KNOWLEDGE MEDIA

KMi

INSTITUTE

State of the art on Semantic Question Answering

A Literature Review

Technical Report kmi-07-03

May 2007

Vanessa Lopez, Enrico Motta, Victoria Uren and Marta Sabou

The Open University

Literature review and *state of the art* on Semantic Question Answering

Vanessa López,
Supervisors: Enrico Motta, Victoria Uren, and Marta Sabou

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

1 Introduction and outline

The goal of Question Answering (QA) systems as described by [38] is to allow users to ask questions in natural language, using their own terminology, and receive a concise answer, possible with enough validating context. As stated in [82] the different kind of searches mode provides complementary affordances: "For example, keyword search and natural language querying offered easy routes in for users, but don't support exploration of the semantic search space, whereas view based search and forms can help the user explore space, but become tedious to use in large spaces and impossible in heterogeneous ones".

The development of a semantic layer on top of web contents and services, the *Semantic Web* [10], has been recognized as the next step in the evolution of the World Wide Web as a distributed knowledge resource that brings to the Web the idea of having data formally defined and linked. The *semantic web* vision [10] is one in which rich, *ontology-based semantic markup* is widely available, enabling us to improve content by adding meta-information. This means that with structured or semi-structured documents, texts can be semantically marked-up and ontological support for term definition provided, both to enable sophisticated interoperability among agents, e.g., in the e-commerce area, and to support human web users in locating and making sense of information. *Ontologies* play a crucial role on the SW: they provide the conceptual infrastructure supporting semantic interoperability, addressing data heterogeneity [1] and opening up opportunities for automated information processing. However, because of the SW's distributed nature, data will inevitably be associated with different ontologies and therefore ontologies themselves will introduce heterogeneity. Different ontologies may describe similar domains, but using different terminologies, while others may have overlapping domains: i.e. given two ontologies, the same entity can be given different names or simply be defined in different ways. As argued in [66] the two processes of building the require infrastructure to produce a large semantic web and the "smart" applications to exploit such a markup (or at least, demonstrator) have to go hand in hand. Until recently, only a limited amount of semantic data was available, therefore, as stated in [66] "the result of this strategy so far has been that, by and large, the early demonstrators produced in the past few years lack many of the key elements that will characterize "real" SW applications that will operate in an open, large-scale, distributed and heterogeneous environment. Consequently they are more akin to traditional knowledge-based systems". However, as the SW is gaining momentum, there is now a reasonable amount of online distributed semantic data[1].

Language technologies and the Semantic Web can mutually benefit from each other [15]. The relevant literature for the research field of question answering (QA) over multiple ontologies, motivated by the emerging semantic web (SW) scenario, covers and combines many different areas. Therefore, first of all this chapter presents a general background to the research field of question answering systems, its aims, issues, history and typical architecture from natural language interfaces to databases, through an overview of the question answering track in the Text Retrieval Conference (TREC) and current approaches for open domain question answering, to semantic question answering using ontological information.

However, when talking about semantic QA, the goal is to improve traditional methods by exploiting semantic data. These traditional methods involve not only typical QA techniques but also many other research areas, like semantic

[1] (Lee & Goodwin, 2004) registered a 300% growth in 2004 alone and thus outpacing the growth of the web. The semantic search engine Swoogle (Ding et al, 2005) claims to index over 10,000 ontologies (March 07). Also, there is research work on the direction of integrating the widely adopted Folksonomies (tagging) with the formal semantics provided by the ontologies.

similarity measures, ontology selection, ranking and mapping algorithms, play an important role. Here we analyze all these inter-related areas from the point of view and dimensions of QA over semantic data, e.g. from traditional approaches that model semantic similarity by computing the semantic distance between definitions within a single ontology that is either a domain independent ontology or the result of the integration of existing ontologies, to determining semantic similarity among entities from different ontologies.

2 The goals and dimensions of Semantic Question Answering Systems.

In a **roadmap** document for QA [38, 18] five **standards** or criteria for QA systems were identified:

1- Answers should be in real time, regardless of the complexity of the question, the size and multitude of the data sources or the ambiguity of the question.
2- Imprecise, incorrect answers are worse than no answers. The system is expected to deal with contradictions. There will be no answer if this one is not in the knowledge sources.
3- Usability, the knowledge of the system should satisfy the needs of the user
4- Answers distributed across multiple sources that require fusion techniques that combine partial answers from different sources should be coherent
5- The answer should be relevant within a specific context / task. Context can be used to clarify a question and resolve ambiguities. The system evaluation must be user-centered.

The **dimensions** of QA systems are based on the different **sources** used to generate an answer:

1. Natural Language interface to databases use structured data
2. QA over semi-structured data (for instance health records, yellow pages)
3. Open Question Answering (the focus of most current research) over free text, as in the case of the TREC collection or the Web.
4. Over semantic data (ontologies). In addition, the source can be annotated images or even video.

Another distinction between QA applications is whether they are **domain-specific** or **domain-independent**. Traditional, NLIDB are domain-specific while open question answering is domain-independent.

Furthermore, current research in question answering focuses on factual question answering, where we can distinguish between "wh-queries" (who, what, how many, …) or commands (name all …) requiring an element or list of elements as an answer or an affirmation / negation type of questions. As pointed out in [40] more difficult kind of questions include those which ask for opinion, like Why and How questions, which require understanding of causality or instrumental relations, What questions which provide little constrain in the answer type, or definition questions. In [65] QA systems are classified into five increasingly sophisticated **types of questions**: 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.

The **traditional intrinsic problems** that QA systems try to solve are mapping the user terminology into the terminology used by the sources by using similarity measures (which require to handle unknown vocabulary) without affecting portability. Moreover, independently of the type of query, any non trivial NL QA system has to deal with ambiguity. Querying semantic information directly is still a novel research area, however the results from many popular research areas like ontology selection and ranking (how well they satisfy user queries), word sense disambiguation ("squash" can be a sport or a vegetable), co-reference of instances and ontology mapping (e.g., mapping the class "car" in one ontology to the "automobile" class in another) can be applied.

Concerning the **querying environment** there are also problems which are intrinsic to the large and dynamic semantic environment. The complexity arises because of its "openness" in which multiple heterogeneous and distributed ontologies from the same or different domain may describe similar or overlapping topics with different terminologies. A study presented in [82] about the potential size of the SW reveals that the SW is currently growing at the same rate than the Web in the early 1990. As stated on [82] "semantic search systems should be able to support both large-scale ontologies (with many classes and instances) and large scale repositories (with many annotations)". An example of very large ontologies with real world instance data are the SWETO ontology [4] which contains around

800000 entities and 1600000 relationships, and very large RDFs like the conversion of the English Wikipedia into RDF[2], which is a monthly updated dataset containing roughly 47 million triples. An extended discussion of these issues of open semantic environment is presented in [67].

Furthermore, as stated on [82] iterative and exploratory search modes are important to the **usability** of all search systems, as well as the ability to provide justifications for an answer or suggest the presence of unrequested but related information or to propose alternate paths of exploration to the user. Usability is not covered in this review so for extended information see [82]. Also, as stated on [47] “An often proposed solution to help casual users is the use of natural language interfaces. Such tools, however, suffer from one of the biggest problems of natural language interfaces: ambiguities. Furthermore, the systems are hardly adaptable for new domains”. User’s can do a lot more, as they can usually disambiguate between a range of possible factoid questions and answers and / or navigate information clusters in an answer space.

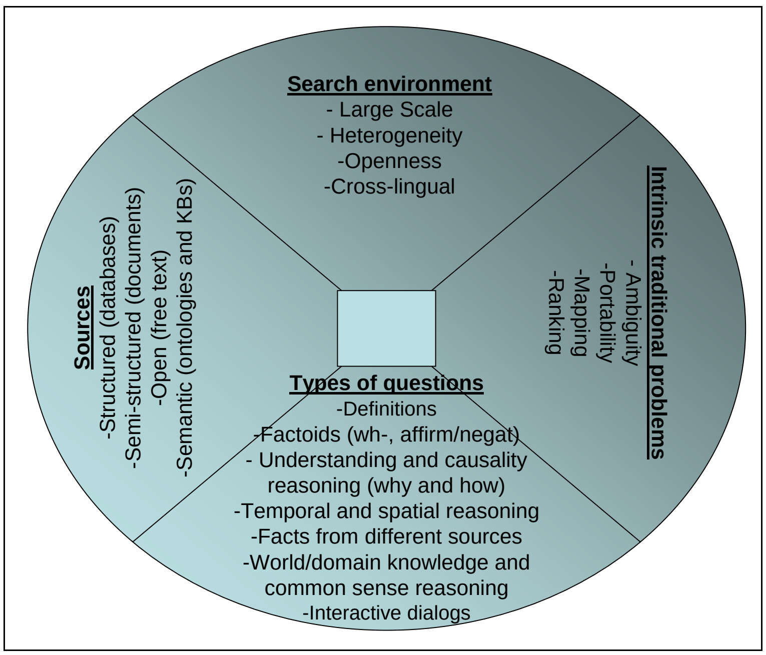

Figure 1. The dimensions of semantic questions answering

Challenges for the future:

NL QA approaches have largely been focused on retrieving the answer from raw text, therefore much work on ontology-driven QA tends to focus on the use of ontologies to support query expansion in information retrieval [59] rather than exploiting semantic metadata. However, we believe that the availability of distributed semantic markup on a large scale opens the way to novel QA systems. The novelty of these systems with respect to traditional QA systems is that they can make use of such semantic information to make sense of the user query to provide precise answers to questions posed in NL, without assuming that the user has any prior information about the semantic resources. In particular, a knowledge based QA system can help with answering questions where multiple pieces of information need to be inferred and combined at run time, rather than simply having a pre-written paragraph of text retrieved [21].

In this review we focus on all the relevant research areas related to the QA dimensions stated on Fig. 1 in an attempt to bridge the **gap** between current QA systems and the future development of QA systems which can search **heterogeneous semantic metadata** on the challenging open semantic web scenario. Or in other words, to move beyond domain specific semantic QA (like traditional NLIDB over structured data) to domain independent open QA over structured semantic information.

