 2007) and (Kauffman and Bernstein, 2007).

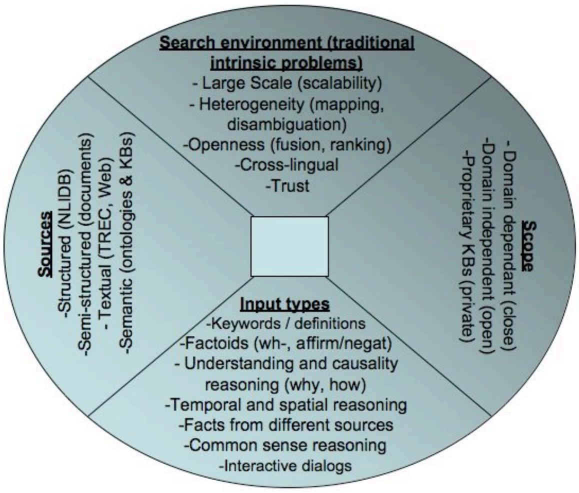
Search environment (traditional intrinsic problems)
- Large Scale (scalability)
- Heterogeneity (mapping, disambiguation)
-Openness (fusion, ranking)
-Cross-lingual
-Trust
Sources
-Structured (NLIDB)
-Semi-structured (documents)
- Textual (TREC, Web)
-Semantic (ontologies & KBs)
Scope
- Domain dependant (close)
-Domain independent (open)
-Proprietary KBs (private)
Input types
-Keywords / definitions
-Factoids (wh-, affirm/negat)
- Understanding and causality reasoning (why, how)
-Temporal and spatial reasoning
-Facts from different sources
-Common sense reasoning
-Interactive dialogs

Figure 2.1. The dimensions of Question Answering and query and search interfaces in general.

## 3. Related work on Question Answering

A short survey of related work on QA targeted to different types of sources: structured databases, unstructured free text and precompiled fact-based KBs, is presented here.

[1] http://clef.isti.cnr.it

### *3.1. NLIDB: Natural Language Interfaces to Databases*

NLIDB has long been an area of research in the artificial intelligence and database communities, even if in the past decade it has somewhat gone out of fashion (Androutsopoulos et al., 1995). However, the SW provides a new and potentially important context in which the results from this area can be applied.

The use of NL to access relational databases can be traced back to the late sixties and early seventies (Androutsopoulos et al., 1995). The first QA systems were developed in 1960s and they were basically NL interfaces to expert systems, tailored to specific domains, the most famous ones being **BASEBALL** and **LUNAR**. Both systems were domain specific, the former answered questions about the US baseball league over the period of one year, the later answered questions about the geological analysis of rocks returned by the Apollo missions. LUNAR was able to answer 90% of the questions in its domain when posed by untrained geologists. In (Androutsopoulos et al., 1995) a detailed overview of the state of the art for these early systems can be found.

Some of the early NLIDBs approaches relied on pattern-matching techniques. In the example described by (Androutsopoulos et al., 1995), a rule says that if a user's request contains the word "capital" followed by a country name, the system should print the capital which corresponds to the country name, so the same rule will handle "what is the capital of Italy?", "print the capital of Italy", "could you please tell me the capital of Italy". This shallowness of the pattern-matching would often lead to failures but it has also been an unexpectedly effective technique for exploiting domain-specific data sources.

The main drawback of these early NLIDBs systems is that they were built having a particular database in mind, thus they could not be easily modified to be used with different databases and were difficult to port to different application domains. Configuration phases were tedious and required a long time, because of different grammars, hard-wired knowledge or hand-written mapping rules that had to be developed by domain experts.

The next generation of NLIDBs used an intermediate representation language, which expressed the meaning of the user's question in terms of high-level concepts, independently of the database's structure (Androutsopoulos et al., 1995), making a separation of the (domain-independent) linguistic process and the (domain-dependent) mapping process into the database to improve the front end portability (Martin et al., 1985).

The **formal semantics approach** presented in (De Roeck et al., 1991) made a clear separation between the NL front ends, which have a very high degree of portability, and the back end. The front end provides a mapping between sentences of English and expressions of a formal semantic theory, and the back end maps these into expressions, which are meaningful with respect to the domain in question. Adapting a developed system to a new application involves altering the domain specific back end alone.

**MASQUE/SQL** (Androutsopoulos et al., 1993) is a portable NL front end to SQL databases. It first translates the NL query into an intermediate logic representation, and then translates the logic query into SQL. The semi-automatic configuration procedure uses a built-in domain editor, which helps the user to describe the entity types to which the database refers, using an is-a hierarchy, and then to declare the words expected to appear in the NL questions and to define their meaning in terms of a logic predicate that is linked to a database table/view.

More recent work in the area (2003) can be found in **PRECISE** (Popescu, et al, 2003). PRECISE maps questions to the corresponding SQL query by identifying classes of questions that are understood in a well defined sense: the paper defines a formal notion of *semantically tractable* questions. Questions are sets of attribute/value pairs and a relation token corresponds to either an attribute token or a value token. Each attribute in the database is associated with a wh-value (what, where, etc.). Also, a lexicon is used to find synonyms. The database elements selected by the matcher are assembled into a SQL query, if more than one possible query is found, the user is asked to choose between the possible interpretations. However, in PRECISE the problem of finding a mapping from the tokenization to the database requires all tokens to be must distinct; questions with unknown words are not semantically tractable and cannot be handled. As a consequence, PRECISE will not answer a question that contains words absent from its lexicon. Using the example suggested in (Popescu, et al, 2003), the question "what are some of the neighbourhoods of Chicago?" cannot be handled by PRECISE because the word "neighbourhood" is unknown. When tested on several hundred questions, 80% of them were semantically tractable questions, which PRECISE answered correctly, and the other 20% were not handled.

NLI have attracted considerable interest in the Health Care area. In the approach presented in (Hallet et al., 2007) users can pose complex NL queries to a large medical repository, question formulation is facilitated by means of ***Conceptual Authoring***. A logical representation is constructed using a query editing NL interface, where, instead of typing in text, all editing operations are defined directly on an underlying logical representation governed by a predefined ontology ensuring that no problem of interpretation arises.

However, all these approaches still need an intensive configuration procedure. To reduce the formal complexity of creating underlying grammars for different domains, (Minock et al., 2008), and most recently **C-PHRASE** (Minock et al., 2010) present a state-of-the-art authoring system for NLIDB. The author builds the semantic grammar through a series of naming, tailoring and defining operations within a web-based GUI, as such the NLI can be configured by non-specialized, web based technical teams. In that system queries are represented as expressions in an extended version of Codd's Tuple Calculus and context-free synchronous grammars extended with lambda functions to represent semantic grammars, which may be directly mapped to SQL queries or first-order logic expressions. High-order predicates are also used to support ranking and superlatives.

## *3.2. Open Domain Question Answering over text*

### *3.2.1. Document-based Question Answering*

We have already pointed out that research in NLIDB is currently a bit 'dormant'; therefore it is not surprising that most current work on QA, which has been rekindled largely by the Text Retrieval Conference (sponsored by the American National Institute, NIST, and the Defense Advanced Research Projects Agency, DARPA) and the monolingual and cross-lingual QA evaluations at CLEF, is somewhat different in nature from querying structured data. These campaigns enable research in QA from the IR perspective, where the task consists in finding the text that contains the answer to the question and extracting the answer. The ARDA's Advanced Question and Answering for Intelligence funded the AQUAINT program, a multi-project effort to improve the performance of QA systems over free large heterogeneous collections of structured and unstructured text or media. Given the large, uncontrolled text files and the very weak world knowledge available from WordNet and gazetteers, these systems have performed surprisingly well. For example, the **LCC system** (Moldovan et al., 2002) that uses a deeper linguistic analysis and iterative strategy obtained a score of 0.856 by answering correctly 415 questions of 500 in TREC-11 (2002).

There are linguistic problems common in most kinds of NL understanding systems (e.g., all question understanding and processing systems are required to recognize equivalent questions, regarding idiomatic forms). As a high-level overview on the state of the art techniques for open QA (Pasca, 2003), some of the methods use shallow keyword-based expansion and contraction techniques to locate interesting sentences from the retrieved documents, based on the presence of strings of the same type of the expected answer type. Ranking is based on syntactic features such as word order or similarity to the query. Templates can be used to find answers that are just reformulations of the question. Most of the systems classify the query based on the type of the answer expected: a name (i.e. person, organization), a quantity (monetary value, distance, length, size) or a date. Question's classes are arranged hierarchically in taxonomies and different types of questions require different strategies. These systems often utilize world knowledge that can be found in high-level ontologies such as WordNet, or the Suggested Upper Merged Ontology (SUMO) to pinpoint question types and match entities to the expected answer type. More sophisticated syntactic, semantic and contextual processing to construct an answer might include: named-entity (NE) recognition, relation extraction, co-reference resolution, syntactic alternations, word sense disambiguation (WSD), logical inferences and temporal-spational reasoning.

Going into more details, QA applications for text typically involve two steps, as cited by (Hirschman et al., 2001): (1) "identifying the semantic type of the entity sought by the question"; and (2) "determining additional constraints on the answer entity". Constraints can include, for example, keywords (that may be expanded using synonyms or morphological variants) to be used in the matching of candidate answers, and syntactic or semantic relations between a candidate answer entity and other entities in the question. Various systems have, therefore built hierarchies of question types based on the types of answers sought (Moldovan et al., 1999) (Hovy et al., 2000) (Wu et al., 2003) (Srihari et al., 2004). An efficient system should group together equivalent question types independently of how the query is formulated.

For instance, in **LASSO** (Moldovan et al., 1999) a question type hierarchy was constructed from the analysis of the TREC-8 training data, a score of 55.5% for short answers and 64.5% for long answers was achieved. Given a question, it can find automatically (a) the type of question (what, why, who, how, where), (b) the type of answer (person, location, etc.), (c) the question focus, defined as the "main information required by the interrogation" (useful for "what" questions which say nothing about the information asked for by the question), (d) the relevant keywords from the question. Occasionally, some words of the question do not occur in the answer (for example, the focus "day of the week" it is very unlikely to appear in the answer). Therefore, it implements similar heuristics to the ones used by NE recognizer systems for locating the possible answers.

NE recognition and information extraction (IE) are powerful tools in free text QA. One study showed that over 80% of questions asked for a named entity as a response (Srihari et al., 2004). As stated in (Srihari et al., 2004) "high-level domain independent IE, able to extract multiple relationships between entities and event information (WHO did WHAT), is expected to bring a breakthrough in QA".

The best results of the TREC9 (De Boni, M., 2001) competition were obtained by the **FALCON** system described in (Harabagiu et al., 2000), with a score of 58% for short answers and 76% for long answers. In FALCON the answer semantic categories mapped into categories covered by a NE Recognizer. When the answer type is identified, it is mapped into an answer taxonomy, the top categories are connected to several word classes from WordNet. In an example presented in (Harabaigiu et al., 2000), FALCON identifies the expected answer type of the question "what do penguins eat?" as food because "it is the most widely used concept in the glosses of the subhierarchy of the noun synset {eating, feeding}". All nouns (and lexical alterations), immediately related to the concept that determines the answer type, are considered among the keywords. Also, FALCON gives a cached answer if the similar question has already been asked before; a similarity measure is calculated to see if the given question is a reformulation of a previous one.

The system described in Litkowski et al. (Litkowski, 2001), called **DIMAP**, extracts "semantic relation triples" after a document is parsed, converting a document intro triples. The DIMAP triples are stored in a database in order to be used to answer the question. The semantic relation triple described consists of a discourse entity (SUBJ, OBJ, TIME, NUM, ADJMOD), a semantic relation that characterizes the entity's role in the sentence and a governing word to which the entity stands in the semantic relation. The parsing process generated an average of 9.8 triples per sentence in a document. The same analysis was done for each question, generating on average 3.3 triples per sentence, with one triple for each question containing an unbound variable, corresponding to the type of question. The discourse entities are the driving force in DIMAP triples, key elements (key nouns, verbs, and any adjective or modifier noun) are determined for each question type (the system categorized questions in six types: time, location, who, what, size and number questions).

### *3.2.2. Question Answering On the Web*

QA systems over the Web have the same three main components than QA systems designed to extract answer to factual questions by consulting a repository of documents (TREC): (1) a query formulation mechanism that translates the NL queries into the required IR queries, (2) a search engine over the Web, instead of a IR engine on the top of the documents, that retrieves the multiple requests, and (3) the answer extraction module that extracts answers from documents. A technique commonly share in Web and TREC-systems, is to use WordNet or NE tagging to classify the type of the answer. For instance, **Mulder** (Kwok al., 2001) is a QA system for factual questions over the Web, which relies on multiple queries sent to the search engine Google. To form the right queries for the search engine, the query is classify using WordNet to determining the type of the object of the verb in the question (numerical, nominal, temporal), then a reformulation module converts a question into a set of keyword queries by using different strategies: varying the specificity of the query to the most important keywords, quoting partial sentences (detecting noun phrases), conjugating the verb, or performing query expansion with WordNet. In Mulder, an answer is extracted from the snippets or summaries returned by Google, which is less expensive than extracting answers directly from a web page. Then, to reduce the noise or incorrect information typically found on the Web and improve accuracy, Mulder cluster similar answers together and picks the best answer with a voting procedure. Mulder takes advantage of Google ranking algorithms base on PageRank and the proximity or frequency of the words, as well as from Google wider coverage: "with a large collection there is a higher probability of finding target sentences". An evaluation using the TREC-8 questions, based on the Web, instead of the TREC document collection, showed that Mulder recall is more than a factor of three higher than AskJeeves.

Search engines such as **AskJeeves**[2] provide NL question interfaces to the Web to retrieve documents, not answers. AskJeeves looks up the user's question in its database and returns a list of matching question that it knows how to answer, the user selects the most appropriate entry in the list, and he is taken to the web pages where the answer can be found. AskJeeves relies on human editors to match question templates with authoritative sites.

Other approaches are based on statistical or semantic similarities. For example, **FAQ Finder** (Burke et al., 1997) is a NL QA system that uses files of FAQs as its KB; it uses two metrics to match questions to answers: statistical similarity and semantic similarity. For shorter answers over limited structured data, NLP-based systems have generally been better than statistical based ones, which need a lot of domain specific training and long documents with large quantities of data that have enough words for statistical comparisons to be considered meaningful. Semantic similarity scores rely on finding connections through WordNet between the user's question and the answer. The main problem here is the inability to cope with words that are not explicitly found in the KB. **Gurevych's approach** (Gurevych et al., 2009) approach tries to identify semantically equivalent questions, which are paraphrases of user queries, already answered in social Q&A sites, such as Yahoo!Answers.

Finally, recently Google is also evolving to a NL search engine, providing precise answers to some specific factual queries, together with the Web pages from which the answer has been obtained, but it does not yet distinguish between queries such as "where Barack Obama was born" or "when Barack Obama was born" (as October 2010).

### *3.3. Latest developments on structured (proprietary) open Question Answering*

As we have seen in the previous subsections, large-scale, open-domain QA has been addressed and stimulated in the last decade (since 1999) by the TREC QA track evaluations. The current trend is to introduce semantics to search for Web pages based on the meaning of the words in the query, rather than just matching keywords and ranking pages by popularity (like Google or Yahoo). Within this third recent trend, notwithstanding the openness of the scenario, there is a new tendency to focus on directly obtaining structured answers to user queries from pre-compile facts, as some kind of semantic information used to understand and disambiguate they intended meaning of the words and how they are connected.

This class of systems includes START, which came online in 1993 as the first QA system available on the Web, and several industrial startups such as Powerset, Wolfram Alpha, True Knowledge[3], among others. These systems use a well-established approach, which consists of semi-automatically building their own homogeneous, comprehensive factual KBs about the world, similarly to OpenCyc and Freebase[4].

START (Katz et al., 2002) answers questions about geography and the MIT infolab, with a performance of 67% over 326 thousand queries, it uses highly edited knowledge-bases to retrieve tuples, in the subject-relation-object form, as pointed out by (Katz et al., 2002), although not all possible queries can be represented in the binary relational model, in practice these exceptions occur very infrequently[5]. START allows annotations in NL to describe the query and compares it against the annotations derived from the knowledge base. However, START suffers from the knowledge engineering bottleneck, as only trained individuals can add knowledge and expand the system's coverage and integrate Web sources.

With respect to commercial systems, PowerSet tries to match the meaning of a query with the meaning of a sentence in Wikipedia, Powerset not only works on the query side of the search (converting the NL queries into database understandable queries, and then highlighting, the relevant passage of the document) but it also reads every word of every (Wikipedia) page to extract the semantic meaning (compiling *factzs* - similar to triples, from pages across Wikipedia, together with the Wikipedia page locations and sentences that support each factz), using Freebase and its semantic resources to annotate it. The Wolfram Alpha knowledge inference engine builds a broad trusted KB about the world by ingesting massive amounts of information (approx. 10TBs, still a

[2] http://www.ask.co.uk

[3] http://www.powerset.com/,
http://www.wolframalpha.com/index.html, and
http://www.trueknowledge.com/

[4] www.opencyc.org, http://www.freebase.com

[5] Knowledge representation formalisms for the SW, such as RDF or OWL (Mc Guinness and Van Harmelen, 2004) also subscribe to this binary relational model and express statements as <subject, predicate, object>.

tiny fraction of the Web), while True Knowledge relies on users to add and curate information

## 4. Semantic ontology-based Question Answering

In this section we look at ontology-based semantic QA systems (also referred in this paper as semantic QA systems because they make use of semantic data sources), which take queries expressed in NL and a given ontology as input, and return answers drawn from one or more KBs that subscribe to the ontology. Therefore, they do not require the user to learn the vocabulary or structure of the ontology to be queried. We look at the performance of these systems and their limitations when considering the SW by large. Aiming to overcome those drawbacks, we look into the latest research in QA over the SW not limited by the single-ontology assumption. Finally, the potential advantages of QA over semantic sources, with respect to the QA systems over other sources (databases, text or precompiled fact-bases) analyzed in Section 3, and the open issues that need to be solved to reach its full potential are discussed.

### *4.1. Ontology-specific QA systems*

Since the steadily growth of the SW and the emergence of large-scale semantics the necessity of NLI to ontology-based repositories has become more acute, re-igniting interest in NL front ends. This have also been supported by usability studies (Kaufmann and Bernstein, 2007), which showed that casual users, typically overwhelmed by the formal logic of the SW, preferred using a NL interface to query an ontology, leading to a new trend on ontology based QA systems, where the power of ontologies as a model of knowledge is directly exploited for the query analysis and translation, providing a new twist on the old issues of NLIDB, and therefore focusing in portability and performance, replacing the costly domain specific NLP techniques with rather shallow ones. A wide range of off-the-shelf components, such as triple stores (e.g. sesame[6]) or text retrieval engines (e.g. Lucene[7]), and freely available domain-independent linguistic and lexical resources, dictionaries like WordNet or FrameNet[8] and NLP Parsers like Stanford Parser[9] (Klein and Manning, 2002), support the evolution of these new NLI.

The systems vary on two main aspects: (1) the degree of domain customization they require, which correlates with their retrieval performance, and (2) the subset of NL they are able to understand (full grammar-based NL, controlled or guided NL, pattern based), in order to reduce both complexity and the habitability problem, pointed out as the main issue that hampers the successful use of NLi (Kaufmann and Bernstein, 2007).

At one end of the spectrum, systems are tailored to a domain and most of the customization has to be performed or supervised by domain experts, for instance QACID is based on a collection of queries from a given domain that are analyzed and grouped into clusters, where each cluster, containing different expressions asking for the same information, is manually associated to SPARQL queries. In the middle of the spectrum, a system such as ORAKEL (Cimiano et al., 2007) requires a significant domain-specific lexicon customization process, for systems like the e-librarian (Linckels, 2005) performance is dependent on the manual creation of a domain dependent lexicon and dictionary. While at the other end, in systems like AquaLog (Lopez et al., 2007), the customization is done on the fly while the system is being used, by using interactivity to learn the jargon of the user over time, GINSENG (Bernstein and Kauffman et al., 2006) guides the user through menus to specify NL queries, while, systems such as PANTO (Wang, 2007), NLP-Reduce, Querix (Kaufmann et al., 2006), QuestIO (Tablan et al., 2008), generate lexicons, or ontology annotations (FREya by Damljanovic et al.), on demand when a KB is loaded. In what follows, we look into these systems in detail. A comparison of these approaches is presented in **Table 4.1**.

**AquaLog** (Lopez, et al, 2004), (Lopez et al., 2007) allows the user to choose an ontology and then ask NL queries with respect to the universe of discourse covered by the ontology. AquaLog is ontology independent because the configuration time required to customize the system 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 the semantics of the ontology, and the use of generic lexical resources, such as WordNet. In a first step, the Linguistic Component uses the GATE infrastructure and resources (Cunningham et al., 2002) to obtain a set of syntactic annotations associated with the input query. The set

[6] http://www.openrdf.org/

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

[8] http://wordnet.princeton.edu
http://framenet.icsi.berkeley.edu

[9] http://nlp.stanford.edu/downloads/lex-parser.shtml

of annotations is extended by the use of JAPE grammars[10] to identify terms, relations, question indicators (who, what, etc.), features (voice and tense) and to classify the query into a category. Knowing the category and GATE annotations for the query, the Linguistic Component creates the linguistic triples or Query-Triples. Then, these Query-Triples are further processed and interpreted by the Relation Similarity Service Component, which uses the available lexical resources, the structure and vocabulary of the ontology to map them to ontology-compliant Onto-Triples, from where an answer is derived. At the syntactic level AquaLog identifies ontology mappings for all the terms and relations in the Query-Triples by considering the entity labels. It relies on string based comparison methods and WordNet. At the semantic level, AquaLog's interactive relation similarity service uses the ontology taxonomy and relationships to disambiguate between the possible terms or relations and to link the ontology triples, in order to obtain the equivalent representation of the user query. When the ambiguity cannot be resolved by domain knowledge the user is asked to choose between the alternative readings. AquaLog includes a learning component to automatically obtain domain-dependent knowledge by creating a lexicon, which ensures that the performance of the system improves over the time, in response to the particular community jargon (vocabulary) used by end users. AquaLog uses generalization rules to learn novel associations between the NL relations used by the users and the ontology structure. Once the question is entirely mapped to the underlying ontological structure the corresponding instances are obtained as an answer.

**QACID** (Fernandez, Izquierdo et al., 2009) relies on the ontology, a collection of user queries, and an entailment engine that associated new queries to a cluster of queries. Each query is considered as a bag of words, and the mapping is done through string distance metrics (Cohen et al., 2003) and an ontological lexicon to map words in NL queries to instances. Prior to launching the corresponding SPARQL query for the cluster, the SPARQL generator replaces the ontology concepts with the data instances appearing in the original query. This system is at the end of the spectrum because the performance depends on the variety of questions collected in the domain, the process is domain-dependent, costly and can only be applied to domains with limited coverage.

**ORAKEL** (Cimiano et al., 2007) is a NL interface that translates factual *wh-queries* into F-logic or SPARQL and evaluates them with respect to a given KB. The main feature is that it makes use of a compositional semantic construction approach thus being able to handle questions involving quantification, conjunction and negation in a classical way. In order to translate factual wh-queries it uses an underlying syntactic theory built on a variant of *Lexicalized Tree Adjoining Grammar* (LTAG), extended to include ontological information. The parser makes use of two different lexicons: the general lexicon and the domain lexicon. The general or domain independent lexicon includes closed-class words such as determiners, i.e.: a, the, every, etc., as well as question pronouns, i.e.: who, which, etc. The domain lexicon, in which natural expressions, verbs, adjectives and relational nouns, are mapped to corresponding relations specified in the domain ontology, varies from application to application and, for each application, this lexicon has to be partially generated by a domain expert. The semantic representation of the words in the domain independent lexicon makes reference to domain independent categories, as given for example by a foundational ontology such as DOLCE. This assumes that the domain ontology is somehow aligned to the foundational categories provided by the foundational ontology. Therefore, the domain expert is only involved in the creation of the domain specific lexicon, which is actually the most important lexicon as it is the one containing the mapping of linguistic expressions to domain-specific predicates. The domain expert has to instantiate subcategorization frames and maps these to domain-specific relations in the ontology. In regards to semantics, it uses a subcategorization information automatically acquired from a big corpus, a statistical parser, and the WordNet synsets (in the most frequent sense) that best generalize the selectional preferences at each argument in the subcategorization frame. The approach is independent of the target language, which only requires a declarative description in Prolog of the transformation from the logical form to the target language.

The "**e-Librarian**" (Linckels, 2005) understands the sense of the user query to retrieve multimedia resources from a KB. First, the NL query is preprocessed into its linguistic classes, in the form of triples, and translated into an unambiguous logical form, by mapping the query to an ontology to solve ambiguities. If a query is composed of several linguistic clauses, each one is translated separately and the logical concatenation depends on the conjunction

[10] JAPE is a language for creating regular expressions applied to linguistic annotations in a text corpus

words used in the question. The system relies on simple, string-based comparison methods (e.g., *edit distance metrics*) and a domain dictionary to look-up lexically related words (synonyms) because general-purpose dictionaries like WordNet are often not appropriate for specific domains. Regarding portability, the creation of this dictionary is costly, as it has to be created for each domain, but the strong advantage of this is that it provides very high performance difficult to obtain with general-purpose dictionaries (from 229 user queries 97% were correctly answered in the evaluation). The e-librarian does not return the answer to the user's question, but it retrieves the most pertinent document(s) in which the user finds the answer to her question.

Moving into the systems that do not necessitate any customization effort or previous pre-processing, (Kaufmann and Bernstein, 2007) presented four different ontology-independent query interfaces with the purpose of studying the usability of NLI for casual end-users. These four systems lie at different positions of what they call the *Formality Continuum*, where the freedom of a full NL and the structuredness of a formal query language are at opposite ends of the continuum. The first two interfaces, NLP-Reduce and Querix allow users to pose questions in full or slightly controlled English. The third interface Ginseng / GINO offers query formulation in a controlled language akin to English. Therefore, the first three interfaces lie on the NL end of the formality Continuum towards its middle, as such they analyze a user query, match it to the content of a KB, and translates these matches into statements of a formal query language (i.e, SPARQL) in order to execute it. The last interface belongs to the formal approaches, as it exhibits a graphically displayed query language. The guided and controlled entry overcomes the habitability problem of NL systems (providing a trade-off between structuredness and freedom) and ensuring all queries make sense in the context of the loaded KB. However, from the four systems, Querix was the interface preferred by the users.

The interface that have the least restrictive and most natural query language, **NLP-Reduce** (Kaufmann, Bernstein and Fischer, 2007), allows almost any NL input (from ungrammatical inputs, like keywords and sentence fragments, to full English sentences). It processes NL queries as bags of words, employing only two basic NLP techniques: stemming and synonym expansion. Essentially, it attempts to match the parsed question words to the synonym-enhanced triples stored in the lexicon (the lexicon is generated from a KB and expanded with WordNet synonyms), and generates SPARQL statements for those matches. It retrieves all those triples for which at least one of the question words occur as an object property or literal, favouring triples which cover most words and with best matches, and joins the resultant triples to cover the query. The second interface **Querix** (Kaufmann, Bernstein and Zumstein, 2006) is also a pattern matching ontology-independent NLI, however, the input is narrowed to full English (grammatically correct) questions, restricted only with regard to sentence beginnings (i.e., only questions starting with "which", "what", "how many", "how much", "give me" or "does"). In contrast with NLP-Reduce, Querix makes use of the syntactical structure of input questions to find better matches in the KB. Querix uses the Stanford parser to analyze the input query, then, from the parser's syntax tree, extended with WordNet synonyms, and identify triple patterns for the query. These triple patterns are matched in the synonym-enhanced KB by applying pattern matching algorithms. When a KB is chosen, the RDF triples are loaded into a Jena model, with the Pellet reasoner to infer all implicitly defined triples, and the WordNet synonym-enhanced triples, including the domain and range specification, pattern matching is done by searching for triples that include one of the nouns or verbs in the query. Querix does not try to resolve NL ambiguities, but asks the user for clarifications in a pop-up dialog menu window to disambiguate. A few dozen triples can be retrieved for the nouns, verbs and their synonyms. Those that matches the query triples are selected. The joining of triples matched in the KB is controlled by domain and range information, and from then, a SPARQL query is generated to be executed in Jena's SPARQL engine.