[2] http://labs.systemone.at/wikipedia3

Ontologies can not solve all the NLP tasks, as they are very basic in the way they express quantifications and have limited capability to reason about temporal and spational issues. Although simple questions formed with “how-long” or “when”, like “when did the akt project start?” can be handled, ontologies cannot cope with expressions like “in the last year”. This is because the structure of temporal data is domain dependent; compare geological time scales to the periods of time in a knowledge base about research projects. There is a serious research challenge in determining how to handle temporal data and causality in a way which would be portable across ontologies. However, they are a powerful tool to provide semantics, and in particular, they can be used to be able to move beyond single facts to answers built from multiple sources. We predict that intelligent behaviour when questioning the semantic web, will be a side effect of the ability of such systems to handle and find interesting connections, on a very large scale metadata from heterogeneous sources, in a meaningful way.

Our interest lies on QA over semantic data. We define semantic metadata as RDF triples based on some semantic web compatible ontology. In the rest of the paper will be review the state-of-the-art systems and algorithms that covers the different dimensions on the semantic QA scenario with respect to the expected standards. For this review, we just concentrate on factual questions from different sources.

2.1. NLIDB: NATURAL LANGUAGE INTERFACES TO DATABASES

This scenario is of course very similar to asking natural language queries to databases (NLIDB), which 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 [6]. However, it is our view that the SW provides a new and potentially important context in which the results from this area can be applied. The use of natural language to access relational databases can be traced back to the late sixties and early seventies [6] [3]. The first QA systems were developed in 1960s and they were basically NL interfaces to expert systems, tailored to specific domains, being the most famous ones BASEBALL, and LUNAR. Both systems were domain specific, the former answer questions about the US baseball league over the period of one year, the later answer question 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 people. In [6] a detailed overview of the state of the art for these systems can be found.

Most of the early NLIDBs systems were built having a particular database in mind, thus they could not be easily modified to be used with different databases or were difficult to port to different application domains (different grammars, hard-wired knowledge or mapping rules had to be developed). Configuration phases were tedious and required a long time. These earlier QA systems had a core hand-crafted database or knowledge system hand-written by the domain experts.

Some of the early NLIDBs relied on pattern-matching techniques. In the example described by Androutsopoulos in [6], 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”. The shallowness of the pattern-matching would often lead to bad failures but it has also been unexpectedly effective techniques to for exploiting the Web as a data source.

Other approaches are based on statistical or semantic similarity. For shorter answers, NLP-based systems have generally been better than statistical based ones, although the latter are surprisingly competitive. For example, FAQ Finder [19] is a natural language QA system that uses files of FAQs as its knowledge base; it also uses WordNet to improve its ability to match questions to answers, using two metrics: statistical similarity and semantic similarity. However, the statistical approach is unlikely to be useful to semantic QA because it is generally accepted that only long documents with large quantities of data have enough words for statistical comparisons to be considered meaningful [19]. 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.

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, independent of the database structure [6]. TEAM [57] is an

---

[3] Example of such systems are: LUNAR, RENDEZVOUS, LADDER, CHAT-80, TEAM, ASK, JANUS, INTELLECT, BBn’s PARLANCE, IBM’s LANGUAGEACCESS, Q&A Symantec, NATURAL LANGUAGE/ DATATALKER, LOQUI, ENGLISH WIZARD, MASQUE

experimental, transportable NLIDB (front end portable) developed in 1987. The TEAM QA system consists of two major components: (1) for mapping NL expressions into formal representations; (2) for transforming these representations into statements from a database, making a separation of the linguistic process and the mapping process into the KB. To improve portability, TEAM requires "separation of domain-dependent knowledge (to be acquired for each new database) from the domain-independent parts of the system". In TEAM the logical form constructed constitutes an unambiguous representation of the English query.

The latest approach in generation of NLIDB systems based on formal semantics (1991) presented in [26] 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 will only involve altering the domain specific back end.

MASQUE/SQL [5] is a portable NL front-end to SQL databases (1993). 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. However, this still entities an intensive configuration procedure.

More recent work in the area (2003) can be found in [71]. PRECISE [71] maps questions to the corresponding SQL query by identifying classes of questions that are easy to understand 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.). In PRECISE, a lexicon is used to find synonyms. However, in PRECISE the problem of finding a mapping from the tokenization to the database requires that all tokens must be 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 [71], the question "what are some of the neighborhoods of Chicago?" cannot be handled by PRECISE because the word "neighborhood" 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 and a paraphrase was requested.

Further discussion on advantages, disadvantages and challenges with respect to semantic QA:

Since the development of the first rich in NL understanding QA system LUNAR (a syntax-based system where the parsed question is directly mapped to a database expression by the use of rules) there have been improvements in the availability of lexical knowledge bases, such as WordNet, String Distance Metrics for Name-Matching Tasks like the open-source API from Carnegie Mellon University [22], and shallow, modular and robust NLP systems, like GATE [24]. Compare it with the latest work on NLIDB, semantic QA can benefit form the ontology semantics and generic thesaurus in a way that:

(1) Later NLDBI systems use intermediate representations therefore although the front end is portable (as Copestake [23] states "the grammar is, to a large extent, general purpose, it can be used for other domains and for other applications") the back end is dependent on the database, so normally longer configuration times are required. However, in the case of semantic QA systems, the processes, both the linguistic and the one that access to the knowledge bases, can potentially be not only ontology portable but ontology independent, where the configuration to change the domain would be almost zero. Optionally, on these semantic systems some manual configuration can be allowed to optimize performance (i.e. for very specific or precise domains). oreover, the user can be required to be involve in the disambiguation process (when the ontology semantics does not provide further way to disambiguate) and mechanisms to learn from the user interaction and its jargon (i.e. based on case base reasoning) can be implemented.

(2) If a word is lexically very dissimilar to the word used by the user, and it does not appear in any manually or automatic created lexicon, instead of failing to provide an answer, in semantic QA 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

Summarizing, as stated on [46] we can say that most existing NLis to databases that allow full NL input are in almost every case restricted to the domain queried by the database. As [46] states “the applications of NLIs to SW ontologies is not as common as expected, which is surprising given the semantic information that is enclosed in ontologies and could be exploited in query analysis and translation”.

2.2. OPEN DOMAIN 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 (see examples below), is somewhat different in nature from querying ontologies and knowledge bases. The Text Retrieval Conference (TREC) is sponsored by the American National Institute (NIST) and the Defense Advanced Research Projects Agency (DARPA). TREC “tracks” on open domain question answering were introduced in 1999 (TREC-8). The ARDA's Advanced Question and Answering for Intelligence funded AQUAINT program is a multi-project effort to improve the performance of QA systems over free large heterogeneous collections of structured and unstructured text or media. TREC tests are a warm-up for research on more ambitious forms of QA supported by the AQUAINT program. Given the large, uncontrolled text files and the very weak world knowledge available from WordNet and gazetteers and so on, these systems have performed extremely well, e.g. the LCC system [63] that uses a deeper linguistic analysis and iterative strategy obtained an score of 0.856 by answering correctly 415 questions of 500 in TREC-11 (2002).

However, there are linguistic problems common in most kinds of NL understanding systems (for instance, all question understanding and processing systems are required to recognize equivalent questions, regarding idiomatic forms). Some of the shallow methods used keyword-based techniques to locate interesting sentences from the retrieved documents based on the presence of the 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. More sophisticated syntactic, semantic and contextual processing to construct an answer might include: named-entity recognition, relation extraction, coreference resolution, syntactic alternations, word sense disambiguation, logical inferences, temporal or spational reasoning and so on. Furthermore, 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)

The current state-of-the-art open QA systems consist on modules which retrieve information from documents, parse sentences and filter small text fragments that contain strings of the same type as the expected answer, pinpoint question types, analyze semantics, reason, access external resources, extract and rank answers.

Going more in detail, question answering applications for text typically involve two steps, as cited by Hirschman [40]: 1. “Identifying the semantic type of the entity sought by the question”; 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 matching 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 [64, 80, 39,87].

For instance, in LASSO [64] a question type hierarchy was constructed from the analysis of the TREC-8 training data, in which 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..), (c) the question focus, defined as the “main information required by the interrogation” (very useful for “what” questions which say nothing about the information asked for by the question). Furthermore, it identifies the keywords from the question. Occasionally, some words of the question do not occur in the answer (for example, if the focus is “day of the week” it is very unlikely to appear in the answer). Therefore, it implements similar heuristics to the ones used by named entity recognizer systems for locating the possible answers.

Named entity recognition and information extraction (IE) are powerful tools in question answering. One study showed that over 80% of questions asked for a named entity as a response [80]. In that work, Srihari and Li argue that:

> “(i) IE can provide solid support for QA; (ii) Low-level IE is often a necessary component in handling many types of questions; (iii) A robust natural language shallow parser provides a structural basis for handling

questions; (iv) High-level domain independent IE is expected to bring about a breakthrough in QA."

Where point (iv) refers to the extraction of multiple relationships between entities and general event information like WHO did WHAT. As described in [80], NE is necessary but not complete in answering questions because NE by nature only extracts isolated individual entities from text, therefore methods like "the nearest NE to the queried key words" are used.

An efficient system should group together equivalent question types independently of how the query is formulated. Most of the open-domain QA systems classify questions according to their answer target. For example, in Wu et al.'s ILQUA system [86] these are categories like person, location, date, region or subcategories like lake, river.

The best results of the TREC9 [25] competition were obtained by the FALCON system described in Harabaigiu et al. [37] where it generates a score of 58% for short answers and 76% for long answers. In FALCON the answer semantic categories are mapped into categories covered by a Named Entity Recognizer. When the question concept indicating 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 [37], 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.

Other NL search engines such as AskJeeves [7] and EasyAsk [28] exist, which provide NL question interfaces to the Web but retrieve documents, not answers. AskJeeves relies on human editors to match question templates with authoritative sites; systems such as START [44], REXTOR [45] and AnswerBus [89], whose goal is also to extract answers from text.