In the middle of the formality continuum, **GINSENG** (Bernstein, Kauffman et al., 2006) controls a user's input via a fixed vocabulary and predefined sentence structures through menu-based options, as such it falls into the category of *guided input NL* interfaces, similar to **LingoLogic** (Thompson et a., 2005), these systems do not try to understand NL queries but they use menus to specify NL queries in a small and specific domains, in order to combat the habitability problem. GINSENG uses a small static grammar that dynamically extends with elements from the loaded ontologies and allows an easy adaptation to new ontologies, without using any predefined lexicon beyond the vocabulary that is defined in the static sentence grammar and provided by the loaded ontologies. When the user enters a sentence, an incremental parser relies on the grammar to constantly (1) propose possible continuations to the sentence, and (2) prevent entries that would not be grammatically interpretable.

**PANTO** (Wang et al., 2007) is a portable NLI that takes a NL question as input and executes a corresponding SPARQL query on a given ontology model. It relies on the statistical Stanford parser to create a parse tree of the query from which triples are generated. These triples are mapped to the triples in the lexicon. The lexicon is created when a KB is loaded into the system, by extracting all entities enhanced with WordNet synonyms. Following AquaLog model, it uses two intermediate representations: the Query-Triples, which rely solely on the linguistic analysis of the query sentence, and the Onto-Triples that match the query triples and are extracted using the lexicon, string distance metrics and WordNet. PANTO can handle conjunctions / disjunctions, negation, comparatives and superlatives (those that can interpreted with *Order by* and *Limit* on *datatype*, superlatives that require the functionality *count* are not supported).

Similarly, in **QuestIO** (Tablan et al., 2008) NL queries are translated into formal queries but the system is reliant on the use of gazetteers initialized for the domain ontology. In QuestIO users can enter queries of any length and form. QuestIO works by recognizing concepts inside the query though the gazetteers, without relying on other words in the query, it analyzes potential relations between concept pairs and ranks them according to string similarity measures, the specifity of the property or distance between terms. QuestIO supports conjunction and disjunction.

**FREyA** (Damljanovic et al., 2010) is the successor of QuestIO. The improvements of FREyA with respect to QuestIO is a deeper understanding of a question's semantic meaning. QuestIO used a very shallow NLP efficient for small and domain-specific ontologies and ill-formed (or grammatically incorrect) questions but inadequate to handle ambiguities when ontologies are spanning diverse domains (Damljanovic et al., 2010). In FREyA queries are not classified to allow users enter queries in any form, but to identify the answer type of the question and present a concise answer to the users a syntactic parse tree is generated by the Stanford parser. In addition, FREyA assists the user to formulate a query through the generation of clarification dialogs, the user's selections are saved and used for training the system in order to improve its performance over time for all users. Similar to AquaLog's learning mechanism, FREyA uses ontology reasoning to learn more generic rules, which could then be reused for the questions with similar context (e.g., for the superclasses of the involved classes). Given a user query, the process starts with finding ontology-based annotations in the query, if there are ambiguous annotations that cannot be solved reasoning over the context of the query (e.g.: "Mississippi" can be a river or a state) the user is engaged in a dialog. The quality of the annotations depends on the ontology-based gazetteer OntoRoot, which is the component responsible in creating the annotations. The suggestions presented to the user in the clarification dialogs have an initial ranking based on synonym detection and string similarity. Each time a suggestion is selected by the user, the system learns to place the correct suggestions at the top for any similar question. These dialogs also allow translating any additional semantics into the relevant operations (such is the case with superlatives, which cannot be automatically understood without additional processing, i.e.: applying a maximum or minimum function to a datatype property value). Triples are generated from the ontological mappings taking into account the domain and range of the properties. The last step is generating a SPARQL query combining the set of triples.

**Table 4.1.** Ontology-based QA approaches classified by the subset of NL and degree of customization

| Ontology-based QA systems | Subset of NL | | | Customization | | Ontology-independent | | |
|---|---|---|---|---|---|---|---|---|
| | Guided NL | Bag of words | Full shallow grammar | Domain grammar / collection | Domain lexicons | User learning | Relation (Triple) based | Pattern-matching (structural lexicon) |
| QACID | | + | | + | + | | | |
| ORAKEL | | | + | + | + | | | |
| e-Librarian | | | + | | + | | | |
| GINSENG | + | | | | | | | + |
| NLPReduce | | + | | | | | | + |
| Querix | | | + | | | | + | + |
| AquaLog | | | + | | | + | + | - (entity lexicon) |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| PANTO | | | + | | | | + | + |
| QuestIO | | + | | | | | + | + (gazetteers) |
| FreyA | | | + | | | + | + | + |

We have select a representative state-of-the-art NL interfaces over ontologies to deeply understand the advances and limitations on this area. However, this study is not exhaustive[11], and other similar systems to structure knowledge sources exist, such as **ONLI** (Mithun et al., 2006), a QA system used as front-end to the RACER reasoner. ONLI transform the user NL queries into a nRQL query format that supports the <argument, predicate, argument> triple format. It accepts queries with quantifiers and number restrictions. However, from (Mithun et al., 2006) it is not clear how much effort is needed to customize the system for different domains. (Dittenbach et al., 2003) also developed a NL interface for a Web-based **tourism platform**. The system uses an ontology that describes the domain, the linguistic relationships between the domain concepts, and parameterised SQL fragments used to build the SQL statements representing the NL query. A lightweight grammar analyzes the question to combine the SQL statements accordingly. The system was online for ten days and collected 1425 queries (57.05% full input queries and the rest were keywords and question fragments). Interestingly, this study shows that the complexity of the NL questions collected was relatively low (syntactically simple queries combining an average of 3.41 concepts), and traceable with shallow grammars.

Another approach with elaborated syntactic and semantic mechanisms that allow the user to input full NL to underlying KBs was developed by (Frank et al., 2006), **Frank et al. system** applies deep linguistic analysis to the question and transforms it into a ontology-independent internal representation based on conceptual and semantic characteristics. From the linguistic representation, they extract the so-called *proto queries*, which provide partial constraints for answer extraction from the underlying knowledge sources. Customization is achieved through hand-written rewriting rules transforming FrameNet like structures to domain-specific structures as provided by the domain ontology. A prototype implemented for two application domains: the Nobel prize winners and the language technology domains, and tested with a variety of question types (wh-, yes-no, imperative, definition, and quantificational questions), achieved precision rates of 74.1%.

To cope with the slowest pace of the introduction of new knowledge in the semantic repositories, **SemanticQA** (Tartir et al., 2010) allows completing the partial answers given by anontology with web documents to answer a question. SemanticQA assists the users to construct an input question as they type, by presenting valid suggestions in the universe of discourse of the selected ontology, which content has been previously indexed with Lucene. The matching of the question to the ontology is performed by exhaustively matching all word combinations in the question to ontology entities. If a match is not found, WordNet is also used. Then all generated triples are combined into a single SPARQL query. If the SPARQL query fails, indicating that some triple have no answers in the ontology, it attempts to answer those triples by searching in the snippets of the Web search engine, Google. The collection of keywords is gathered from the labels on the ontological entities plus WordNet. The answers are ranked using a semantic answer score, based on the expected type and the distance between all terms in the keyword set. To avoid ambiguity it allows restricting the document search to a single domain (e.g. PubMed if the user is looking for bio-chemical information). A small scale ad-hoc test was performed with only eight samples of simple factoid questions using the Lehigh University Benchmark ontology[12](63% precision), and six sample queries using SwetoDblp ontology (83% precision) (Aleman-Meza et al., 2007).

One can conclude that the similarity techniques used to solve the lexical gap between the users and the structured knowledge are largely comparable across all systems: off-the shelf parsers or shallow parsing to create a triple-base representation of the user query, or to extract the keywords, string distance metrics, WordNet, and heuristics rules to match and rank the different semantic relationships among the ontological mappings.

### *4.2. Limitations of domain-specific QA approaches on the large SW*

Most of the semantic QA systems reviewed in this paper are portable or agnostic to the domain of the ontology, even though, in practice they differ considerably in the degree of domain customization they

[11] See, for example, the EU funded project QALL-ME on multimodal QA: http://qallme.fbk.eu/

[12] http://swat.cse.lehigh.edu/projects/lubm/

require. Regardless of the various fine-grained differences between them, most ontology-aware systems suffer from the following main limitation when applied to Web environment: ***they are restricted to a limited set of domains****:* The domain restriction may be identified by the use of just one, or a set of, ontology(ies) covering one specific domain at a time, or the use of one large ontology which covers a limited set of domains. The user still needs to tell these systems which ontology is going to be used, for instance, in AquaLog the user can select one of the pre-loaded ontologies or load a new ontology into the system (to be queried the ontology is temporarily stored in a Sesame store in memory). Like in NLIDB, the key limitation of all the aforementioned systems is the same limitation already pointed out in (Hirschman et al., 2001), with the exception of FREyA (see Section 4.3) these systems presume that the knowledge the system needs to answer a question is limited to the knowledge encoded in one, or a set of homogeneous ontologies at a time. Therefore, they are essentially designed to support QA in corporate databases or *semantic intranets*, where a shared organizational ontology (or a set of them) is typically used to annotate resources. In such a scenario ontology-driven interfaces have been shown to effectively support the user in formulating complex queries, without resorting to formal query languages. However, these systems remain brittle, and any information that is either outside the semantic intranet, or simply not integrated with the corporate ontology remains out of bounds.

As a result, it is difficult to predict the feasibility of these models to scale to open and heterogeneous environments, where an unlimited set of topics is covered. Nonetheless, next we detail the intrinsic characteristics for these systems, which restrict and challenge, in principle, their suitability to scale to the open SW in the large:

**Domain-specific grammar-based systems:** in these systems grammars are used to syntactically analyze the structure of a NL query and interpret, if there are no linguistic ambiguities, how the terms in a query link to each other. According to (Copestake at al., 1990) it is difficult to devise grammars that are sufficiently expressive and seem reasonably natural and often, they are quite limited with regard to the syntactic structures they are able to understand or domain dependent (although grammars can also be fully domain independent, as it is the case with AquaLog). Nevertheless, according to (Linckels and Meinel, 2006) users tend to use a limited language when interacting with a system interface, so grammars do not need to be complete. Systems like ORAKEL that involve the user in the difficult task to provide a domain-specific grammar, are not a suitable solution in a multi-ontology open scenario.

**Pattern-matching or bag-of-words approaches**: these systems search for the presence of constituents of a given pattern in the user query. As stated in (Kaufmann and Bernstein, 2007) “the more flexible and less controlled a query language is, the more complex a system’s question analyzing component needs to be to compensate for the freedom of query language”. However, naïve and flexible pattern-matching systems work well in closed scenarios, like the NLP-Reduce system, in which complexity is reduced to a minimum by only employing two basic NLP techniques: stemming and synonym expansion. Their best feature is that they are ontology independent and even ungrammatical and ill-formed questions can be processed. Nevertheless, their little semantics and lack of sense disambiguation mechanisms hampers their scalability to a large open scenario. In a non-trivial scenario pattern-matching or bag-of-words approaches (QACID, QuestIO) together with the almost unlimited freedom of the NL query language result in too many possible interpretations of how the words relate together, increasing the risk of not finding correct (SPARQL) translations, and therefore the habitability problem (Kauffman, 2009). As stated in an analysis of semantic search systems in (Hildebrand et al., 2007) “Naïve approaches to semantic search are computationally too expensive and increase the number of results dramatically, systems thus need to find a way to reduce the search space”.

**Guided interfaces:** guided and controlled interfaces like GINO, which generates a dynamic grammar rule for every class, property and instance and present pop-up boxes to the user to offer all the possible completions to the user’s query, are not feasible solutions in a large multi-ontology scenario. As stated in (Kaufmann, 2009) when describing GINO “It is important to note that the vocabulary grows with every additional loaded KB, though users have signaled that they prefer to load only one KB at a time”.

**Disambiguation by dialogs and user interaction:** dialogs are a popular and convenient feature (Kaufmann and Bernstein, 2007) to resolve ambiguous queries, for the cases in which the context and semantics of the ontology is not enough to choose an interpretation. However, to ask the user for assistance every time an ambiguity arises (AquaLog, Querix) can make the system not usable in a multi-domain scenario where many ontologies participate in the QA processes. In FREyA, the suggestions presented on the dialogs are ranked using a combination of string similarity and synonym detection with WordNet and Cyc[13], however, as stated in (Damljanovic et al., 2010): "the task of creating and ranking the suggestions before showing them to the user is quite complex, and this complexity arises [sic] as the queried knowledge source grows".

**Domain dependent lexicons and dictionaries**: high performance can be obtained with the use of domain dependent dictionaries at the expense of portability (as in the e-librarian system), however it is not feasible to manually build, or rely on the existence of, domain dictionaries in an environment with a potential unlimited number of domains.

**Lexicons generated on demand when a KB is loaded**: the efficiency in querying and automatically generating triple patterns (including inferred triples formed applying inherency rules) and big structured lexicons (including WordNet lexically related words independently of their sense) for querying multiple large-scale ontologies simultaneously is on itself a challenging issue. Differently from structured indexes used by PANTO or NLP-Reduce, entity indexes can benefit from less challenging constrains in terms of index space, creation time and maintenance, however, ignoring the remaining context provided by the query terms can ultimately lead to an increase in query execution time to find the adequate mappings.

### *4.3. Open QA over the Semantic Web*

Latest research on QA over the SW focuses on overcoming the domain-specific limitations of previous approaches. The importance of the challenge, for the SW and also NLP communities, to scale QA approaches to the open web, i.e., Linked Data, has been recognized by the appearance of the first evaluation challenge for QA over Linked Data in the 1st workshop on QA over Linked Data (QALD-1)[14].

From the QA systems analyzed in 4.1, FREyA is currently the only one able to query large, heterogeneous and noisy single sources (or ontological graph) covering a variety of domains, such as DBpedia (Bizer, Lehmann et al., 2009).

Similarly, moving into the direction of suitable systems for open domain QA systems, PowerAqua (Lopez, Sabou et al., 2009) evolved from the AquaLog system presented in Section 4.1, which works using a single ontology, to the case of multiple heterogeneous ontologies. PowerAqua is the first system to perform QA over structured data in an open domain scenario, allowing the system to benefit, on the one hand from the combined knowledge from the wide range of ontologies autonomously created on the SW, reducing the knowledge acquisition bottleneck problem typical of KB systems, and on the other hand, to answer queries that can only be solved by composing information from multiple sources.