Further discussion on advantages, disadvantages and challenges with respect to semantic QA:

There are many similarities between open-domains systems and semantic question answering; both would need to attempt to find synonyms, plus their morphological variants, for the terms or keywords. Also in both cases, at times, the rules keep ambiguity unresolved and produce non-deterministic output for the asking point (for instance, "who" can be related to a "person" or to an "organization"). The main two differences between are:

(1) Open domain QA classify queries based on hierarchies of question types based on types of answer sought. In semantic QA there is not need for building hierarchies, or heuristics to recognize named entities, as the semantic information needed is in the ontology.

(2) Semantic systems can greatly benefit from exploiting the ontological relationships in order to understand a query.

(3) In semantic systems the classification on the query can be based on the kind of answer and semantic equivalent representation (relationships between the terms) of the question rather than only on the type of the answer expected (categories like person, location, date, region or subcategories like lake, river). Therefore, there is not need to build complex hierarchies of answers. In a semantic system, different queries, formed by "what is", "who", "where", "tell me", etc., may belong to the same category, e.g. "who works in the semantic web area?" is structurally similar to "which research areas are covered by semantic question answering?", and both require as an answer a list of elements of the type of the ground term that occur in the relationship.

(4) Some kinds of questions are harder than others when querying *free text*. The ontology simplifies the way to handle what-is queries, in which the type of the answer expected in unknown, because the possible answer types are constrained by the types of the possible relations in the ontology.

2.2.1. The use of triples on the literature

The semantic web is represented by triples, so a system to target the Semantic Web should be able to handle and find the ontology triples that can answer a user query. Moreover, for semantic QA to be effective, most of the questions need to be translated into one or more ontology compliant triples. A triple is generally equivalent to a logical form (where the operator is the semantic relation, although is not strictly required). First of all, as pointed out by Katz et al. [44], although not all possible queries can be represented in the binary relational model, in practice these exceptions occur very infrequently. Secondly, RDF-based knowledge representation (KR) formalisms for the semantic web, such as RDF itself [73] or OWL [58] also subscribe to this binary relational model and express statements as <subject, predicate, object>. Hence, it makes sense for a query system targeted at the semantic web to adopt a triple-based model that shares the same format as many millions of other triples on the Semantic Web.

START come online in 1993 as the first QA system available on the Web. START focuses on questions about geography and the MIT infolab. START adopts a triple-base data model called "object-property-value". Using an example presented in [44]: "what languages are spoken in Guernsey?", for START the property is "languages" between the Object "Guernsey" and the value "French". However, the NL is not always semantically sound, the question "Who invented the transistor?" yields two answers: the inventor of the transistor, but also a description about it.

The system described in Litkowski et al. [50], called DIMAP, extracts "semantic relation triples" after the document is parsed and the parse tree is examined. 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 which is "the word in the sentence that the discourse entity stood in relation to". The parsing process generated an average of 9.8 triples per sentence. The same analysis was 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. DIMAP-QA converts the document into triples. The discourse entities are the driving force in DIMAP triples, key elements (key nouns, key 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.

PiQASso [8] uses a "coarse-grained question taxonomy" consisting of person, organization, time, quantity and location as basic types plus 23 WordNet top-level noun categories. The answer type can combine categories. For example, in questions made with who, where the answer type is a person or an organization. Categories can often be determined directly from a wh-word: "who", "when", "where". In other cases additional information is needed. For example with "how" questions the category is found from the adjective following "how" ("how many" or "how much") for quantity, "how long" or "how old" for time, etc. The type of "what <noun>" question is normally the semantic type of the noun which is determined by WNSense, a tool for classifying word senses. Questions of the form "what <verb>" have the same answer type as the object of the verb. "What is" questions in PiQASso also rely on finding the semantic type of a query word. However, as the authors say "it is often not possible to just look up the semantic type of the word, because lack of context does not allow identifying *(sic)* the right sense". Therefore, they accept entities of any type as answers to definition questions, provided they appear as the subject in an "is-a" sentence. If the answer type can be determined for a sentence it is submitted to the relation matching filter during this analysis, the parser tree is flattened into a set of triples and certain relations can be made explicit by adding links to the parser tree. For instance in [8] the example "man first walked on the moon in 1969" is presented, in which "1969" depends on "in", which in turn depends on "moon". Attardi et al. propose short circuiting the "in" by adding a direct link between "moon" and "1969" following a rule that says that "whenever two relevant nodes are linked through an irrelevant one, a link is added between them".

2.4. SEMANTIC QUESTION ANSWERING OVER ONTOLOGIES

We have already mentioned that many systems simply use an ontology as a mechanism to support query expansion in information retrieval. In contrast with these systems, the systems reviewed here provide answers, derived from semantic annotations, to queries expressed in NL. We look into the understanding and query disambiguation process to map NL queries into ontological queries at syntactic and semantic level.

AquaLog is one of the first's ontology-based NL QA systems to query the semantic mark-up. The novelty of the system with respect to traditional QA systems is that it relies on the knowledge encoded in the underlying ontology and its explicit semantics to disambiguate the meaning of the questions and provide answers, and as pointed on the first AquaLog conference publication [51] it provides a new ‘twist’ on the old issues of NLIDB. To bridge between the terminology used by the user and the concepts used by the underlying ontology. In a first step, by using linguistic techniques based on GATE [24], the system classifies and breaks up the question into binary linguistic triples. Then, these terms are linked and mapped to ontology elements, generating ontology compliant triples from where the answer is derived. At the syntactic level AquaLog identifies ontology mappings for all the terms and relations in the linguistic triple by considering the entity labels. It relies on simple, *string-based* comparison methods (e.g., *edit distance metrics*) and WordNet to look-up lexically related words (synonyms). At the semantic level, AquaLog’s interactive *ontology based relation similarity service* uses the ontology taxonomy and relationships (domain knowledge) to semantically disambiguate between two or more possible terms or relations in order to interpret a query, and to link the ontology triples between themselves 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 is agnostic to the domain of the ontology that it exploits and it needs very little customization being almost ontology independent. While AquaLog is ontology portable, the user still needs to tell the system which ontology is going to be used. Furthermore, AquaLog includes a learning component, which ensures that the performance of the system improves over time, in response to the particular community jargon used by the end users.

Some major limitations identified in AquaLog so far are due to linguistic and semantic coverage, lack of appropriate reasoning services or lack of enough mapping mechanism to interpret a query, and the constraints imposed by the ontology structures. For instance, AquaLog does not yet fully exploit quantifier scoping (“each”, “all”, “some”) and negative words (such as “not” and “never” or implicit negatives such as “only”, “except” and “other than”), and as AquaLog is used for stand-alone questions, it does not handle the linguistic phenomenon called anaphora, in which pronouns (e.g. “she”, “they”), and possessive determiners (e.g. “his”, “theirs”) are used to denote implicitly entities mentioned in an extended discourse. Also AquaLog is limited to queries that are not translated into more than two triples (for example in the question “which researchers have publications in KMi related to social aspects”, where the linguistic triples obtained are: <researchers, have publications, KMi> <which is/are, related, social aspects>, the relation “have publications” refers to “publication: has author” in the ontology, therefore, we need not two but three triples to represent three relationships between the terms: <research, ?, publications> <publications, ?, KMi> <publications, related, social aspects>). AquaLog similarity services do not detect compound nouns (i.e. “French researchers” is in fact a compound noun as there is an implicit relation between “French” and “researchers”) and at the moment AquaLog cannot associate linguistic relations to non-atomic semantic relations. In other words, it cannot infer an answer if there is not a direct relation in the ontology between the two terms implied in the relationship, even if the answer is in the ontology. Also, the ontology design determines the questions which may be successfully answered. For example, when one of the terms in the query is not represented as an instance, relation or concept in the ontology but as a value of a relation (string).

Orakel [4] [20] is a NL interface which translates wh-queries into F(rame)-logic and evaluates them with respect to a given knowledge base. 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[5] it uses an underlying syntactic theory built on a variant of *Lexicalized Tree Adjoining Grammar* (LTAG), extended to include ontological information. The scope of ORAKEL is limited to the domain lexicon and the F-Logic KB. 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. and thus is domain independent. The domain lexicon varies from application to application and its generated partially automatically for each application out of the KB (the ontological lexicon is derived fully automatically from the domain ontology loaded in the system, it relies on the labels of the concepts and instances to generate appropriate grammar trees) and by a domain expert. The only lexicon component which has to be generated by the user is the *domain-specific lexicon* in which verbs, adjectives and relational nouns are mapped to corresponding relations specified in the domain ontology. 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 obviously assumes that the domain ontology is somehow

[4] http://ontoware.org/projects/orakel/

[5] Factual or factoid in this context means that answers are ground facts as typically found in KBs

aligned to the foundational categories provided by the foundational ontology. In other words, the user 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 user 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 theWordNet synsets (in the most frequent sense) that best generalize the selectional preferences at each argument in the subcategorization frame.

Another semantic search engine, which understands the sense of the user query to retrieve multimedia resources from a knowledge base, is the “Librarian” found in [49]. First, the NL query is translated into an unambiguous logical form and then the query is mapped to an ontology to generate a semantic query and solve ambiguities. It relies on simple, *string-based* comparison methods (e.g., *edit distance metrics*) and a domain dictionary to look-up lexically related words (synonyms). General purpose dictionaries like WordNet are not appropriate for very specific domains. On the other hand, creating a dictionary is very expensive in terms of portability but the strong advantage of this is a very high performance (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/his question.

As stated on [11] “the SW provides a stable scaffolding for machine-based processing but the casual user is typically overwhelmed by the formal logic of the Semantic Web”. To bridge the gap between the end user and the SW’s capabilities they created GINO a *guided input natural language editor* that allows user to edit and query ontologies in a quasi-NL English. It uses a small static grammar which dynamically extends with elements from the loaded ontologies and allows an easy adaptation to new ontologies. GINO provides a guided and controlled entry to overcome the habitability problem of NL systems. GINO relies on a static sentence structure grammar used to parse sentences which is dynamically extended based on the structure and vocabulary of the loaded ontologies. When the user enters a sentence, an incremental parser relies on the grammar to constantly check the user entry to (1) propose possible continuations to the sentence and (2) prevent entries that would not be grammatically interpretable. GINO translates quasy English sentences into new triple sets or SPARQL statements. However, the major different between GINO and NLIs is that GINO does not use any predefined lexicon beyond the vocabulary that is defined in the static sentence structure grammar and provided by the loaded ontologies. Furthermore, the vocabulary is closed and limited, it does not try to semantically understand the entries or go beyond the vocabulary defined in the grammar and the ontologies.