PowerAqua follows a pipeline architecture, the query is first transformed by the linguistic component into a triple based intermediate format, or Query-Triples, in the form <subject, property, object>. At the next step, the Query-Triples are passed on to the PowerMap mapping component (Lopez, Sabou et al., 2006), which identifies potentially suitable semantic entities in various ontologies that are likely to describe query terms and answer a query. PowerMap uses both WordNet and the SW itself (owl:sameAs) to find synonyms, hypernyms, derived words, meronyms and hyponyms. In the third step, the Triple Similarity Service, exploring the ontological relations between these entities, matches the Query-Triples to ontological expressions specific to each of the considered semantic sources, producing a set of Onto-Triples that jointly cover the user query, from which answers are derived as a list of entities matching the given triple patterns in each semantic source. Finally, because each resultant Onto-Triple may only leads to partial answers, they need to be combined into a complete answer. The fourth component merges and ranks the various interpretations produced in different ontologies. Among other things, merging requires the system to identify entities denoting the same individual across ontologies. Once answers are merged, ranking, based on the quality of mappings and popularity of the answers, can also be applied to sort the answers. As shown in (Lopez, Nikolov, et al., 2009), merging and ranking algorithms enhance the quality of the results with respect to a scenario in which merging and ranking is not applied.

To scale PowerAqua model to an open web environment, exploiting the increasingly available semantic metadata in order to provide a good coverage of topics, PowerAqua is coupled with: a) the Watson SW gateway, which collects and provides fast access to the increasing amount of online available semantic data, and b) its own internal mechanism to index and query selected online ontological stores, as an alternative way to manage large repositories, like those offered by the Linked Data community, often not available in Watson due to their size and format (RDF dumps available as compressed files).

[13] http://sw.opencyc.org

[14] To be held as part of the ESWC 2001: http://www.sc.cit-ec.uni-bielefeld.de/qald-1

### *4.4. Performance of ontology-based QA systems based on their state-of-the-art evaluations*

We look at the performance of the ontology-based QA systems previously presented by looking at the evaluation results carried out in the literature. In contrast to the IR community, where evaluation using standardized techniques, such as those used for TREC competitions, has been common for decades, systematic and standard evaluations benchmarks to support independent datasets, verifications and performance comparisons between systems are not yet in place for semantic search tools. Important efforts have been done recently towards the establishment of common datasets, methodologies and metrics to evaluate semantic technologies, e.g., the SEALS project[15] to assess and compare different interfaces within a user-based study in a controlled scenario. 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. Therefore evaluations are generally small scale with ad-hoc tasks that represent the user needs and the system functionality to be evaluated (Uren et al., 2010), (Sure et al., 2002), (McCool et al., 2005) but fell short of providing a statistical sample over a wide range of users. Although, the different evaluation set-ups and techniques undermine the validity of direct comparisons, they probe the feasibility and reveal the missing or weak features of developing NLIs taking advantage of the semantics provided by the ontologies. We hereby briefly describe the different evaluation methods and performance results. Results are presented in **Table 4.2**.

Evaluations performed on the early days of the SW had to cope with the sparseness and limited access to high quality and representative public semantic data. As a result, to test the AquaLog system (Lopez et al., 2007) two (manually built) rich ontologies were used and the query sets were gathered from 10 users (with almost no linguistic restrictions imposed on the questions). This approach gave a good insight about the effectiveness of the system and to what extent it satisfied user expectations about the range of queries it is able to answer across two different domains. In order for an answer to be correct, AquaLog had to correctly align the vocabularies of both the asking query and the answering ontology (63.5% success). Because of the sequential nature of AquaLog architecture, failures were classified as according to which component caused the system to fail. The major limitations were due to lack of appropriate reasoning services defined over the ontology, e.g. temporal reasoning, quantifier scoping, negations ("not", "other than", "except"), comparatives and superlatives, a limited linguistic coverage (e.g. queries that were too long and needed to be translated into various triples), and lack of semantic mechanisms to interpret a query and the constraints imposed by the ontology structures (e.g., AquaLog could not properly handle anaphoras[16], compound nouns, non-atomic semantic relations, or reasoning with literals).

Alternatively, the criteria success on the retrieval evaluations performed in (Kaufmann, 2009) for NLP Reduce, Querix and Ginseng was measured with the standard IR performance metrics: precision and recall. Failures are categorized according to whether they are due to: 1) "no semantically tractable queries" (Tang and Mooney, 2001) (Popescu et al., 2003), i.e., questions that were not accepted by the query languages of the interfaces or 2) irrelevant SPARQL translations. As such, recall is defined as the number of questions from the total set that were correctly answered (% success), while precision is the number of queries that were correctly matched to a SPARQL query with respect to the number of semantically tractable questions (see Figure 4.1). Thus, the average recall values are lower than the precision values, a logical consequence of the fact that recall is based on the number of semantically tractable questions (those that the system can transform into SPARQL queries, independently if the query produced is appropriate or not). For instance Ginseng has the highest precision but the lowest recall and semantic tractability due to its limited query language (some of the full NL test queries could not be entered into the system). Also, the use of comparative and superlative adjectives in many of the questions decreased the semantic tractability rate in NLP–Reduce, which cannot process them. To enable a comparison, these NLIs were benchmarking with the same three externally sourced test sets with which other NLI systems (PANTO by Wang et al. and the NLIDBs PRECISE by Popescu et al.) were already evaluated. These three datasets are based on the *Mooney NL Learning Data* provided by Ray Mooney and his group from

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

[16] A linguistic phenomenon in which pronouns (e.g. "she", "they"), and possessive determiners (e.g. "his", "theirs") are used to implicitly denote entities mentioned in an extended discourse.

the University of Texas at Austin (Tang and Mooney, 2001) and translated to OWL for the purposes of the evaluation in (Kaufmann, 2009). Each dataset supplies a KB and set of English questions and belonging to one of the following domains: geography (9 classes, 28 properties and 697 instances), jobs (8 classes, 20 properties, 4141 instance) and restaurants (4 classes, 13 properties and 9749 instances).

$$\text{recall} = \frac{\text{number of correct SPARQL queries produced}}{\text{total number of questions}}$$

$$\text{precision} = \frac{\text{number of correct SPARQL queries produced}}{\text{number of semantically tractable questions}}$$

Figure 4.1 Definition of precision and recall by (Kaufmann, 2009)

PANTO assesses the rate on how many of the translated queries correctly represent the semantics of the original NL queries by comparing the output of PANTO with the manually generated SPARQL queries[17]. The metrics used are precision and recall, defined in (Wang et al., 2007) as "precision means the percentage of correctly translated queries in the queries that PANTO produced an output; recall refers to the percentage of queries that PANTO produced an output in the total testing query set". Note that this makes the notion of correctness somewhat subjective, even between apparently similar evaluations. Recall is defined differently between PANTO and the approaches in (Kaufmann, 2009). For (Kaufmann, 2009) recall is the number of questions from the total correctly answered, which is defined as a %success in AquaLog, while for PANTO is the number of questions from the total that produce an output, independently of whether the output is valid or not. Thus, to measure %success (how many NL question the system successfully transformed in SPARQL queries) in PANTO we need to multiply precision by recall and divide it by 100, the results are in **Table 4.2**. There are also some discrepancies in the number of queries in the Mooney datasets between (Kauffman, 2009) and (Wang et al, 2007).

QuestIO was tested on a locally produced ontology, generated from annotated postings in the GATE mailing list, with 22 real user queries that could be answered in the ontology and a Travel Guides Ontology with an unreported number of queries, to demonstrate portability. The initialization time of QuestIO with the Travel Guides ontology (containing 3194 resources in total) was reported to be 10 times longer, which raises some concerns in terms of scalability. A query is considered correctly answer if the appropriate SeRQL query is generated (71.8% success). As in AquaLog, a failed query is when, either no query is being generated or the generated query is not correct.

FREyA is also evaluated using 250 questions from the Mooney geography dataset. Correctness is evaluated in terms of precision and recall, defined in the same way as in (Kaufmann, 2009). The ranking and learning mechanism was also evaluated, they report and improvement of 6% in the initial ranking based on 103 questions from the Mooney dataset. Recall and precision values are very high, both reaching 92.4% (7.6% of failure, e.g. questions with negation).

The system that reports higher performance is the e-Librarian, in an evaluation with 229 user queries 97% were correctly answer, and in nearly half of the questions only one answer was retrieved, the best one, a very high performance. Two prototypes were used: a computer history expert system and a mathematics expert system. The higher precision performance of e-Librarian with respect a system like PANTO reflects the difficulty with precision performance on completely portable systems.

QACID has been tested with an OWL ontology in the cinema domain, where 50 users were asked to generate 500 queries in total for the given ontologies, 348 queries were automatically annotated by an Entity Annotator and queries with the same ontological concepts were grouped together, generating 54 clusters that were manually associated to SPARQL queries. The results reported in an on-field evaluation, where 10 users were asked to formulate spontaneous queries about the cinema domain, show an 80% of precision (80 out of 100 queries were well-answered).

As already mentioned, the different evaluation setups and techniques undermine the validity of direct comparisons, even for similar evaluations as the ones between PANTO and the systems in (Kaufmann, 2009) because the different sizes of the selected query samples and the different notions of evaluating correctness.

These performance evaluations share in common the pattern of being ad-hoc, user-driven and using unambiguous, relatively small and good quality semantic data. Although, they probe the feasibility of developing portable NLIs with high retrieval performance, these evaluations also highlight that the NLIs with better performance usually tend to require a degree of expensive customization or training. As already pointed out in (Damljanovic et al., 2008), to bridge the gap between the two extremes, domain independency and performance, the quality of the

[17] It is not clear how they performed the comparison of the PANTO generated queries with the manually generated ones and if 877 + 238 + 517 SPARQL queries were actually manually generated.

semantic data have to be very high, ensuring a better lexicalization of the ontology and KBs, with enough understandable labels and a good coverage of the vocabulary, or a domain dictionary with synonyms. Nonetheless, as previously reported in AquaLog, and recently evaluated in FREyA, the inclusion of a learning mechanism offers a good trade-off between user interaction and performance, ensuring an increase in performance over time by closing the lexical gap between users and ontologies, without compromising portability.

**Table 4.2.** Performance results of the ontology-based QA systems evaluated in the state of the art

| | Datasets | Nº queries | % Success (S) | | Domain independent |
|---|---|---|---|---|---|
| AquaLog | KMi semantic portal[18] | 69 | **58%**(S) | 63.5% | Yes (NL queries) |
| | Wine and food[19] | 68 | **69.11%**(S) | | |
| NLP Reduce | Geography | 887 | 95.34%(P)/ **55.98%**(S) | 55.3% | Yes (NL and keyword queries) |
| | Restaurants | 251 | 80.08%(P)/ **97.10%**(S) | | |
| | Jobs | 620 | 81.14%(P)/ **29.84%**(S) | | |
| Querix | Geography (USA) | 887 | 91.38%(P)/ **72.52%**(S) | 54.4% | Yes (NL wh-queries) |
| | Restaurants | 251 | 94.31%(P)/ **59.36%**(S) | | |
| | Jobs | 620 | 80.25(P)/ **31.45%**(S) | | |
| Ginseng | Geography (USA) | 887 | 98.86%(P)/ **39.57%**(S) | 48.6% | Yes (guided interface) |
| | Restaurants | 251 | 100%(P)/ **78.09%**(S) | | |
| | Jobs | 620 | 97.77%(P)/ **28.23%**(S) | | |
| PANTO | Geography (USA) | 877 out 880 | 88.05%(P)/ 85.86%(R)= **75.6%**(S) | 80% | Yes (NL queries) |
| | Restaurants | 238 out of 250 | 90.87%(P)/ 96.64%(R)= **87.8%**(S) | | |
| | Jobs | 517 out of 641 | 86.12%(P)/ 89.17%(R)= **76.8%**(S) | | |
| ORAKEL | Geography (Germany) | 454 | **93%** | | Domain-dependent grammar (NL queries) |
| QuestIO | GATE ontology | 22 | **71.88%** | | Yes (NL queries) |
| | Travel guides | No reported | | | |
| e-Librarian | Computer history and mathematics | 229 | **97%** | | Domain-dependent dictionary (NL queries) |
| QACID | Cinema | 100 | **80%** | | Domain-dependent collection NL queries |
| FREyA | Geography | 250 | **92.4%**. | | Yes |

Large ontologies pose additional challenges with respect to usability, as well as performance. The ontologies used in the evaluations are relatively small; allowing to carry out all processing operations in memory, thus, scalability is not evaluated. Linked Data initiatives are producing a critical mass of se-

[18] The akt ontology: http://kmi.open.ac.uk/projects/akt/ref-onto/

[19] W3C, OWL Web Ontology Language Guide: http://www.w3.org/TR/2003/CR-owl-guide- 0030818/

mantic data, adding a new layer of complexity in the SW scenario, from the exploitation of small domain specific ontologies to large generic open domain data sources containing noisy and incomplete data. Thus, two main user-centric evaluations have been conducted to test PowerAqua: before and after using Linked Data, to investigate whether it can be used to exploit the data offered by Linked Data. In the first evaluation (Lopez, Sabou et al., 2009), PowerAqua was evaluated with a total of 69 queries, generated by 7 users, that were covered by at least one ontology in the semantic information space (consisting in more than 130 Sesame repositories, containing more than 700 ontological documents). PowerAqua successfully answered 48 of these questions (69.5%). The second evaluation was focused on scalability and performance when introducing into the previous evaluation setup one of the largest and most heterogeneous datasets in Linked Data, DBpedia (Lopez, Nikolov et al., 2010). The time needed to answer a query depends on two main factors: (1) the total number of (sparql-like) calls send to the ontologies to explore relevant connections between the mappings, which depends directly on the number of semantic sources and mappings that take part in the answering process, and (2) the response times to these calls, which depends on the complexity of the (sparql) queries and the size of the ontology. PowerAqua algorithms were optimized by introducing heuristics to balance precision and recall to analyze the most likely solutions first (iteratively refining candidates only as needed). These heuristics reduced by 40% in average the number of queries sent to the ontologies, however the response times to answer a query increased from 32 to 48 secs. Initial experiments using a different back-end for large-scale sources, i.e. Virtuoso instead of Sesame, reduced the average time to 20 secs. PowerAqua usability as a NL interface to semantic repositories, has also been evaluated following the formal benchmark proposed in SEALS 2010 (Lopez et al., 2011), focused on the usability aspects of different search tools (in particular keyword-based, form-based and NL) within a controlled user study using the Mooney geography dataset. Of the systems tested, PowerAqua was the system with better usability results, evaluated as "good" by the users.