The approach in domain independent systems like AquaLog and QUERIX to allow NL queries and address ambiguities is asking the user for clarification whenever needed.

In the paper by R. Basili [9], the possibility of building an ontology-based question answering system in the context of the semantic web is discussed. Their approach is being investigated in the context of EU project MOSES, with the “explicit objective of developing an ontology-based methodology to search, create, maintain and adapt semantically structured Web contents according to the vision of semantic web”. As part of this project, they plan to investigate whether and how an ontological approach could support QA across sites. They have introduced a classification of the questions that the system is expected to support and see the content of a question largely in terms of concepts and relations from the ontology. Basili et al. say that they will investigate how an ontological approach could support QA across a “federation” of sites within the same domain: “since each node has its own version of the domain ontology, the task of passing a question from node to node may be reduced to a mapping task between (similar) conceptual representations”. However, to support QA across ontologies they use the XeOML [70] mapping language to allow them to manually specify simple as well as complex mappings a priori between two ontologies describing the same domain.

The knowledge-based approach described in [21] is to “augment on-line text with a knowledge-based question-answering component, […] allowing the system to infer answers to users’ questions which are outside the scope of the prewritten text”. It assumes that the knowledge is stored in a knowledge base and structured as an ontology of the domain. Like many of the systems we have seen, it has a small collection of generic question types which it knows how to answer. Question types are associated with concepts in the KB. For a given concept, the question types which are applicable are those which are attached either to the concept itself or to any of its superclasses. A difference between this system and others we have seen is the importance it places on building a scenario, part assumed and part specified by the user via a dialogue in which the system prompts the user with forms and questions based on an answer schema that relates back to the question type. The scenario provides a context in which the system can answer the query. The user input is thus considerably greater than we would wish to have in an open semantic scenario.

Further discussion on advantages, disadvantages and challenges with respect to semantic QA:

The aim is to provide a system which does not require users to learn specialized vocabulary, or to know the structure of the knowledge base, but as pointed out in [23], though they have to have some idea of the contents of the domain they may have some misconceptions. Therefore, to be realistic, some process of familiarization at least is normally required. Moreover, as indicated in [23], it is difficult to devise a sublanguage which is sufficiently expressive yet avoids ambiguity and seems reasonably natural.

We see ontological semantic systems, like AquaLog, to be complementary to open domain QA systems. Open QA systems use the Web as the source of knowledge and provide answers in the form of selected paragraphs (where the answer actually lies) extracted from very large open-ended collections of unstructured text. The key limitation of an ontology-based system (as for closed-domain systems) is that it presumes the knowledge the system is using to answer the question is in a structured knowledge-base in a limited domain. However, ontological semantic systems exploit the power of ontologies as a model of knowledge and the availability of semantic markup offered by the SW to give precise, focused answers rather than retrieving possible documents or pre-written paragraphs of text. In particular, semantic markup facilitates queries where multiple pieces of information (that may come from different sources) need to be inferred and combined together. For instance, when we ask a query such as "what are the homepages of researchers who have an interest in the Semantic Web?", we get the precise answer. Behind the scenes, AquaLog is not only able to correctly understand the question but is also competent to disambiguate multiple matches of the term "researchers" on the SW and give back the correct answers by consulting the ontology and the available metadata. As Basili argues in [9] "open domain systems do not rely on specialized conceptual knowledge as they use a mixture of statistical techniques and shallow linguistic analysis. Ontological Question Answering Systems […] propose to attack the problem by means of an internal unambiguous knowledge representation".

Probably, the main benefits of an ontology-based QA system on the SW, when compared to other type of QA systems, is that it can use the domain knowledge provided by the ontology to cope with words apparently not found in the KB and to cope with ambiguity (mapping vocabulary or modifier attachment) in a way that made the system completely portable. For instance, the AquaLog disambiguation techniques are sufficiently general to be applied in different domains. In AquaLog, disambiguation is regarded as part of the translation process, if the ambiguity is not solved by domain knowledge, then the ambiguity is detected and the user is consulted before going ahead (to choose between alternative reading on terms, relations or modifier attachment).

As stated on [11] the major drawbacks of these systems are the domain-dependency of intelligent NLI that tend to require long amounts of domain specific knowledge to perform both complex semantic interpretations and high performance, and also the habitability problem[6]. Both of them account for the fact that we are still far from the successful use of full NL interfaces to the SW. GINO tries to solve the habitability problem by using a guided and controlled NLI.

However, we believe that the main limitation of the *state-of-the-art* semantic systems is that although portable (the system being agnostic to the domain of the underlying ontology) the scope of these systems are limited to the amount of knowledge encoded in one ontology (AquaLog, ORAKEL, GINO, e-librarian) or set of a priori defined ontologies in the same domain (Basili et al.). One way to overcome these limitations is to extend semantic systems in the direction of open question answering. So, the lack of very complex reasoning is substituted by the ability to deal and find connections with large amounts of heterogeneous data. However, in multi-ontology SW scenario, neither to involve the user in the difficult task to provide a domain-specific grammar for the system like in ORAKEL, nor the use of 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 feasible solutions in a multi-ontology scenario. So, by using n-ontologies the habitability problem is partially reduced as the brittleness problem (it can only tell you what it already knows), which is typical of ontology driven systems, is also reduced. However, the usability of the system, so the user knows what subset on NL is possible to ask and the system

[6] the habitability problem is the mismatch between the user's expectations and the capabilities of the NL system [32 ester].

can actively help them to refine or recommend the searches, remains an open issue.

Therefore, the goal of the next generation of semantic systems should be to use all available ontologies on the SW to produce an answer to a query, where the system has to deal not only with the heterogeneity introduced by ontologies themselves but also, and in contrast to Basili approach, on the semantic web the systems can not assume that the relevant ontologies will refer to the same domain; they can have overlapping domains or may refer to different domains and even have very dissimilar structures.

To solve questions in which the system must be able to carry out reasoning based on the information present in the ontology, like to answer "what are the most successful projects on the SW?", the system needs ranking services that must be able to manipulate both general and ontology-dependent knowledge, as for instance, the criteria which make a project successful could be the number of citations or the amount funding. Therefore, the first problem identified is how can we make these criteria as ontology independent as possible and available to any application that may need them. An example of deductive components with an axiomatic application domain theory to infer an answer can be found in [84], however, the approach presented there is not portable to other domains. The implementation of temporal structures, causality and answer questions that requires ranking services without scarifying portability are still open research issues out of the scope of this review.

3 *State-of-the-art* on addressing the intrinsic problems for QA over the Semantic Web

All the semantic systems built so far are domain specific and portable. However, it is often the case that queries can only be solved by composing information derived from multiple and autonomous information sources. To build a multi-ontology QA system portability is no longer enough and "openness" is required. Open domain semantic QA brings up new challenges that have to be solved in order to interpret a query by means of different ontologies, among others, the ability to consult and aggregate information derived from multiple heterogeneous semantic sources in real time, without making any assumptions about the ontological structure of the relevant information, plus the performance and scalability issues

The big challenge arises from the combined issues of heterogeneity and scale seen in the real world semantic data, which requires the work on mapping, ranking techniques, etc, to be further advanced to balance search across multiple large ontologies with giving results in real time. First of all in a heterogeneous, open domain scenario it is not possible to determine in advance which ontologies will be relevant to a particular query or make assumptions to find relevant information a priori, so the areas of automatic ontology selection and ranking of the potentially relevant ontologies becomes relevant. Second, user terminology may be translated into semantically sound terminology distributed across ontologies (relevant information from different repositories is combined to generate a complete ontological translation for the user queries); in any strategy that focuses on information content, the most critical problem is that of different vocabularies used to describe similar information across domains [60]. Here results on mapping and Word Sense disambiguation techniques can be applied to avoid incoherent constructions (e.g. "conference chair has four legs") so the concepts which are shared by assertions taken from different ontologies must have the same sense. Finally, as queries may have to be answered not by a single knowledge source but by consulting multiple sources identifying common objects is a requirement. Among other things, this requires the ability to recognize whether two instances coming from different sources may actually refer to the same individual or co-reference of instances.

Furthermore, as stated in [66], when operating at scale on a large and distributed SW, it becomes much more difficult, if not impossible, ensuring strict data quality. The strength of the SW will be more a by-product of its size than its absolute quality (light weigh approach to ensure data quality). In [66], the authors predict a continuous tendency to move towards applications that utilize existing semantic data rather than having to generate their own. Because these data will be heterogeneous, the complexity of the tools will be a function of their ability to make sense of such heterogeneity.

In this section of the review we will go in detail through these related research areas that are relevant to the various issues of resource selection, heterogeneous vocabularies and combination of knowledge from different sources, highlighting their advantages and limitations under the perspective of semantic QA.

3.1. QUERYING GLOBAL INFORMATION SYSTEMS AND SEMANTIC SEARCH

Most of current *state-of-the-art* do not target the need of integrating information from multiple documents or sources. Here, we look at the solutions proposed in the literature to address semantic heterogeneity in information systems.