### *4.5. Achievements and competencies*

The main clear advantage of the use of NL search tools is the easy interaction for non-expert users. As the SW is gaining momentum, it provides the basis for QA applications to go beyond the brittle traditional knowledge acquisition task to exploit and reuse the structured knowledge available on the SW. Beyond the commonalities between all forms of QA (in particular for the question analysis), in this section, we analyze the competencies of ontology-based QA with respect to the main traditional forms of QA.

#### *4.5.1. Ontology-based QA with respect to NLIDB*

Since the development of the first QA systems, like LUNAR (Androutsopoulos et al., 1995), there have been improvements in the availability of lexical resources, like WordNet, string distance metrics for name-matching tasks (Cohen et al., 2003) and shallow, modular and robust NLP systems, like GATE (Cunningham t al., 2002) and NLP Parsers, like the Stanford parser. Comparing it with the latest work on NLIDB, the benefits of ontology-based QA are:

- **Ontology independence:** later NLIDB systems (Copestake, et al., 1990) use intermediate representations to have a portable front end with general purpose grammars but the back end is dependent on the database, so normally long configuration times are required to change the domain. Ontology-based QA systems have successfully solved the portability problem, the knowledge encoded in the ontology, together with (often shallow) domain-independent syntactic parsing, are the primary sources for understanding the user query, without the need to encode specific domain-dependent rules. As such, these systems are practically ontology independent, less costly to produce, and it requires little effort to bring in new sources (AquaLog, PANTO, Querix, QuestIO, FREyA). Optionally, on these systems some manual configuration or automatic learning mechanisms based on user feedback can optimize performance.
- **Able to map unknown vocabulary in the user query:** NLIDB systems, like PRECISE (Popescu et al., 2003), require all the tokens in a query to be distinct, questions with unknown words are not semantically tractable and the system fails to provide an answer. In ontology-based QA if a word is lexically dissimilar to the word used by the user, and it does not appear in any manually or automatically created lexicon, the ontology can be used to study the ontology "neighborhood" of the other terms in the query, which may lead us to the value of the term or relation we are looking for. In many cases this

would be all the information needed to interpret a query.

- **Deal with ambiguities:** when ontologies are directly used to give meaning to the queries expressed by the user and retrieve answers, the main advantage is the possibility to link words to obtain their meaning based on the ontological taxonomy, inherit relationships and deal with ambiguities more efficiently.

Summing up, the main benefits of ontology-based QA systems is that they make use of the semantic information to interpret and provide precise answers to questions posed in NL and are able to cope with ambiguities (mapping vocabulary or modifier attachment), in a way that made the system completely portable (i.e. by using disambiguation techniques that are sufficiently general to be applied in different domains).

### *4.5.2. Ontology-based QA with respect to QA on text*

Although most of the state-of-the-art of ontology-based QA still presumes that the knowledge needed is encoded in one ontology in a closed domain scenario, we envision ontology-based QA in an open scenario as complementary to free open QA we envision ontology-based QA in an open SW scenario as complementary to free-text open QA. While the first targets the open, structured SW to give precise answers, the second targets unstructured documents on the Web. Under such a perspective, document search space is replaced by a semantic search space composed of a set of ontologies and KBs, providing a new context in which the results from traditional open QA can be applied. Although, linguistic and ambiguity problems are common in most kinds of NL understanding systems, building a QA system over the SW has the following advantages:

- **Balancing relatively easy design and accuracy**: as seen in Section 3.2 the current state of the art open systems to query documents on the Web require sophisticated syntactic, semantic and contextual processing to construct an answer, including NE recognition (Harabaigiu et al., 2000). These open QA systems classify queries based on hierarchies of question types based on types of answer sought (like person, location, date, or subcategories like lake, river) and filter small text fragments that contain strings with the same type as the expected answers (Moldovan et al., 1999) (Srihari et al., 2004). In ontology-based QA there is no need to build complex hierarchies, to hand map specific answer types to WordNet conceptual hierarchies or to build heuristics to recognize named entities, as the semantic information needed to determine the type of an answer is in the publicly available ontology (ies). As argued in (Mollá and Vicedo, 2007) a major difference between open-domain QA and ontology-based QA is the existence of domain-dependent information that can be used to improve the accuracy of the system.
- **Exploiting relationships for query translation:** NE recognition and IE are powerful tools for free-text QA (Section 3.2.1), although these methods scale well discovering relationships between entities is a crucial problem (Srihari et al., 2004). IE methods do not often capture enough semantics, answers hidden in a form not recognized but the patterns expected by the system could be easily disregarded, and one cannot always rely on WordNet coverage to determine the answer type or the type of the object of the verb in the question (Pasca, 2003). On the contrary, QA systems over semantic data can benefit from exploiting the explicit ontological relationships and the semantics of the ontology schema (e.g., type, subclassOf, domain and range) to understand and disambiguate a query, linking the word meanings and inheriting or inferring relationships. WordNet can be used for query expansion, to bridge the gap between the user vocabulary and the ontology terminology through synonyms and lexically related words.
- **Handling queries in which the answer type is unknown:** *what* queries, in which the type of the expected answer is unknown, are harder than other types of queries when querying free text (Hunter, 2000). However, the ontology simplifies the way to handle what-is queries because the possible answer types are constrained by the types of the possible relations in the ontology.
- **Structured answers are constructed from ontological facts**: arbitrary query concepts are mapped to existent ontology entities, answers are then obtained by extracting the list of semantic entities that comply with the facts, or fulfill the ontological triples or SPARQL queries. The approach for answer extraction in text-based QA requires first identifying entities matching the expected answer in the passages, e.g., using the WordNet mapping approach. Second, select the answers within these relevant passages, e.g., using a set of mainly proximity-based heuristics

whose weights are set by a machine-learning algorithm (Pasca, 2003).

- **Combining multiple pieces of information**: Ontological semantic systems can exploit the power of ontologies as a model of knowledge to give precise, focused answers, where multiple pieces of information (that may come from different sources) need to be inferred and combined together rather than by retrieving pre-written paragraphs of text or answer strings (typically NPs or named entities) extracted verbatim from relevant text passages (Pasca, 2003).

### *4.5.3. Ontology-based QA with respect to proprietary QA.*

It is costly to produce the large amounts of domain background knowledge to provide the semantics, which are required for the proprietary approaches described in Section 3.3, to return information from Web pages and construct structured answers in an open domain scenario. Although based on semantics, these systems do not take advantage of the freely available structured information on the SW, therefore, they cannot be considered as QA systems that reuse the semantic data offered by the open SW. This is a key difference as they impose an internal structure on their knowledge and claim ownership of a trusted and curated homogeneous KB, rather than supporting the user exploring the increasing number of multiple, heterogeneous, distributed and often noisy or incomplete knowledge sources available on the Web.

### *4.6. Open research issues on open QA on the SW*

Evaluations in (Lopez, Nikolov et al., 2010) considered the results encouraging and promising, if one considers the openness of the scenario, and probe, to some extend, the feasibility and potential of the approach. Nonetheless, several issues remain open to any approach that wishes to benefit from exploiting the vast amount of emerging open web data to elicit the most accurate answer to a user query:

- Heterogeneity and openness: the high ambiguity in the sources means that it is not always possible to have enough context to focus on precision when, because of heterogeneity, there are many alternative translations and interpretations to a query. For example, the main issue for PowerAqua is to keep real time performance in a scenario of perpetual change and growth, in particular when both very large heterogeneous sources from the Linked Data cloud, or thousands of small RDF sources from crawled data from Watson are added (Lopez et al., 2011).
- Dealing with scalability as well as knowledge incompleteness: existing techniques focusing simply on effectiveness may not scale to large amounts of data. There are often a huge number (from hundreds to thousands in many cases) of potential ontological hits with different meanings (domains), across and within the same dataset, that can syntactically map the terms in a user query. It is unfeasible to explore all possible solutions to obtain semantically sound mappings, however, filtering and domain-coverage heuristics to shift focus onto precision require making certain assumptions about quality of sources. If filtering heuristics are too strict, recall is dramatically affected in a noisy environment, where sources contain redundant and duplicated terms and incomplete information, either because not all ontological elements are populated at the level of instances or because of a lack of schema information (no domain and range for properties, or type for classes, difficult to parse literals, etc.).
- Sparseness: the potential is overshadowed by the sparseness and incompleteness of the SW when compared to the Web (Polleres, 2010). During the search process, it may happen that a) there are no available ontologies that cover the query, or b) there are ontologies that cover the domain of the query but only contain parts of the answer.

## 5. Related work on open user-friendly querying interfaces for the SW

In the previous sections, we have seen that QA systems have proven to be ontology independent or easily adaptable to new domains, while keeping their efficiency and retrieval performance even when shallow NLP techniques are used. By opening up to the SW scenario, these systems can reach their full potential and enhance or complement traditional forms of QA. In this section we broaden our scope and look at the user-friendly semantic search systems and Linked Data querying interfaces, in search for models, beyond NL QA systems, that can in principle scale enough to open up, and even integrate, heterogeneous data sources on the Web of data.

Many approaches exist to translate user queries into formal queries. Semantic search, a broader area than semantic QA, faces the similar challenges as QA systems when dealing with heterogeneous data sources on the SW. Here, we look at the solutions proposed in the literature for semantic search and how they address semantic heterogeneity from early information systems to the latest approaches for Linked Data and the SW by large.

In Section 6, we further discuss how all QA approaches presented till now and the SW user-friendly querying models presented in this section (based on facets, keywords, etc.) are classified and compared according to the criteria presented in Section 2, and how both research directions can converge into large scale open ontology-based QA for the SW, with the aim to solve the bottlenecks and limitations of both.

### *5.1. Early global-view information systems*

The idea of presenting a conceptually unified view of the information space to the user, the "world-view", has already being studied in (Levy et al, 1995). In early global information systems with well-defined boundaries, the solutions for interfacing and integrating heterogeneous knowledge sources, in order to answer queries that the original sources alone were unable to handle, are based on two approaches (Mollá and Vicedo, 2007): either all the information from multiple sources is extracted to create a unified database, or the set of databases can be seen as a federated database system with a common API, as in (Basili et al., 2004). However, this type of centralized solution that forces users and systems to subscribe to a single ontology or shared model are not transferable to the open-world scenario, where the distributed sources are constantly growing and changing. The manual effort needed to maintain any kind of centralized, global shared approach for semantic mapping is not only very costly, in terms of maintaining the mappings rules in a highly dynamic environment that evolves quickly (Mena et al., 2000), but it also has the added difficulty of "negotiating" a shared model, or API, that suits the needs of all the parties involved (Bouquet et al., 2003).

***Lessons and remaining open issues***: Interestingly, the problems faced by early information systems are still present nowadays. Linked Data assumes re-use of identifiers and the explicit specification of strong inter-dataset linkage in an open distributed fashion, without forcing users to commit to an ontology, but still on the SW the heterogeneity problem can hardly be addressed only by the specification of mapping rules. As stated in (Polleres et al., 2010) "although RDF theoretically offers excellent prospects for automatic data integration assuming re-use of identifiers and strong inter-dataset linkage, such an assumption currently only weakly holds". Therefore, open semantic applications need to handle heterogeneity and mappings on the fly, in the context of a specific task.

### *5.2. Evolution of semantic search on the Web of Data*

Aiming to overcome the limitations of keyword-based search, mainly based on an importance ranking on Web links and statistical methods that study the co-occurrence and frequency of terms, without considering the polysemy of words, semantic search has been present in the IR field since the eighties (Croft, 1986), through the use of domain knowledge and linguistic approaches (thesaurus and taxonomies) to expand user queries. Ontologies were soon envisaged as key elements to represent and share knowledge (Gruber, 1993) and move beyond the capabilities of current search technologies (Guarino et al., 1999). According to (Maedche et al., 2003) and as stated by (Fernandez, Cantador et al., 2010) in the area of semantic-oriented technologies "the most common way in which semantic search has been addressed is through the development of search engines that execute a user query in the KB, and return tuples of ontology values which satisfy the user request". Thus, since the emergence of the SW vision in the late nineties, semantic search using ontologies as a semantic model has been explored till nowadays.

A wide-ranging example is TAP, one of the first keyword-based semantic search systems, presenting a view of the search space where (annotated) documents and ontological concepts are nodes alike in a semantic network. In TAP (Guha et al., 2003) the first step is to map the search term to one or more nodes. A term is searched by using its rdfs:label, or one of the other properties indexed by the search interface. In ambiguous cases it chooses a search term based on the popularity of the term (frequency of occurrence in a text corpus), the user profile, the search context, or by letting the user pick the right denotation. The nodes that are the selected denotation of the search term provide a starting point to collect and cluster all triples in their vicinity (the intuition being that proximity in the graph reflects mutual relevance between nodes). As (Guha et al., 2003) argues any approach that does not require the user to manually specify each class of interest, or set of properties, has the disadvantage of being sensitive to the representational choices made by the semantic sources.

In 2004 the annual SW Challenge was launched, whose first winner was CS Aktive Space (Schraefel et al., 2004). This application gathers and combines a wide range of heterogeneous and distributed Computer Science resources to build an interactive portal. The top two ranked entries of 2005 challenges, Flink (Mika, 2005) and Museum Finland (Hyvonen, 2005) are similar to CS Aktive Space as they combine heterogeneous and distributed resources to derive and visualize social networks and to expose cultural information gathered from several museums respectively. However, there is no semantic heterogeneity and "openness" in them: these tools simply extract information, scraped from various relevant sites, to populate a single, pre-defined ontology. A partial exception is Flink, which makes use of some existing semantic data, by aggregation of online FOAF files.