The Semantic Knowledge Articulation Tool (SKAT) [62] uses a first order logic notation to specify declarative matching rules between ontology terms. SKAT initially attempts to match nodes in the two graphs based on their labels and their structural similarity. The idea of presenting a conceptually unified view of the information space to the user, the world-view, is studied in [48]. The user can pose declarative queries in terms of the objects and the relations in the world-view. Given a query to the world-view, the query processor in the global information system poses subqueries to the external sources that contain the information relevant to answer a query. In order to do that, the semantics of the contents of the external sites is related to the world-view through the use of a description language. These solutions have an intrinsic limitation to be applied to the open-world domain introduced by the SW scenario, where the distributed sources are constantly growing. And therefore, it is not possible to apply any closed-domain solution for environments with well-defined boundaries, like corporate intranets, in which the problem can be addressed by the specification of shared models like mapping rules, global ontologies/vocabularies, and definitions of conversion libraries or functions between semantic data/values, among others. The manual effort needed to maintain any kind of centralized/global shared approach for semantic mapping in the SW is not only very costly, in terms of maintaining the mappings for such a highly dynamic environment that evolves quickly, but also has the added difficulty of "negotiating" a shared model that suits the needs of all the parties involved [13].

In Query Processing in Global Information Systems [60] user queries are rewritten by using inter-ontology relationships to obtain semantic translations across ontologies. There are two restrictions: firstly the user must subscribe to the terminology and model captured by a chosen ontology. Secondly, the solution to the vocabulary problem is obtained through the declarative representation of synonym relationships relating ontology terms. The disadvantages are: 1) synonym relationship mappings must be maintained between terms in the user ontology and the underlying repositories. 2) Every time there is a change in the structure of underlying repositories the mappings of the component ontology must be changed. 3) Such synonym relationships should be defined when a new ontology is added to the system (its centralized nature may affect the efficiency of the system). Therefore, to force systems to commit to a single ontology is costly if not impractical The advantage is that different partial answers can be easily correlated since all of them are expressed in the language of the user ontology.

CUPID [55] analyzes the factors that affect effectiveness of algorithms for automatic semantic reconciliations; however, this is a complementary goal to ours: our system matches terms and relations in an user's query with distributed ontologies while they match data repositories and ontologies. In GLUE [27] the probability of matching two concepts is studied by analyzing the available ontologies using relaxation labeling methods; however, this approach is not very adaptable because it analyzes all the ontology concepts. Finally, in a QA-driven scenario there is no need for obtaining mappings for each pair of concepts belonging to different ontologies, in which the level of effort is at least linear in the number of matches to be performed [55].

Semantic web searches face the same problems as open domain systems in regards to dealing with heterogeneous data sources on the Semantic Web. Guha et al. [72] argue that "in the Semantic Web it is impossible to control what kind of data is available about any given object [..] Here, there is a need to balance the variation in the data availability by making sure that at a minimum, certain properties are available for all objects. In particular data about the kind of object and how is referred to, i.e., rdf:type and rdfs:label". In the TAP system [72] search is augmented with data from the SW. To determine the concept denoted by the search query the first step is to map the search term to one or more nodes of the SW (this might return no candidate denotations, a single match or multiple matches). 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 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 node which is the selected denotation of the search term provides a starting point. Triples in the vicinity of each of these nodes are collected. As [72] argues:

> "the intuition behind this approach is that proximity in the graph reflects mutual relevance between nodes. This approach has the advantage of not requiring any hand-coding but has the disadvantage of being very sensitive to the representational choices made by the source on the SW [...]. A different approach is to manually specify for

each class object of interest, the set of properties that should be gathered. A hybrid approach has most of the benefits of both approaches".

Therefore, in a portable ontology-based system for the open SW scenario we can not expect complex reasoning over very expressive ontologies, because this requires detailed knowledge of ontology structure. A similar philosophy was applied to the design of the query interface used in the TAP framework for semantic searches [72]. This query interface, called GetData, was created to provide a lighter weight interface (in contrast to very expressive query languages that require lots of computational resources to process, such as the query languages SeRQL developed for Sesame RDF platform and SPARQL for OWL). Guha et al. argue that "The lighter weight query language could be used for querying on the open, uncontrolled Web in contexts where the site might not have much control over who is issuing the queries, whereas a more complete query language interface is targeted at the comparatively better behaved and more predictable area behind the firewall" [72]. GetData provides a simple interface to data presented as directed labeled graphs, which does not assume prior knowledge of the ontological structure of the data.

Semantic Web applications. In 2004 the annual Semantic Web Challenge was launched, whose first winner was CS Aktive Space [78]. 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 [61] and Museum Finland [41] 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 not semantic heterogeneity and "openness" here: these tools simply extract information to populate a single, pre-defined ontology. For instance, CS Aktive Space is close with respect to the semantic data they use, it can not take into account RDF data available from a particular web site in response to an user request. CS Aktive Space can only use the data that the system developers have scraped from the various relevant sites and re-described in terms of the AKT reference ontology[7]. A partial exception to this rule is Flink, which makes use of some existing semantic data, by aggregation of online FOAF files.

As argued in [66], obviously, the major challenge faced by these early applications and tools was the lack of online semantic information. Therefore, in order to demonstrate their methods, they had to produce their own semantic metadata, before being able of utilizing them. As a result, either the focus is on a single, well defined domain (Flink, CS Aktive Space, MuseumFinland), or the tool is domain independent, but only one ontology can be active at the time (Magpie[8], AquaLog). Taking a step back it is easy to see that all these applications follow the paradigm of smart database / knowledge base centered applications rather than truly explore the dynamic heterogeneous nature of the SW, embracing the SW paradigm. Although they set out to integrate distributed and heterogeneous resources, these resources end up centralized in a semantic repository aligned under a single ontology [66].

3.2. ONTOLOGY AND SCHEMA BASED MATCHERS

The importance of mapping for the SW has been widely recognized [79] and a wide range of techniques and tools have already been developed. An analysis of the state of the art in mapping systems is presented in [79]. Basically, current approaches to ontology mapping combine a range of:

- Lexical or terminological: non semantic techniques that exploit string similarity between meaningful labels.
- Structural: relies on the structure of the mapped ontologies
- Instance-based: mapping concepts on the basic of shared instances
- Background knowledge: use of external sources as an oracle (WordNet, high level or domain dependant ontologies) to find mappings typically missing by previous approaches.

The majority of approaches use a combination of lexical and structural methods, where lexical overlap is used to produce an initial mapping that is subsequently improved by the structure of the source and the target. For instance, Falcon-AO [68] outperformed all other ontology matchers in the 2005 Ontology Alignment Context OAC [29].

[7] http://www.aktors.org/akt/

[8] Magpie (Dzbor et al., 2003) tool, in absence of available semantic markup, this tool automatically generates a semantic layer, by mapping items on the current web page to an ontology, by means of Named Entity Recognition technology.

Falcon-OA regards ontologies as graph-like structures to produce mappings by using both linguistic and structural similarity. In many cases string and structural similarities can imply meaningful mappings but also as observed to some extend in the OAC 05, traditional methods fail when there is little overlap between the labels of the ontology entities, or when the ontologies have weak or dissimilar structures, i.e. “academics” and “researchers” have the same meaning in a scientific scenario but they are syntactically very different.

As stated in [76], the constituents of mapping can only be given meaning in the context of their own ontology, therefore, a semantic mapping between two ontologies could only be interpreted in a larger domain to the ones of these ontologies (chicken is related to food in a domain about food but not in a domain about living beings). Therefore, in order to achieve semantic mapping predefined sources like WordNet and reference domain ontologies have been used as an oracle for background knowledge as in SMatch[32]. SMatch for example, translates ontology labels into logical formula between their constituents and maps those in corresponding WN sense, where a SAT solver is then used to derive semantic mappings. Other few approaches [3, 81] have considered the use of external background knowledge and rely on a reference domain ontology as a way to obtain semantic mappings between syntactically dissimilar ontologies. In [3] terms from two vocabularies are first mapped to so called anchor terms in the DICE ontology used as background knowledge, and then their mapping is deduced based on the semantic relation of the anchor terms.

However, as stated on [76] obtaining the right background knowledge is problematic, because they rely on axiomatized domain ontologies [3] that unfortunately do not exist in all domains or they are unlikely to cover all intended mappings between the input ontologies, or it is not possible to select the relevant ontology in advance but in real time (like in a SW scenario). In [83] online textual resources (i.e. google) are used as a source of background knowledge, so there is not need for a manual and domain dependent ontology selection task prior to mapping, but the drawback is that the required knowledge extraction techniques lead to considerable noise and human validation is needed.

A more suitable and promising approach in the SW scenario to overcome the limitation of syntactic approaches and obtain semantic relation even between dissimilar ontologies is presented in [76], where ontology mapping can actually exploit the heterogeneity of the Semantic Web while trying to cope with it, and use it as background knowledge sources. Using the semantic data as sources of background knowledge is likely to be less noisy that the one derived from textual sources and therefore lead to the discovery of better mappings. This approach requires two candidate words or mappings as input, then *Swoogle* is used to find ontologies containing concepts with the same names as the candidate concepts and to derive mapping from their relationship between them or the terms in their neighborhood in the selected ontologies or distributed over several ontologies (cross-ontology mapping) when there is not a single ontology that relates them together (recursive task). However, the limitation of this approach is that the terms have to be found as classes (or properties) in the pool of ontologies in the first place by using string exact matching techniques, second is the so called *knowledge sparseness phenomenon:* some domains are well covered by existing ontologies (e.g. academic research and medicine), while others are not covered at all; and third it does not consider the meaning of the mappings in the different ontologies but because we are dealing with ontologies this feature can be easily added by introducing semantic similarity measures (like the ones presented in Section 3.3).

However, the predominant view of mapping is that it will be performed at ***“design time”***, e.g. when deciding on mapping rules between a set of ontologies [13]. This was a plausible assumption because, until recently, only a limited amount of semantic data was available; therefore, there was little need for run time integration. Indeed, one of the main characteristics of SW based applications built so far is that they tackle the data heterogeneity problem in the context of a given domain or application by integrating a few, a-priori determined sources [41, 61]. Hence, we are now slowly reaching a key point in the history of this very young discipline, where we can start moving away from the early applications characterized by limited degree of scale and heterogeneity and start developing the kind of applications, which will define the SW of the future. Obviously, this new scenario brings novel challenges for ontology mapping techniques.