Later semantic systems with interesting approaches to query interpretation, where keyword queries are mapped and translated into a ranked list of formal queries, include SemSearch (Lei et al., 2006), XXPloreKnow! (Tran et al., 2007) and QUICK (Zenz et al., 2009). For instance, SemSearch supports the search for semantic relations between two terms in a given semantic source, e.g., the query 'news:PhD students' results in all instances of the class news that are related to PhD students. SemSearch and XXPloreKnow! construct several formal queries for each semantic relation or combination of keywords' matches, ranking is used to refine the most relevant possible meanings of keywords and limit the number of different combinations. To go beyond the expressivity of keywords and translate a keyword query into a set of semantic queries that are most likely to be the intended ones by the user, QUICK computes all possible semantic queries among the keywords for the user to select one, in each selection the space of semantic interpretations is reduced, and the query is incrementally constructed by the user.

The approach in (Fazzinga et al., 2010) combines standard Web search queries with ontological search queries. It assumes that Web pages are enriched with annotations that have unique identifiers and are relative to an underlying ontology. Web queries are then interpreted based on the underlying ontology, allowing the formulation of precise complex ontological conjunctive queries as SW search queries. Then these complex ontology queries are translated into sequences of standard Web queries answered by standard Web search. Basically, they introduce an offline ontological inference step to compute the completion of all semantic annotations, augmented with axioms deduced from the annotations and the background ontologies, and an online step that converts the formal conjunctive ontological queries into semantic restrictions before sending it to the search engine.

Different to previous approaches, restricted by a domain ontology, the system presented in (Fernandez et al., 2008) exploits the combination of information spaces provided by the SW and by the (non-semantic) Web, supporting: (i) semantic QA over ontologies and (ii) semantic search over non-semantic documents. First, answers to a NL query are retrieved using the PowerAqua system (Lopez, Sabou et al., 2009). Second, based on the list of ontological entities obtained as a response to the user's query and used for query expansion, the semantic search over documents is accomplished by extending the system presented in (Castells et al., 2007) for annotating documents. The output of the system consists of a set of ontology elements that answer the user's question and a complementary ranked list of relevant documents. The system was evaluated reusing the queries and judgments from the TREC 9 and TREC 2001. However, at that time, only 20% of queries were partially covered by ontologies in the SW. For those queries, where semantic information was available, it led to important improvements over the keyword-base baseline approach, degrading gracefully when no ontology satisfied the query.

***Lessons and remaining open issues***: as argued in (Motta and Sabou, 2006), the major challenge faced by early semantic applications was the lack of online semantic information. Therefore, in order to demonstrate their methods, they had to produce their own semantic metadata. As a result, the focus of these tools is on a single, well-defined domain, and they do not scale to open environments. Later semantic applications set out to integrate distributed and heterogeneous resources, although these resources end up centralized in a semantic repository aligned under a single ontology. Therefore, these approaches follow the paradigm of smart KB-centered applications, rather than truly exploring the dynamic heterogeneous nature of the SW, embracing the SW paradigm (Motta and Sabou, 2006). Furthermore as discussed in (Fazzing at al., 2010), pressing research issues on approaches to semantic search on the Web are on the one hand, the ability to translate NL queries into formal ontological queries (the topic of this survey), and on other hand, how to automatically add semantic annotations to Web content, or alternatively, extract knowledge from Web content without any domain restriction (Fernandez et al., 2008), an important challenge for future work.

### *5.3. Latest work on large scale Semantic Search and Linked Data interfaces*

New technologies have been developed to manipulate large sets of semantic metadata available online. Search engines for the SW collect and index large amounts of semantic data to provide an efficient keyword-base access point and gateway for other applications to access and exploit the growing SW. Falcons (Cheng et al., 2008) allows concept (classes and properties) and object (instance) search. The system recommends ontologies on the basis of a combination of the TF-IDF technique and popularity for concept search, or the type of objects the user is likely to be interested in for object search. Falcons indexes 7 million of well-formed RDF documents and 4,400 ontologies (Cheng et al., 2008). Swoogle (Ding et al., 2005) indexes over 10,000 ontologies, Swoogle claims to adopt a Web view on the SW by using a modified version of the PageRank popularity algorithm, and by large ignoring the semantic particularities of the data that it indexes. Later search engines such as Sindice (Oren et al., 2008) index large amounts of semantic data, over 10 billion pieces of RDF, but it only provides a *look-up* service that allows applications and users to locate semantic documents. Watson (D'Aquin et al., 2007) collects the available semantic content from the Web, indexing over 8,300 ontologies, and also offers an API to query and discover semantic associations in ontologies at run time, e.g. searching for relationships for specific ontological entities. Indeed out of these four ontology search engines, only Watson allows the user to exploit the reasoning capabilities of the semantic data, without the need to process these documents locally. The other engines support keyword search but fail to exploit the semantic nature of the content they store and therefore, are still rather limited to exploit online ontologies dynamically.

Other notable exceptions to this limited-domain approach is some of the search applications demonstrated in the Semantic Web Challenge competitions, and more recently the Billion Triples Challenge (btc)[20], aimed at stimulating the creation of novel demonstrators that have the capability to scale and deal with heterogeneous data crawled from the Web. An example of these applications are SearchWebDB (Wang et al., 2008), the second prize-winner of the btc in 2008, which offers a keyword-based interface to integrated data sources available in the btc datasets. However, as keywords express the user needs imprecisely, the user needs to be asked to select among all possible interpretations. In this system the mappings, between any pairs of data sources at the schema or data levels, are computed a priori and stored in several indexes: the *keyword index*, the *structure index* and the *mapping index*. The disadvantage being that, in a highly dynamic environment, static mappings and complex structural indexes are difficult to maintain, and the data quickly becomes outdated.

The eRDF infrastructure (Gueret at al., 2009) explores the Web of data by querying distributed datasets in live SPARQL endpoints. The potential of the infrastructure was shown through a prototype Web application. Given a keyword, it retrieves the first result in Sindice to launch a set of SPARQL queries in all SPARQL end points, by applying an evolutionary anytime query algorithm, based on substitutions of possible candidate variables for these SPARQL queries. As such, it retrieves all entities related to the original entity (because they have the same type or a shared relationships to the same entity, for example Wendy Hall and Tim Berners Lee both hold a professorship at the university of Southampton).

Faceted views have been widely adopted for many RDF datasets, including large Linked Data datasets such as DBPedia, by using the Neofonie[21] search technology. Faceted views, over domain-dependent data or homogenous sources, improve usability and expressivity over lookups and keyword searches, although, the user can only navigate through the relations explicitly represented in the dataset. Faceted views are also available over large-scale Linked Data in Virtuoso (Erling et al., 2009), however scalability is a major concern, faceted interfaces become difficult to use as the number of possible choices grows. The ranking of predicates to identify important facets is obtained only from text and entity frequency, while semantics associated to the links is not explored.

Mash-ups (Tummarello et al., 2010) are able to aggregate data coming from heterogeneous repositories and semantic search engines, such as Sindice, however these systems do not differentiate among different interpretations of the query terms, disambiguation has to be done manually by the user.

***Lessons and remaining open issues***: these systems have the capability to deal with the heterogeneous data crawled from the Web. However, they have limited reasoning capabilities: mappings are either found and stored a priori (SearchWebdB), or disambiguation between different interpretations is not per-

[20] http://challenge.semanticweb.org/

[21] http://www.neofonie.de/index.jsp

formed (eRDF). The scale and diversity of the data put forward many challenges, imposing a trade-off between the complexity of the querying or reasoning process and the amount of data that can be used. Expressivity is also much limited that the one obtained by using query languages, which hinders the widespread exploitation of the data Web for the non-expert user. Like in both facets and mash-ups, the burden to formulate queries is shifted from the system to the user. Furthermore, they do not perform a semantic fusion or ranking of answers across sources.

## 6. QA on the SW: achievements and research gap

An overview of related work shows a wide range of approaches that have attempted to support end users in querying and exploring the publicly available SW information. It is not our intention to exhaustively cover all existing approaches, but to look at the state of the art and applications to figure out the capabilities of the different approaches, considering each of the querying dimensions presented in Section 2 (sources, scope, search environment and input), to identify promising directions towards overcoming their limitations and filling the research gaps.

### *6.1. Sources for QA and their effect on scalability.*

We have shown through this paper that ontologies are a powerful source to provide semantics and background knowledge about a wide range of domains, providing a new important context for QA systems.

Traditionally, the major drawbacks of intelligent NLIDB systems are that to perform both complex semantic interpretations and achieve high performance, these systems tend to use computationally intensive algorithms for NLP and presuppose large amounts of domain dependent background knowledge and hand-crafted customizations, thus being not easily adaptable or portable to new domains. On the other side, open QA systems over free text require complicated designs and extraordinary implementation efforts, due to the high linguistic variability and ambiguity they have to deal with to extract answers from very large open-ended collections of unstructured text, the pitfalls of these systems arise when a correct answer is unlikely to be available in one document but must be assembled by aggregating answers from multiple ones.

Notwithstanding, we believe that open semantic ontology-based QA systems can potentially fill the gap between closed domain QA over structured sources (NLIDB) and domain independent QA over free text (Web), as an attempt to enhance the limitations of these two different research areas. Ontology-based QA systems are able to handle a much more expressive and structured search space. Semantic QA systems have probe to be ontology independent (Section 4.1) and even able to perform QA in open domain environments by assembling and aggregating answers from multiple sources that are autonomously created (Section 4.3). We look at the capabilities of the different semantic query interfaces, to potentially scale to the Web of data in its entirety and to the Web of documents (see Table 7.1):

- Most ontology-specific QA systems, although ontology-independent, are still limited by the single ontology assumption and they have not been evaluated with large-scale datasets.
- Proprietary QA systems, although they scale to open and large scenarios in a potentially unlimited number of domains, they cannot be considered as interfaces to the SW, as they use their own encoding of the sources. Nonetheless, they are a good example of open systems that integrate structured and non-structured sources, although, currently they are limited to Wikipedia (Powerset, TrueKnowledge) or a set of annotated documents link to the KB (START).
- Although not all keyword-based and semantic search interfaces (including facets) scale to multiple sources in the SW, we are starting to see more and more applications that can scale, by accessing to search engines (e.g., mash-ups), large collections of datasets (i.e., provided by the billion triple challenge), SPARQL endpoints, or various distributed online repositories (previously indexed). We have also seen an example of semantic search approaches (Fazzinga et al., 2010) that can retrieve accurate results on the Web, however it is limited by the single-ontology assumption and it is based on the assumption that documents in the Web are annotated. In (Fazzinga et a., 2010) conjunctive semantic search queries are not formulated yet in NL, logical queries need to be created according to the underlying ontology, making the approach inaccessible for the causal user. DBPedia has also been used as a source for a query completion component in normal Web queries on the mainstream Yahoo search engine (Meij et al., 2009). The current implementation is based on a large but single dataset and the results of a large-scale

evaluation suggested that the most common queries were not specific enough to be answered by factual data. Thus, factual information may only address a relatively small portion of the user information needs.

- Open Semantic QA approaches, as seen in (Fernandez et al., 2008) based on a NL interface to SW repositories and a scalable IR system to annotate and rank the documents in the search space, can in principle scale to the Web and to multiple repositories in the SW in a potentially wide number of domains. However, semantic indexes need to be created offline for both ontologies and documents. Although, also coupled with Watson, its performance with the search engine has not been formally evaluated.

Summing up, the integration of semantic and non-semantic data is an important challenge for future work on ontology-based QA. Current implementations, in particular those based on a limited number of sources, still suffer from the knowledge incompleteness and sparseness problems.

### *6.2. Scope and tendencies towards open QA approaches*

One main dimension over which these approaches can be classified is their scope. In a first level we can distinguish the **closed domain approaches**, whose scope is limited to one (or a set of) a-priori selected domain(s) at a time. As we have seen, ontology-based QA systems, which give meaning to the queries expressed by a user with respect to the domain of the underlying ontology, although portable, their scope is limited to the amount of knowledge encoded in one ontology (they are brittle). As such, they are closer to NLIDB, focused on the exploitations of unambiguous structure data in closed-domain scenarios to retrieve precise answers to questions, than to QA over a document collection or free text, which aims to identify the best text in which an answer can be found in an open document retrieval scenario. While these approaches have proved to work well when a pre-defined domain ontology is used to provide an homogenous encoding of the data, none of them can handle complex questions by combining domain specific information typically expressed in different heterogeneous sources.

In a second level, and enhancing the scope embraced by closed domain models, we can distinguish those **approaches restricted to their own semantic resources**. While successful NL search interfaces to structure knowledge in an open domain scenario exist (popular examples are Powerset, Wolfram Alpha, or TrueKnowledge), they are restricted to the use of their own semi-automatically built and comprehensive factual knowledge bases. This is the most expensive scenario as they are typically based on data that are by and large manually coded and homogeneous.

In a third level, we can highlight the latest **open semantic search approaches.** These systems are not limited by closed-domain scenarios, neither by their own resources, but provide a much wider scope, attempting to cover and reuse the majority of publicly available semantic knowledge. We have seen examples of these different approaches: a) using Linked Data sources, i.e. DBpedia, for a query completion component on the Yahoo search engine, b) keyword-based query interfaces to data sources available in the billion triple challenge datasets and live SPARQL endpoints, c) mash-ups able to aggregate heterogeneous data obtained from the search engine Sindice about a given keyword, d) Open Linked Data facets, which allows the user to filter objects according to properties or range of values, and e) NL QA system over multiple heterogeneous semantic repositories, including large Linked Data sources (i.e. DBpedia) and (with some decrease in performance) the search engine Watson.

We can see that there is a continuous tendency to move towards applications that take advantage of the vast amount of heterogeneous semantic data and get free of the burden of engineering their own semantic data, as predicted by (Motta and Sabou, 2006), heading into a new generation of semantic systems (D'Aquin, Motta et al., 2008), able to reuse all available ontologies in the SW and handle the scalability, heterogeneity and openness issues posed by this new challenging environment.

As such, the next key step towards the realization of QA on the SW is to move beyond domain specific semantic QA to robust open domain semantic QA over structured and distributed semantic data. In this direction the PowerAqua system provides a single NL access approach for all the diverse online resources, stored in multiple collections, opening the possibility of searching and combining answers from all the resources together. Nonetheless, as seen in (Lopez, Nikolov at al., 2009), it is often the case that queries can only be solved by composing information derived from multiple and autonomous information sources, hence, portability is not enough and openness is required. QA systems able to draw precise, focused answers by locating and integrating information, which can be distributed across heterogeneous and distributed semantic sources, are required to go beyond the state of the art in interfaces to query the SW.