Challenges and requirements for mapping in the Context of Semantic Web tools:

The problem of ontology schema mapping has been investigated by many research groups which have proposed a large variety of approaches [79, 72]. While all this research has produced increasingly complex algorithms, the setting in which the mapping problem was tackled was almost always the same: given two ontologies, find all the possible mappings between their entities attaching a confidence level to the mappings that are returned. One of the challenges in the field of ontology mapping now is not so much perfecting these algorithms, but rather trying to adapt them to novel scenarios, which require SW applications to automatically select and integrate semantic data available online.

Obviously, mapping techniques are crucial in achieving this goal. However, the setting in which the mapping would take place is quite different from the "traditional" ontology mapping scenario, rather than being performed during the development of the application it now needs to be performed at ***"run time"***. These new scenarios impose a number of requirements:

1) **More ontologies** – when integrating data from online ontologies it is often necessary to map between **several** online ontologies. This is very unlike the traditional scenario where only two ontologies were mapped at a time.
2) **Increased heterogeneity –** traditional mapping techniques often assume that the ontologies to be matched will be similar in structure, describe more or less the same topic domain. For example, S-Match [32] is targeted towards matching classification hierarchies. Or, due to its structure based techniques, Anchor-PROMPT [69] works best if the matched ontologies have structures of similar complexity. Such similarity assumptions fail on the SW: we cannot predict whether relevant information will be provided by a simple FOAF file or by WordNet, or top level ontologies, or combined from these different sources. Mapping techniques should function without any pre-formulated assumptions about the ontological structure.
3) **Time Performance is important** - As already pointed out in [54], the majority of mapping approaches focus on the effectiveness (i.e., quality) of the mapping rather than on its efficiency (i.e., speed). This is a major challenge that needs to be solved in the context of run-time mappings where the speed of the response is a crucial factor. The above mentioned paper also shows that some minor modifications of the mapping strategy can highly improve response time and have only a marginal negative effect on the quality of the mappings. Unfortunately the work presented in [54] is rather unique in the context of mapping research, although we think that such research is crucial for making mapping techniques usable during run-time.
4) **Consider relation and instance mappings –** much of the work in ontology mapping has focused on matching the concepts in two schemas, while other ontology entities, such as relations and instances, have largely been ignored so far (although relations and instances are taken into account as evidence to support the matching process in some approaches). However, SW tools are often used to find out information about specific entities (traditionally modeled as ontology instances), as well as the relations between entities. Therefore, we think that mapping techniques should be developed to efficiently map also between these kinds of entities.
5) **Cross-ontology mapping filtering** - several approaches adopt the model of first generating all possible mappings and then filtering the relevant ones. However, in these approaches mappings are typically created between two ontologies describing the same domain. When performing mappings on the SW, we are also likely to discover several mappings but this time the mapping candidates might be drawn from different ontologies. Therefore we need to be able to reason about ontologies which may only have very few concepts in common. As discussed later in this paper, this requires mechanisms to assess whether or not such 'sparse concepts' are related.
6) **Produce Semantic output –** with the exception of S-Match, most mapping algorithms simply determine a similarity coefficient between the concepts that are mapped. Such coefficients are not very useful if the mappings have to be automatically used by a tool. In the scenario of SW tools, to support automatic processing of the mapping results, it would be more useful to return the semantic relations between the mapped entities (equivalent, more generic/specific) rather than just a number. Because schemas do not provide explicit semantics for their data, light weight ontologies are preferred to catalogs or classifications. Mappings should be performed based on the meaning. A critical problem is to distinguish between different vocabularies used by different ontologies to describe similar information across domains and similarly spelled classes that may not have precisely matched meanings.

3.2. ONTOLOGY SELECTION AND RANKING

Currently, there is now a reasonable amount of online semantic data, to such an extent that the need has arisen for a semantic search engine, such as Swoogle [88] and different RDF ontology storage technologies suitable for processing SW information [36], e.g. 3store and Sesame servers.

For the Semantic Web applications to take advantage of the vast amount of the heterogeneous semantic data available by the growth of the SW and get free of the burden of engineering their own semantic data, they need to concentrate on meaningfully finding the relevant ontologies and semantic markup. Robust mechanisms for selecting ontologies are crucial to support knowledge reuse in this large scale open environment. Some of the requirements imposed by the context of automatic knowledge reuse, like coverage even if returning ontology combinations is needed, performance, dealing with instances and relations as well as classes, and modularization, have been highlighted on [77]

OntoSelect [16] ontology selection algorithms, among others, relies on the "connectedness" criteria since they look at how well an ontology is connected to other ontologies, through the ontologies imported as semantic link between ontologies, in order to determine is popularity. It also uses *structure* metrics to measure the number of properties relative to the number of classes in the ontology. The rationale behind is that "more advanced ontologies have a large number of properties". It also applies some coverage criteria.

AktiveRank [2], currently the most advanced algorithm following the study in [77], uses a set of ontology structure based metrics like compacteness, density or richness of knowledge structure, and the coverage of an ontology given the search terms , as a measure in the ontology ranking approach. To broaden the search space, AktiveRank, uses a fuzzy match between terms and concept names (i.e. "project" is mapped to "projectile") but makes no provision to filter out obviously irrelevant hits. Moreover, AktiveRank needs 2 minutes to evaluate each ontology.

Most of ontology search systems only look for classes or instances that have labels matching a search term either exactly or partially [2] without looking for synonymy information, therefore, there is a need to maximize recall to find similar ontological terms with dissimilar labels. Moreover, the approaches for ontology selection fail to consider the meaning of the concepts given by their positions in the ontology hierarchy (e.g. Queen/Royal, Queen/Bee). Also, most approaches (except OntoSelect) focus only in concepts and ignore the existence of relations between concepts, which can potentially provide valuable information to narrow down the right ontologies.

Furthermore, Swoogle by far the most advance ontology search available today, it still rather limited for exploiting online ontologies dynamically at run time. Swoogle claims to adopt a Web view on the Semantic Web by using a modified version of the PageRank popularity algorithm, and by large ignores the semantic particularities of the data that it indexes, and therefore it only measures the quality of its ontologies in terms of their popularity and its querying facilities are limited to keyword based search. Swoogle can not find ontologies which relevant concepts are syntactically dissimilar and its fuzzy search functionality is rather brittle and generates invalid hits (e.g. "update" when searching for "date"). Moreover, Swoogle does not provide the needed capabilities to look beyond a syntactic analysis to perform a semantic analysis to check the soundness of the previously identified mappings in real time. Swoogle does not provide reasoning capabilities and inheritance that current ontology repositories (like sesame or 3store) provide and that are required to perform a semantic analysis (e.g. finding the relations between two instances), or like BRAHMS [43] a RDF storage system adequate for efficiently discover of semantic associations in ontologies by searching for relationships, variable in length and with unspecified directionality, paths in reasonable time.

3.3. SEMANTIC MEASURES AND WORD SENSE DISAMBIGUATION

Because similarly spelled words (labels) may have not precisely matched meanings. Relationships between word senses, not words, are needed. Word sense disambiguation (WSD), is measured through the notion of *similarity*. Many reasonable similarity measures and strategies exist in the literature for WSD (see [42] for a state of the art).

Note that **similarity** is a more specialized notion than **association or relatedness**. Similar entities are semantically related by virtue of their similarity (bank-trust company). Dissimilar entities may also be semantically related by lexical relationships such as meronym (*car-wheel)* and antonymy (*hot-cold)*, or just by any kind of functional relationship or frequent association (*pencil-paper, penguin-Antarctica*) [14]. Taking the example in [74] doctors are minimally similar to medicines and hospitals, since these things are all instances of "something having concrete existence, living or nonliving" (although they may be highly associated), but they are much more similar to lawyers, since both are kinds of professional people, and even more similar to nurses, since both are professional people within the health professions.

Although there have been numerous proposals for semantic distance, similarity and relatedness measures, the majority of these have been based on the following underlying approaches (in [12] a library of similarity measures to use in ontologies are compared with a "gold standard" established by surveying 84 human subjects in two ontologies):

- Ontology-based: the common ancestor based specification (depth) seems to better reflect the common sense understanding the closeness of two objects in a taxonomy.

- Information theory based: measures similarity in an ontology (i.e. WN) in terms of information theoretic entropy measures, e.g. the information of a class is the probability of encountering a class use (or its instances or its descendants).

- Vector space and string based: The typically used similarity measures for vectors are the cosine measure, the extended Jaccard measure and the overlap measure. These methods assume that all vectors have been ordered the same way and scaled to the same length (i.e by adding zeros to properties-values not including in the vector). To calculate similarity between strings and tackle minor morphological variations there is a number of string distance metrics proposed by different communities (e.g., *edit or Levenshtein distance metrics*) [14].

The most intuitive similarity measure between concepts in an ontology is their distance within the ontology following the number of IS_A relations between them. So, that the shorter the path between two terms [74] the more similar they are. However, a widely acknowledged problem is that the approach typically "relies on the notion that links in the taxonomy represent uniform distances", but typically this is not true and there is a wide variability in the "distance" covered by a single taxonomic link [14]. Resnik [74] established that one criterion of similarity between two concepts is the extent to which they share information in common, which, in an IS-A taxonomy, can be determined by inspecting the relative position of the most-specific concept (lso) that subsumes them both. The number of links (depth) is still important to distinguish between any two pairs of concepts having the same lso. One of the variations of this edge-counting method is the conceptual similarity introduced by Wu & Palmer [85]:

$$Similarity\ (C1, C2) = (2 \times N)\ /\ (N1 + N2 + 2 \times N)$$

Where Ni = length of the path from Ci to C; N= length of path from C to root; C = lso (C1, C2)

Researchers in the NLP domain have proposed measuring both similarity and relatedness between two concepts by commonly using WordNet as the reference ontology. The current version of WordNet provides *a priori* lexical and domain knowledge. As Ide and Veronis state [42], WordNet is the most used lexical resource at present for disambiguation in English. Most of the research methods in the literature are limited to WordNet [14]. Nouns, verbs, adjectives, and adverbs are each organized into networks of synonyms sets (*synsets).* Each synset has a gloss to define it. There are nine types of semantic relations defined on the noun subnetwork: hyponymy (IS-A) relation, and its inverse hypernymy; six meronymic (PART-OF) relations – COMPONENT-OF, MEMBER-OF, SUBSTANCE-OF and their inverses; and the COMPLEMENT-OF relation. the fine-grainedness of WordNet sense distinctions, e.g. in this case city#1 and city#2, is a frequently cited problem. Other high level ontologies can be also useful, like the SUMO upper level ontology that has been extended with the mappings to the WordNet lexicon [74].