### *6.3. Traditional intrinsic problems derived from the search environment*

However, a new layer of complexity arises when moving from a classic KB system to an open and dynamic search environment. If an application wishes to use data from multiple sources the integration effort is non-trivial.

While the latest open Linked Data and semantic search applications shown in 5.3 present a much wider scope, scaling to the large amounts of available semantic data, they perform a shallow exploitation of this information: 1) they do not perform semantic disambiguation, but do need users to select among possible query interpretations, 2) they to not generally provide knowledge fusion and ranking mechanisms to improve the accuracy of the information retrieved to the users, and 3) they do not discover mappings between data sources on the fly, but need to pre-compute them beforehand.

Automatic disambiguation (point 1) can only be performed if the user query is expressive enough to grasp the conceptualizations and content meanings involved in the query, as in NL systems. If the context of the query cannot be used to choose the correct interpretation, the only alternative is to call the user to disambiguate or to rank the different meanings based on the popularity of the answers.

Although ontology-based QA can use the context of the query to disambiguate the user query, it still faces difficulties to face large-scale and heterogeneous environments. The complexity arises because of its "openness", as argued in (Mollá and Vicedo, 2007), QA systems in restricted domains can attack the answer-retrieval problem by means of an internal unambiguous knowledge representation, however, in open-domain scenarios, or when using open-domain ontologies, as is the case of DBpedia or WordNet that map words to concepts, systems face the problem of polysemous words, which are usually unambiguous in restricted domains. On the other hand, 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 without requiring a complex language model. As such, in a large-scale, open scenario the complexity of the tools will be a function of their ability to make sense of the heterogeneity of the data to perform a deep exploitation beyond simple lookup and mash-up services. Also, ranking techniques are crucial to scale to large-scale sources or multiple sources.

With regards to fusion (point 2) only mash-ups and open ontology-based QA systems aggregate answers across sources, however, so far, mash-ups do not attempt to disambiguate between the different interpretations of a user keyword.

With regards to on the fly mappings (point 2), most SW systems analyzed here perform mappings on the fly given a user task, and some of them are able to select the relevant sources on the fly. There are three different mechanisms for the later: (1) through search engines (mash-ups, semantic search, open ontology-based QA), (2) by accessing various distributed online SPARQL end-points providing full text search capabilities (semantic search), (3) by indexing multiple online repositories (open ontology-based QA, semantic search). State of the art open ontology-based QA and semantic search systems perform better by indexing multiple online repositories for its own purposes. When a search engine such as Watson, which provides enough functionality (API) to query and perform a deep analysis of the sources, is used the performance is just acceptable from a research point of view demo (Lopez et al., 2011). More work is needed to achieve real time performance – beyond prototypes, for ontology-based QA to directly catch and query the relevant sources from a search engine that crawls and indexes the semantic sources.
In Table 7.1 we compare how the different approaches to query the SW, translating user information needs while hiding the complexity behind an intuitive and easy-to-use interface, tackle these traditional intrinsic problems derived from the openness of the search environment (automatic disambiguation of user needs, ranking portability, heterogeneity and fusion across sources).

### *6.4. Input and higher expressivity*

Finally, the expressivity of the user query is defined by the input the system is able to understand, as shown in Table 7.1, keyword-based systems lack the expressivity to precisely describe the user's intent, as a result ranking can at best put the query intentions of the majority on top. Most of approaches look at the expressivity at the level of relationships (factoids), however, different systems provide different support for complex queries, from including reasoning services to understand comparisons, quantifications and negations, to the most complex systems (out of the scope of this review) that go beyond factoids and are able to understand anaphora resolution and dialogs (Basili et al., 2007). Ontologies are a powerful tool to provide semantics, and in particular, they can be used to move beyond single facts to answers built from multiple sources, however, regarding the input, ontologies have limited capability to reason about temporal and spatial queries and do not typically store time dependent information. There is a serious research challenge, in determining how to handle temporal data and causality across ontologies. In a search system for the open SW we cannot expect complex reasoning over very expressive ontologies, because this requires detailed knowledge of ontology structure. Complex ontology-dependent reasoning is substituted by the ability to deal with and find connections across large amounts of heterogeneous data.

## 7. Directions ahead

Despite all efforts Semantic Search still suffers from the knowledge incompleteness problem, together with the cost of building and maintaining rich semantic sources and the lack of ranking algorithms to cope with large-scale information sources (Fernandez, Cantador et al., 2010). Due to all this, semantic search cannot yet compete with major search engines, like Google, Yahoo or Microsoft Bing[22].

Nonetheless, by efforts such as the Linked Open Data initiative, the Web of Data is becoming a reality, growing and covering a broader range of topics, and it is likely that soon we will have so much data that the core issues would not be only related to sparseness and brittleness, the necessary terms may not yet exist (Polleres et al., 2010), as to scalability and robustness. Novel approaches that can help the typical Web user to access to the open, distributed, heterogeneous character of the SW and Linked Data are urgently needed to realize the vision of the SW.

Scalability is a major open issue, a study presented in (Lee and Goodwin, 2005) about the potential size of the SW reveals that the SW mirrors the growth of the Web in its early stages, growing at the same rate as the Web in the early 1990. Therefore, semantic systems should be able to support large-scale data sources both in terms of ontology size and the number of them (as of September 2011 Linked Data contained more than 19 billion triples).

While semantic search technologies have been proven to work well in specific domains, they still have to confront many challenges to scale up to the Web in its entirety, in response to a user query need. The latest approaches to exploit the massive amount of distributed SW data represent a considerable advance with respect to previous systems, which restrict their scope to a fraction of the publicly available SW content or that rely on their own semantic resources. These approaches are ultimately directed by the potential capabilities of the SW to provide accurate responses to NL user queries, but are NL QA approaches fit for the SW?.

In this scenario, QA over semantic data distributed across multiple sources has been introduced as a new paradigm, that integrate ideas of traditional QA research into scalable SW tools, for mastering scalability and heterogeneity together with user-friendliness. In our view, there is  great potential for open QA approaches in the SW. As show in

Table **7.1** semantic open QA has tackled more problems than other methods for many of the analyzed criteria. In an attempt to overcome the limitations of search approaches, that restrict their scope to homogenous or domain-specific content, or perform a shallow exploitation of it, current QA systems have developed syntactic, semantic and contextual information processing mechanisms that allow a deep exploitation of the semantic information space.

As such, we believe that open semantic QA is a promising research area that goes beyond the state of the art in user-friendly interfaces to support users in querying and exploring the heterogeneous SW content. Thereby, to:

- Bridge the gap between the end-user and the real SW by providing a NL QA interface that opens up to the Web of data.
- Taking advantage of the structured information distributed on the SW to retrieve aggregate answers to factual queries that extend beyond the coverage of single datasets and are built across multiple ontological statements obtained from different sources. Consequently, smoothing the habitability and brittleness problems (it can only tell you what it already knows) and the knowledge acquisition bottleneck problem intrinsic to closed domain KB systems.

The ultimate goal for a NL QA system in the SW is to answers queries by locating and combining information, which can be massively distributed across heterogeneous semantic resources, without imposing any pre-selection or pre-construction of semantic

[22] Google: http://www.google.com, Yahoo! Search: http://www.yahoo.com, Bing: http://www.bing.com

knowledge, but rather locating and exploring the increasing number of multiple, heterogeneous sources currently available on the Web.

Performance and scalability issues still remain open. Balancing the complexity of the querying process in an open-domain scenario (i.e., the ability to handle: complex questions requiring making deductions on open-domain knowledge, capture the interpretation of domain-specific adjectives, e.g., "big", "small", and in consequence superlatives, e.g., "largest", "smallest" (Cimiano et al., 2009), or combining domain specific information typically expressed in different sources) and the amount of semantic data is still an open problem. The major challenge is, in our opinion, the combination of scale with the considerably heterogeneity and noise intrinsic to the SW. Moreover, information on the SW originates from a large variety of sources and exhibits differences in granularity and quality, and therefore, as the data is not centrally managed or produced in a controlled environment, quality and trust become an issue. Publishing errors and inconsistencies arise naturally in an open environment like the Web (Polleres et al., 2010). Thus, imperfections (gaps in coverage, redundant data with multiple identifiers for the same resource, conflicting data, undefined classes and properties without a formal schema description, invalid or literal datatypes, etc.) can be seen as an inherent property of the Web of data. As such, the strength of the SW will be more a by-product of its size than its absolute quality.

Thus, most research interest has been on factual QA (such as in TREC) over semantic data distributed across multiple sources as the foundations for research on more ambitious or complex forms of QA or multilinguality, see (Burger et al., 2002) for a QA roadmap. In these factual systems the lack of very complex reasoning is substituted by the ability to deal and find connections with large amounts of heterogeneous data to provide coherent answers within a specific context or task. As a consequence, exploiting the real SW is by and large about discovering interesting connections between items. We believe that in those large scale semantic systems, intelligence becomes a side effect of a system's ability to operate with large amounts of data from heterogeneous sources in a meaningful way rather than being primarily defined by their reasoning ability to carry out complex tasks. All the same, most of the large datasets published in Linked Data are light-weight (see http://triplify.org/Overview) OWL-Full reasoning is well-known to be undecidable, while OWL-DL is not suited to reasoning over inconsistent, noisy and potentially massive data (Polleres et al., 2010).

Furthermore, besides scaling up to the SW in its entirety to reach the full potential of the SW, we still have to bridge the gap between the semantic data and unstructured textual information available on the Web. We believe, that as the number of annotated sites increases, the answers to a question extracted in the form of lists of entities from the SW, can be used as a valuable resource for discovering classic Web content that is related to the answers given as ontological entities. Ultimately, complementing the structured answers from the SW with Web pages to enhance the expressivity and performance of traditional search engines with semantic information.

**Table 7.1.** Querying approaches classified according to their intrinsic problems and search criteria

| **Criteria** | **Input** | | **Scope** | | | **Search environment (research issues)** | | | | **Sources** | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | Expres pres-sivity | Reasoning services | Porta-bility | Open Domain | Heteroge-neity | Rank-ing | Disam-biguat. | Fusion | Sources on-the-fly | Scale SW | Scale Web |
| NLIDB | √ | √ | Ø | Ø | Ø | Ø | √ | Ø | Ø | Ø | Ø |
| QA-Text/Web | √ | Ø | √ | √ | √ | √ | √ | Ø | √ | Ø | √ |
| Ontology-QA | √ | √ | √ | Ø | Ø | +/- | √ | Ø | Ø | +/- | Ø |
| Proprie-tary QA | √ | √ | √ | √ | Ø | √ | √ | Ø | Ø | Ø | +/- |
| Keyword-search | +/- | Ø | √ | √ | √ | √ | +/- | Ø | √ | √ | +/- |
| Mash-ups | Ø | Ø | √ | √ | √ | +/- | Ø | √ | √ | √ | Ø |
| Facets | √ | Ø | √ | √ | √ | √ | Ø | Ø | Ø | √ | Ø |
| Semantic open QA | √ | Ø | √ | √ | √ | √ | √ | √ | +/- | √ | +/- |

## Acknowledgments

Research funded under the Smart-Products european project (EC-231204).

## References

Aleman-Meza, B., Hakimpour, F., Arpinar, I.B., and Sheth, A.P. (2007): Swetodblp Ontology of Computer Science Publications. Journal of Web Semantics, 5(3): 151-155.

Androutsopoulos, I., Ritchie, G.D., and Thanisch, P. (1993) MASQUE/SQL - An Efficient and Portable Natural Language Query Interface for Relational Databases. In Proc. of the 6th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems.

Androutsopoulos, I., Ritchie, G.D., Thanisch P. (1995): Natural Language Interfaces to Databases - An Introduction. Natural Language Engineering, 1(1): 29-81.

Basili, R., De Cao, D., Giannone, C. (2007). Ontological modeling for interactive question answering. OTM Workshops(1): 544-553

Basili, R., Hansen, D. H., Paggio, P., Pazienza, M. T., and Zanzotto, F. M. (2004): Ontological resources and question answering. In Workshop on Pragmatics of Question Answering, held jointly with NAACL 2004.

Baeza, R. A., Raghavan, P. (2010): Next generation Web search. In Search Computing, LNCS 5950, p. 11-23, Springer.

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

Bernstein, A., Kauffmann, E., Kaiser, C., and Kiefer, C. (2006). Ginseng: A Guided Input Natural Language Search Engine. In Proc. of the 15th workshop on Information Technologies and Systems.

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

Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., Hellman, S. (2009): DBPedia. A Crystallization Point for the Web of Data. In the Journal of Web Semantics, 7(3).

Bouquet P., Serafini L. and Zanobini S. (2003): Semantic coordination: a new approach and an application. In Proc. of 2nd International Semantic Web Conference, 130-145,

Burger, J., Cardie, C., Chaudhri, V. (2001): Issues, Tasks and Program Structures to Roadmap Research in Question & Answering (Q&A) NIST.

Burke, R., D., Hammond, K., J., Kulyukin, V. (1997) Question Answering from Frequently-Asked Question Files: Experiences with the FAQ Finder system. Tech. Rep. TR-97-05, Department of Computer Science, University of Chicago.

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): 261-272.

Cheng, G., Ge, W., Qu, Y. (2008): Falcons: Searching and browsing entities on the Semantic Web. In Proc. of the International Conference on World Wide Web , p. 1101-1101.

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

Cimiano, P., Minock, M. (2009): Natural Language Interfaces: What Is the Problem? - A Data-Driven Quantitative Analysis. In Proc. of the conference on Natural Language Interfaces to Databases, p.192-206

Cohen, W., W., Ravikumar, P., Fienberg, S., E. (2003) A Comparison of String Distance Metrics for Name-Matching Tasks. Workshop on Information Integration on the Web, IIWeb-03.

Copestake, A., Jones, K., S. (1990): Natural language interfaces to databases. Knowledge Engineering Review. 5(4): 225-249.

Croft. (1986). User-specified domain knowledge for document retrieval. In Proc. of the 9th Annual International ACM Conference on Research and Development in Information Retrieval (SIGIR 1986), p. 201-206.

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.

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.

Damljanovic, D., Tablan, V., Bontcheva, K. (2008): A text-based query interface to owl ontologies. In Proc. of the 6th Language Resources and Evaluation Conference (LREC). .

D'Aquin, M., Baldassarre, C., Gridinoc, L., Angeletou, S., Sabou, M., Motta, E. (2007): Characterizing knowledge on the semantic web with Watson. In Proc. of 5th EON Workshop at International Semantic Web Conference.

D'Aquin, M., Motta, E., Sabou, M., Angeletou, S., Gridinoc, L., Lopez, V., Guidi, D. (2008): Towards a new Generation of Semantic Web Applications. IEEE Intelligent Systems, 23 (3).

De Boni, M. (2001), 'TREC 9 QA Track Overview'.

De Roeck, A., N., Fox, C., J., Lowden, B., G., T., Turner, R., Walls, B. (1991) A Natural Language System Based on Formal Semantics. In Proc of the International Conference on Current Issues in Computational Linguistics.

Ding, L., Pan, R., Finin, T., Joshi, A., Peng, Y., Kolari, P. (2005): Finding and Ranking Knowledge on the Semantic Web. In Proc. of International Semantic Web Conference, p. 156 – 170.

Dittenbach, M., Merkl, D., and Berger, H. (2003): A Natural Language Query Interface for Tourism Information. In Proc. of the 10th International Conference on Information Technologies in Tourism (ENTER-03).

Erling, O., Mikhailov, I. 2009. Faceted Views over Large-Scale Linked Data. Linked Data on the Web (LDOW2009).

Fazzinga, B., and Lukasiewicz. T. (2010) Semantic Search on the Web. SemanticWeb – Interoperability, Usability, Applicability.

Fernandez, M., Cantandor, I., Lopez, V., Vallet, D., Castells, P., Motta, E. (2010): Semantically enhanced Information Retrieval: an ontology-based approach. Journal of Web Semantics. Special Issue on Semantic Search. In Press.

Fernandez, M., Lopez, V., Sabou, M., Uren, V., Vallet, D., Motta, E., and Castells, P. (2008): Semantic Search meets the Web, In Proc. of the International conference on Semantic Computing.

Fernandez O, Izquierdo R, Ferrandez S, Vicedo J. L (2009): Addressing Ontology-based question answering with collections of user queries. Information Processing and Management, 45 (2): 175-188.

Forner, P., Giampiccolo, D., Magnini, B., Peñas, A., Rodrigo, A., and Sutcliffe, R. (2010). Evaluating Multilingual Question Answering Systems at CLEF. *In Proc. of the conference on Int. Language Resources and Evaluation (LREC).*

Frank, A., Hans-Ulrich K., Feiyu, X., Hans, U., Berthold, C.,Brigitte, J., and Ulrich, S. (2006): Question answering from structured knowledge sources. Journal of Applied Logic, Special Issue on Questions and Answers: Theoretical and Applied Perspectives, 1 (29).

Gruber, T. R. (1993). A Translation Approach to Portable Ontology Specifications. Knowledge Acquisition, 5(2), p.199-220.

Guarino, N., Masolo, C., and Vetere, G. (1999): OntoSeek: Content-based Access to the Web. IEEE Intelligent Systems, 14(3): 70-80

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

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

Hallett, C., Scott, D. and Power, R. (2007): Composing Questions through Conceptual Authoring. Computational Linguistics 33 (1).

Harabagiu, S., Moldovan, D., Pasca, M., Mihalcea, R., Surdeanu, M., Bunescu, R., Girju, R., Rus, V., Morarescu, P. (2000) Falcon - Boosting Knowledge for Answer Engines. In Proc of the 9th Text Retrieval Conference (Trec-9),

Hendler, J (2010): Web 3.0: The Dawn of Semantic Search, IEEE Computer, 43 (1), p. 77-80.

Hildebrand, M., Ossenbruggen, J., van Hardman, L. (2007): An analysis of search-based user interaction on the semantic web. Report, CWI, Amsterdam, Holland.

Hirschman, L., Gaizauskas, R. (2001) Natural Language question answering: the view from here. Natural Language Engineering, Special Issue on Question Answering, 7(4) 275-300.

Hovy, E.H., Gerber, L., Hermjakob, U., Junk, M., Lin, C.-Y. (2000) Question Answering in Webclopedia. In Proc of the TREC-9 Conference. NIST.

Hunter, A. (2000): Natural Language database interfaces. Knowledge Management.

Hyvonen, E., Makela, E., Salminen, M., Valo, A., Viljanen, K., Saarela, S., Junnila, M., Kettula, S. (2005): MuseumFinland – Finnish Museums on the SemanticWeb. Journal of Web Semantics, 3(2): 224-241.

Gurevych, I., Bernhard, D., Ignatova, K., and Toprak, C. (2009) Educational Question Answering based on Social Media Content, In Proc. of the 14th International Conf. on Artificial Intelligence in Education, p. 133-140.

Katz, B., Felshin, S., Yuret, D., Ibrahim, A., Lin, J., Marton, G., McFarland A. J., Temelkuran, B. (2002): Omnibase: Uniform Access to Heterogeneous Data for Question Answering. In Proc. of the 7th International Workshop on Applications of Natural Language to Information Systems (NLDB).

Kauffmann, E., Bernstein, A., and Fischer, L. (2007). NLP-Reduce: A "naïve" but Domain-independent Natural Language Interface for Querying Ontologies. In Proc. of the 4th European Semantic Web Conference.

Kauffmann, E., Bernstein, A., and Zumstein, R. (2006). Querix: A Natural Language Interface to Query Ontologies Based on Clarification Dialogs. In Proc. of the 5th International Semantic Web Conference.

Kaufmann, E. (2009): Talking to the Semantic Web - Natural Language Query Interfaces for Casual End-Users. PhD thesis. University of Zurich, Switzerland.

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.

Klein, D. and Manning, C. D. (2002): Fast Exact Inference with a Factored Model for Natural Language Parsing.

Advances in Neural Information Processing Systems. 15, p. 3-10.

Kwok, C., Etzioni, O. and Weld, D. (2001). Scaling question answering to the Web. In Proc. of International Conference on World Wide Web (WWW'10).

Lee, J., Goodwin, R. (2005): The semantic webscape: a view of the semantic web. In the Special interest tracks and posters of the 14th International Conference on World Wide Web (WWW'05).

Lei, Y., Uren, V., and Motta, E. (2006): SemSearch: A Search Engine for the Semantic Web. In Proc. of the 15th International Conference of Knowledge Engineering and Knowledge Management, EKAW.

Levy A., Y., Srivastava D. and Kirk T. (1995): Data Model and Query Evaluation in Global Information Systems. Journal of Intelligence Information Systems. 5(2): 121-143.

Linckels S., M.C. (2005): A Simple Solution for an Intelligent Librarian System. In Proc of the IADIS International Conference of Applied Computing.

Linckels, S., and Meinel, C. (2006). Resolving ambiguities in the Semantic Interpretation of Natural Language Questions. In Proc. of the Intelligence Data Engineering and Automatic Learning.

Litkowski, K. C. (2001) Syntactic Clues and Lexical Resources in Question-Answering. Information Technology: The Ninth Text REtrieval Conference (TREC-9), NIST Special Publication 500-249.

Lopez, V. and Motta, E. (2004): Ontology Driven question answering in AquaLog, In Proc. of the 9th International Conference on Applications of Natural Language to Information Systems (NLDB 2004).

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. on the Asian Semantic Web Conference.

Lopez, V., Nikolov, A., Sabou, M., Uren, V., Motta, E., d'Aquin, M. (2010) Scaling up Question-Answering to Linked Data. In Proc. of the Knowledge Engineering and Knowledge Management by the Masses.

Lopez, V., Sabou, M. and Motta, E. (2006): PowerMap: Mapping the Real Semantic Web on the Fly. In Proc. of the International Semantic Web Conference.

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

Lopez, V., Uren, V., Motta, E. and Pasin, M. (2007): AquaLog: An ontology-driven question answering system for organizational semantic intranets, Journal of Web Semantics, 5 (2), p. 72-105

Lopez, V., Fernndez, M., Motta, E., Stieler, N. (2011): PowerAqua: Supporting Users in Querying and Exploring the Semantic Web. Semantic Web Journal. Demo at kmi.open.ac.uk/technologies/poweraqua/demo.html

Maedche, A., Staab, S., Stojanovic, N., Studer, R., & Sure, Y. (2003): SEmantic portAL: The SEAL Approach. Spinning the Semantic Web. MIT Press , p. 317-359.

Martin, P., Appelt, D., E., Grosz, B., J., Pereira, F., C., N. (1985) TEAM: An Experimental Transportable Natural-Language Interface. IEEE Database Engineering. 8(3):10-22.

Mc Guinness, D. (2004): Question Answering on the Semantic Web. IEEE Intelligent Systems, 19(1).

Mc Guinness, D., van Harmelen, F. (2004): OWL Web Ontology Language Overview. W3C Recommendation http://www.w3.org/TR/owl-features/.

McCool, R., Cowell, A. J., & Thurman, D. A. (2005). End-User Evaluations of Semantic Web Technologies. Workshop on End User Semantic Web Interaction. In Proc. of the International Semantic Web Conference.

Meij, E., Mika, P., Zaragoza, H. Investigating the Demand Side of Semantic Search through Query Log Analysis. In Proc. of the Workshop on Semantic Search at the 18th International World Wide Web Conference (WWW 2009).

Mena E., Kashyap V., Sheth A. and Illarramendi A. (2000) OBSERVER: An approach for Query Processing in Global Information Systems based on Interoperation across Pre-existing Ontologies. Distributed and Parallel Databases 8(2): 223-271.

Mika, P. (2005): Flink: SemanticWeb Technology for the Extraction and Analysis of Social Networks. Journal of Web Semantics, 3(2).

Minock, M. (2010) "C-Phrase: A System for Building Robust Natural Language Interfaces to Databases", Journal of Data Engineering (DKE), Elsevier, 69(3):290-302.

Minock, M., Olofsson, P., Naslund, A. (2008): Towards building robust Natural Language Interfaces to Databases. In Proc. of the 13th international conference on Natural Language and Information Systems .

Mithun, S., Kosseim, L., Haarslev, V. (2007). Resolving quantifier and number restriction to question owl ontologies. In Proc. of The First International Workshop on Question Answering (QA2007).

Moldovan, D., Harabagiu, S., et al. (2002) LCC Tools for Question Answering. In Proc of the TREC-11.

Moldovan, D., Harabagiu, S., Pasca, M., Mihalcea, R., Goodrum, R., Girju, R., Rus, V. (1999) LASSO: A Tool for Surfing the Answer Net. In Proc. of the Text Retrieval Conference (TREC-8).

Moldovan, D., Pasca, M., Harabagiu, S., Surdeanu, M. (2003): Performance issues and error analysis in an open-domain question answering system. ACM Trans. Inf. Syst. 21(2), p. 133-154.

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

Motta, E., and Sabou, M. (2006): Language technologies and the evolution of the semantic web. In Proc. of the Fifth International Conference on Language Resources and Evaluation (LREC).

Oren, E., Delbru, R., Catasta, M., Cyganiak, R., Stenzhor, H., and Tummarello, G. (2008): Sindice.com: A document-oriented lookup index for open linked data. International Journal of Metadata, Semantics and Ontologies, 3(1): 37-52.

Pasca M. (2003): Open-Domain Question Answering from Large Text Collections. CSLI Publications, CSLI Studies in Computational Linguistics.

Polleres, A., Hogan, A., Harth,A., Decker, S. (2010). Can we ever catch up with the Web?. To appear in the Journal of Semantic Web - Interoperability, Usability, Applicability, 1.

Popescu, A., M., Etzioni, O., Kautz, H., A. (2003): Towards a theory of natural language interfaces to databases. In Proc of the 2003 International Conference on Intelligent User Interfaces, p. 149-157.

Schraefel, m.c., Shadbolt, N., Gibbins, N., Glaser, H., and Harris, S. (2004) CS AKTive Space: Representing Computer Science in the Semantic Web. In Proc of the International World Wide Web Conference.

Srihari, K., Li, W., Li, X. (2004) Information Extraction Supported Question- Answering, In Advances in Open-Domain Question Answering. Kluwer Academic Publishers.

Sure, Y., & Iosif, V. (2002): First Results of a Semantic Web Technologies Evaluation. Common Industry Program at the federated event: ODBASE'02 Ontologies, Databases and Applied Semantics.

Tablan, V, Damljanovic, D., and Bontcheva, K. (2008): A Natural Language Query Interface to Structured Information. In Proc of the European Semantic Web Conference.

Tang, L. R., Mooney, R. J. (2001): Using multiple clause constructors in inductive logic programming for semantic parsing. In Proc of the 12th European Conference on Machine Learning (ECML-2001).

Tartir, S., and Arpinar, I. B. (2010): Question Answering in Linked Data for Scientific Exploration. In the 2nd Annual Web Science Conference, .

Thompson, C. W., Pazandak, P., Tennant, H. R. (2005): Talk to Your Semantic Web. IEEE Internet Computing,. 9 (6), p. 75-78.

Tran, T., Cimiano, P., Rudolph, S., and Studer. R. (2007). Ontology-based interpretation of keywords for semantic search. In Proc. of the 6th International Semantic Web Conference.

Tummarello, G., Cyganiak, R., Catasta, M., Danielczyk, S., Delbru, R., and Decker, S.. Sig.ma: Live views on the Web of data. In Proc. World Wide Web Conference (WWW-2010), p. 1301–1304.

Uren, V., Lei, Y., Lopez, V., Liu, H., Motta, E. and Giordanino, M. (2007) The usability of semantic search tools: a review, Knowledge Engineering Review, 22 (4), p. 361-377

Uren, V., Sabou, M., Motta, E., Fernandez, M., Lopez, V., Lei, Y. (2010): Reflections on five years of evaluating semantic search systems. In the International Journal of Metadata, Semantics and Ontologies. 5(2), p.87-98.

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.

Wang, H., Tran, T., Haase, P., Penin, T, Liu, Q., Fu, L., Yu, Y. (2008): SearchWebDB: Searching the Billion Triples! Billion Triple Challenge 2008.

Wu, M., Zheng, X., Duan, M., Liu, T., Strzalkowski, T. (2003) Question Answering by Pattern Matching, Web-Proofing, Semantic Form Proofing. NIST Special Publication: The Twelfth Text REtrieval Conference (TREC), p. 500-255.

Zenz, G., Zhou, X., Minack, E, Siberski, W., Nejdl, W. (2009): From keywords to semantic queries—Incremental query construction on the semantic web. In the journal of Web Semantics: Science, Services and Agents on the World Wide Web, 7(3).