Information theoretic measures while outperforming Resnik's similarity algorithm slightly they do still require a probabilistic model of the application domain. This limitation makes it problematic for smaller ontologies [12]

Most approaches assume that words that appear together in a sentence can be disambiguated by assigning to them the senses that are most closely related to their neighboring words [30]. In *Hierarchy distance based matchers* [31] the relatedness between words is measured by the distance between two concepts/senses in a given input hierarchy. In particular, similarity between words is measured by looking at the shortest path between two given concepts/senses in the WordNet "IS-A" taxonomy of concepts. For instance, the algorithm described in [56] to make explicit the semantics hidden in schema models: Let L be a generic label for a concept and L1 either an ancestor label or a descendant label of L and let s* and s1* be respectively the sets of WordNet senses of a word in L and a word in L1. If one of the senses belonging to s* is either a synonym, hypernym, holonym, hyponym or a meronym of one of the senses belonging to s1*, these two senses are retained and all the other senses are discarded. As an example, imagine *Apple* (which can denote either a fruit or a tree) and *Food* as its ancestor; since there is a hyponymy relation between *apple#1* (denoting a fruit) and *food#1*, we retain *apple#1* and discard *apple#2* (denoting a tree).

Pendersen and his colleagues [30] have made available a Perl implementation of six WordNet measures evaluated in [14] plus their own sense disambiguation algorithm based on glosses [30] to assign a meaning to every content word in a text. Basically, these measures look for a path connecting a synset associated with each word, e.g. in Hirst and St-Onge measure the intuition behind is "the longer the path and the more changes of direction (upward for hypernym and meronym; downward for hyponymy and holonymy and horizontal for antonymy) the lower the weight". In [30] *extended semantic gloss matchers* measure semantic relatedness between concepts (and its ancestors/descendants according to the *is-a* WordNet hierarchy) that is based on the number of shared words in their definitions (glosses).

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

Moreover, WordNet is (1) very limited on the coverage of compound terms frequently used in ontologies (e.g. "municipal-unit" instead of "municipality"), so if the keyword is not found then the semantic similarity or relatedness cannot be computed, and (2) rather static, e.g. *developer* as the intended sense of *software developer*, or *apple* as a *computer* do not appear in WordNet 3.0.

A novel approach introduced by [33] consists in using not only WordNet but also the whole SW as background knowledge, so that different ontologies represent different views of the world and therefore different senses. The authors propose a multi ontology based method to disambiguate the senses of keywords used in a search engine (e.g., in (astronomy, star, planet), "star" is used in its sense of celestian body*)* while traditionally methods would have relied on WordNet alone to collect possible sense for the keywords, now they can exploit all online ontologies to gather a much larger set of senses.

Furthermore, tree or graph-based similarities are not addressed here because they are computationally very expensive and therefore not applicable for run time scenarios.

To summarize, in multiple information systems environments, with their own models and ontologies, the general approach to data integration has been to map the local terms of distinct ontologies onto a single shared ontology. Then, semantic similarity is determined as a function of the path distance between terms in the hierarchy of the underlying single ontology. Further work is needed to extrapolate these techniques for cross-ontology comparisons (i.e. to calculate similarity in QA scenarios where the candidate concepts belong from different ontologies), which require to deal with one of the major problems of the SW: the mapping between ontologies of different origin without constructing a priori a shared ontology. An exception is the work presented on [75] which propose a weak form of integration establishing links among ontologies while keeping each ontology autonomous. The authors emphasize the importance of properties and attributes as distinguished features of entity classes. For example, a "hospital" and "apartment building" have a common superclass "building", however this information falls short when trying to differentiate between both classes, since the is-a relation does not indicate the important difference in terms of entities' functionality or features. They classify features into functions, parts and attributes. However, the drawback is the mismatches associated with the classification of features in different ontologies

3.4. CO-REFERENCE OF INSTANCES.

Identifying whether two instances from semantically equivalent concepts are the same is not an easy task. Instances may not have the same name, and information about the same instance can have different purposes, e.g. the description of a car for sale or for an environmental study. We can use the OWL mechanism which identifies the attributes that provide sufficient evidence that two instances are the same. However, further mechanisms need to be adopted, e.g., use of joint probability approaches similar to GLUE[27] over the instance full name (from the taxonomy root) and its textual content (word frequency over attributes and values).

For the instance fusion task, the SW community has adopted the distance measurement approaches in the database community in which the distance between records was calculates as a weighted average of distances over attributes. The main distinctive feature of the SW as opposed to the database domain is the issue of incompleteness [34].

In [17] Wikipedia is used to find similar or alternative names, synonyms, misspellings, abbreviations and acronyms in the case of queries about name entities and solve ambiguities between multiple name entities that can be denoted by the same proper name (e.g. the disambiguation page for the name *John Williams* lists 22 associated entities). Also, Wikipedia captures references that are distinct to senses, like *Morning_star* as a potential referent for Venus. Also, every article in Wikipedia is required to have at least one category and one or more topics. However, in practice there may be contexts where entities, like *John Williams*, refers to an non popular entity and therefore is not covered in Wikipedia.

To merge profile information, Flink [61] makes use of the rule language for carrying out Identity reasoning (smushing) to determine if different instances (in this case individuals) across multiple information sources refer to the same individual. The methods for smushing are based on name matching or the similarity of names as strings (differences in the last names are disallowed) and object identification based on the inverse functional properties (IFPs) of FOAF. For example, if the instances A and B have the same value for the *foaf:mbox* property, the match is recorded using the A *owl:sameAs* B property.

References

1. Adams T., Dullea J., Clark P., Sripada, S. and Barrett T. (2000) Semantic Integration of Heterogeneous Information Sources Using a Knowledge-Based System. *In Proc. of the 5th International Conference on Computer Science and Informatics*.
2. Alani, H. and Brewster, C. (2005) Ontology Ranking based on the Analysis of Concept Structures. *In Proc, of the 3th International Conference on Knowledge Capture.*
3. Aleksovski, Z., Klein, M., ten Katen, W., van Harmelen, F. (2006) Matching Unstructured Vocabularies using a Background Ontology. *In Proc. of the EKAW.*
4. Aleman-Meza, B., Halaschek, C., Sheth, A., Arpinar, I. B., Sannapareddy, G. (2004) "SWETO: Large-Scale Semantic Web Test-bed. *In Proc. of the16th Intl. Conf. on Software Engineering & Knowledge Engineering: Intl. Workshop on Ontology in Action*
5. 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.*
6. Androutsopoulos, I., Ritchie, G.D., Thanisch P.: Natural Language Interfaces to Databases - An Introduction. *Natural Language Engineering*, 1(1) (1995) 29-81.
7. AskJeeves: http://www.ask.co.uk.
8. Attardi, G., Cisternino, A., Formica, F., Simi, M., Tommasi, A., Zavattari, C. (2001) PIQASso: PIsa Question Answering System. *In Proc of the text Retrieval Conferemce (Trec-10),* 599-607, NIST, Gaithersburg(MD), Nov. 13-16.
9. Basili, R., Hansen, D., H., Paggio, P., Pazienza M., T., Zanzotto F., M. (2004) Ontological resources and question data in answering. *Workshop on Pragmatics of Question Answering, held jointly with NAACL.* Boston, Massachusetts.
10. Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web. *Scientific American,* 284(5), 2001.
11. Bernstein, A., Kaufmann, E. (2006) GINO - A Guided Input Natural Language Ontology Editor. *In Proc of the International Semantic Web Conference*: 144-157
12. Bernstein, A., Kaufmann, E., Kiefer, C., and Burki, C. (2005) SimPack: A Generic Java Library for Similarity Measures in Ontologies. Technical report, University of Zurich, Department of Informatics.
13. Bouquet P., Serafini L. and Zanobini S. (2003) Semantic coordination: a new approach and an application. *International Semantic Web Conference* 130-145.
14. Budanitsky, A. and Hirst, G. Evaluating WordNet-based measures of semantic distance. *Computational Linguistics* (2006).
15. Buitelaar, P., Declerck, T., Calzolari, N., Lenci, A. (2003) Language Resources and the Semantic Web. *In Proc. of the ELSNET/ENABLER Workshop.*
16. Buitelaar, P., Eigner, T, Declerck, T. (2004) OntoSelect: A Dynamic Ontology Library with Support for Ontology Selection. *In Proc. of the Demo Session at the International Semantic Web Conference*.
17. Bunescu, R., Pasca, M. (2006) Using Encyclopedic Knowledge for Named Entity Disambiguation. *In Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL-06).*
18. Burger, J., Cardie, C., et al.: Issues, Tasks and Program Structures to Roadmap Research in Question & Answering (Q&A) *NIST,* 2002
19. 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.
20. 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*.
21. Clark, P., Thompson, J., Porter., B.: A Knowledge-Based Approach to Question-Answering. In the *AAAI Fall Symposium on Question-Answering Systems*, CA:AAAI. (1999) 43-51.
22. Cohen, W., W., Ravikumar, P., Fienberg, S., E. (2003) A Comparison of String Distance Metrics for Name-Matching Tasks. *In IIWeb Workshop.*
23. Copestake, A., Jones, K., S.: Natural language interfaces to databases. *Knowledge Engineering Review*. 5(4) (1990) 225-249.
24. 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 (ACL'02).* Philadelphia.

25. De Boni, M.: TREC 9 QA track overview.
26. 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*, Pengang, Malaysia,.
27. Doan A., Madhavan J., Domingos P., and Halevy A. (2002) Learning to Map between Ontologies on the Semantic Web. *In Proc. of the World-Wide Web Conference*.
28. EasyAsk: http://www.easyask.com.
29. Euzenat, J., Stuckenschmidt, H., Yatskevich, M. (2005) Introduction to the Ontology Alignment Evaluation 2005. *In Proc. of the Integrating Ontologies Workshop*
30. Extended Gloss Overlaps as a Measure of Semantic Relatedness (2003) Banerjee S., and Pedersen T. *In Proc. of the International Joint Conference on Artificial Intelligence*.
31. Giunchiglia F. and Yatskevich M. (2004) Element Level Semantic Matching. *Meaning Coordination and Negotiation Workshop, ISWC*.
32. Giunchiglia F., Shvaiko P and Yatskevich M. (2004) S-Match: an algorithm and an implementation of semantic matching. *In Proc. of the 1t European Semantic Web Symposium*.
33. Gracia, J., Trillo, R., Espinoza, M., Mena, E. (2006). Querying the web: a multiontology disambiguation method. *In Proc. of the 6th international conference on Web engineering.*
34. Guha, R. (2004) Object Co-Identification on the Semantic Web, *In Proc of the 13th WWW.*
35. Guha, R., McCool, R., and Miller, E. (2003). Semantic search. *In Proc of the 12th international conference on WWW.*
36. Guo, Y., Pan, Z., Heflin, J. (2004) An Evaluation of Knowledge Base Systems for Large OWL Datasets. *International Semantic Web Conference* 274-288
37. 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)*, Gaithersburg.
38. Hirschman, L., Gaizauskas, R.: Natural Language question answering: the view from here. ***Natural Language** Engineering, Special Issue on **Question Answering***, 7(4) (2001) 275-300.
39. 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, Gaithersburg, MD.
40. Hunter, A.: Natural language database interfaces. *Knowledge Management*, (2000).
41. Hyvonen, E., Makela, E., Salminen, M., Valo, A., Viljanen, K., Saarela, S., Junnila, M., Kettula, S.. MuseumFinland – Finnish Museums on the SemanticWeb. *Journal of Web Semantics*, 3(2), 2005
42. Ide N. and Veronis J. Word Sense Disambiguation: The State of the Art. *Computational Linguistics*, 24(1):1-40. (1998).
43. Janik, M., Kochut, K. (2005) BRAHMS: A WorkBench RDF Store and High Performance Memory System for Semantic Association Discovery. *International Semantic Web Conference*.
44. 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).*
45. Katz, B., Lin, J.: REXTOR (2000) A System for Generating Relations from Natural Language. *In Proc of the ACL-2000. Workshop of Natural Language Processing and Information Retrieval (NLP&IR).*
46. Kaufmann, E. (2006) Talking to the Semantic Web - Query Interfaces to Ontologies for the Casual User. *In Proc of the International Semantic Web Conference*
47. Kaufmann, E., Bernstein, A., Zumstein, R. (2006) Querix: A Natural Language Interface to Query Ontologies Based on Clarification Dialogs", *In Proc of the* 5th International Semantic Web Conference (ISWC 2006).
48. Levy A., Y., Srivastava D. and Kirk T. Data Model and Query Evaluation in Global Information Systems. *Journal of Intelligence Information Systems.* 5(2): 121-143 (1995).
49. Linckels S., M.C. (2005) A Simple Solution for an Intelligent Librarian System*. In Proc of the IADIS International Conference of Applied Computing.*
50. Litkowski, K. C. Syntactic Clues and Lexical Resources in Question-Answering, in Voorhees, E. M. and Harman, D. K. (Eds) *Information Technology: The Ninth Text REtrieval Conferenence (TREC-9)*, NIST Special Publication 500-249. Gaithersburg, MD: National Institute of Standards and Technology, (2001) 157-66.
51. Lopez, V. and Motta, E. (2004) Ontology Driven question answering in AquaLog, 9th International Conference on Applications of Natural Language to Information Systems (NLDB 2004), Manchester, UK
52. Lopez, V., Motta, E. and Pasin, M. (2005) AquaLog: An Ontology portable Question Answering Interface for the Semantic Web, *In Proc. of the European Semantic Web Conference*, Heraklion, Crete
53. Lopez, V., Motta, E., Uren, V. and Pasin, M. (2007) AquaLog: An ontology-driven Question Answering System as an interface to the Semantic Web, *To appear in Journal of Web Semantics*, 2007.
54. M. Ehrig and S. Staab. QOM: Quick ontology mapping. In *Proceedings of ISWC*, 2004, pages 683–697
55. Madhavan, J., Bernstein, P.A. and Rahm, E. Generic schema matching with cupid. *The Very Large Databases Journal*: 49-58 (2001)
56. Magnini B., Serafín L., and Speranza M. (2003) Making Explicit the Semantics Hidden in Schema Models. *In Proc. of the Workshop on Human Language Technology for the Semantic Web and Web Services*, held at *ISWC-2003*.
57. Martin, P., Appelt, D., E., Grosz, B., J., Pereira, F., C., N.: TEAM: An Experimental Transportable Natural-Language Interface. *IEEE Database Eng*. Bull. 8(3) (1985) 10-22.

58. Mc Guinness, D., van Harmelen, F.: OWL Web Ontology Language Overview. W3C Recommendation 10 (2004) http://www.w3.org/TR/owl-features/.
59. Mc Guinness, D.: Question Answering on the Semantic Web. *IEEE Intelligent Systems*, 19(1), 2004.
60. Mena E., Kashyap V., Sheth A. and Illarramendi A. 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 (2000)
61. Mika, P. Flink: SemanticWeb Technology for the Extraction and Analysis of Social Networks. *Journal of Web Semantics*, 3(2), 2005
62. Mitra P., Wiederhold G., Jannink J. (1999) Semi-automatic Integration of Knowledge Sources. *In Proc. of the 2nd International Conference on Information Fusion.*.
63. Moldovan, D., Harabagiu, S., et al. (2002) LCC Tools for Question Answering. *In Proc of the TREC*
64. Moldovan, D., Harabagiu, S., Pasca, M., Mihalcea, R., Goodrum, R., Girju, R., Rus, V. (1999) LASSO: A Tool for Surfing the Answer Net. *In Proceedings of the Text Retrieval Conference (TREC-8)*, Nov.
65. Moldovan, D., Pasca, M., Harabagiu, S., Surdeanu, M.: Performance issues and error analysis in an open-domain question answering system. ACM Trans. Inf. Syst. 21(2): 133-154 (2003)
66. 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)*, Genoa, Italy.
67. Motta, E., Sabou, M. (2006) Next Generation Semantic Web Applications. *In Proc. of the 1st Asian Semantic Web Conference*. Beijing.
68. N. Jian, W. Hu, G. Cheng and Y. Qu (2005) Falcon-AO: Aligning Ontologies with Falcon. Proceedings of the K-CAP. *Workshop on Integrating Ontologies*
69. N. Noy and M. Musen. (2001) Anchor-PROMPT: using non-local context for semantic matching. *In Proceedings of the workshop on Ontologies and Information Sharing at the International Joint Conference on Artificial Intelligence (IJCAI).*
70. Pazienza, M. T., Stellato, A., Henriksen, L., Paggio, P., and Zanzotto, F. M. (2005) Ontology Mapping to support ontology-based question answering. ***In Proc of the second MEANING workshop***
71. 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*, Miami, FL, USA, Jan. 12-15, 149-157.
72. Rahm E. and Bernstein P. A. A survey of approaches to automatic schema matching. *The International Journal on Very Large Data Bases* 10(4): 334-350, 2001.
73. RDF: http://www.w3.org/RDF/.
74. Resnik P. (1995) Disambiguating noun grouping with respect to WordNet senses. *In Proc. of the 3rd Workshop on very Large Corpora*. MIT.
75. Rodríguez, M.., and Egenhofer, M. Determining semantic similarity among entity classes from different ontologies. *IEEE Trans. on Knowledge and Data Eng.*, 15(2):442--456, 2003.
76. Sabou, M., d'Aquin, M., and Motta, E. (2006). *In Proc. of the International Workshop on Ontology Matching.*
77. Sabou, M., Lopez, V., Motta, E. (2006) Ontology Selection on the Real Semantic Web: How to Cover the Queens Birthday Dinner?. *In Proc. of the EKAW*.
78. 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 WWW Conference*.
79. Shvaiko, P., Euzenat, J.: A Survey of Schema-Based Matching Approaches. J*ournal. of Data Semantics IV*. 146-171 (2005).
80. Srihari, K., Li, W., Li, X. (2004) Information Extraction Supported Question- Answering, In T. Strzalkowski & S. Harabagiu (Eds.), *In Advances in Open- Domain Question Answering*. Kluwer Academic Publishers.
81. Stuckenschmidt, H., van Harmelen, F., Serafini, L., Bouquet, P., and Giunchiglia, F. (2004). Using C-OWL for the Alignment and Merging of Medical Ontologies. *In Proc. of the First Int. WS on Formal Biomedical K.R. (KRMed)*
82. Uren, V., Lei, Y., Lopez, V., Liu, H.: A review of usability issues in Semantic Search Tools. *To appear in ???*, 2007.
83. Van Hage, W., Katrenko, S., and Schreiber, G. (2005). A method to Combine Linguistic Ontology-Mapping Techniques. *In Proc. of ISWC.*
84. Waldinger, R., Appelt, D. E., et al.: Deductive Question Answering from Multiple Resources. In Maybury, M. (Eds.), *New Directions in Question Answering*, AAAI Press (2003).
85. Wu Z., and Palmer, M. (2004) Verb Semantics and Lexical Selection. *In Proc of the 32nd Annual Meeting of the Associations for Computational Linguistics.*
86. Wu, M., Duan, M. Y., Shaikh, S., Small, S., and Strzalkowski, T. (2005). University of Albany's ILQUA in TREC 2005. *In Proc of the Text Retrieval Conference (TREC 2005)*. NIST.
87. 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*) 500-255.
88. X L. Ding et al. (2004) Swoogle: A semantic web search and metadata engine. *In Proc. 13th ACM Conf. on Information And Knowledge Management*
89. Zheng, Z. (2002) The AnswerBus Question Answering System. *In Proeedings.of the Human Language Technology Conference (HLT 2002)*. San Diego, CA. March 24-27.