# COOPERATIVE QUESTION ANSWERING FOR THE SEMANTIC WEB

Dora Melo[1], Irene Pimenta Rodrigues[2] and Vitor Beires Nogueira[2]
[1]*Instituto Politécnico de Coimbra and CENTRIA, Coimbra, Portugal*
[2]*Universidade de Évora and CENTRIA, Évora, Portugal*

Keywords: Natural language, Ontology, Question answering, Semantic web.

Abstract: In this paper we propose a Cooperative Question Answering System that takes as input queries expressed in natural language and is able to return a cooperative answer obtained from resources in the Semantic Web, more specifically DBpedia databases represented in OWL/RDF. Moreover, when the DBpedia provides no answer, we use the WordNet in order to build similar questions. Our system resorts to ontologies not only for reasoning but also to find answers and is independent of prior knowledge of the semantic resources by the user. The natural language question is translated into its semantic representation and then answered by consulting the semantics sources of information. If there are multiple answers to the question posed (or to the similar questions for which DBpedia contains answers), they will be grouped according to their semantic meaning, providing a more cooperative and clean answer to the user.

## 1 INTRODUCTION

Ontologies and the Semantic Web (Horrocks, 2008) became a fundamental methodology to represent the conceptual domains of knowledge and to promote the capabilities of semantic Question Answering systems (Guo and Zhang, 2009). These systems by allowing search in the structured large databases and knowledge bases of the Semantic Web can be considered as an alternative or as a complement to the current Web search.

There is a gap between users and the Semantic Web: it is difficult for end-users to understand the complexity of the logic-based Semantic Web. Therefore it is crucial to allow a common Web user to profit from the expressive power of Semantic Web data-models while hiding its potential complexity. There is a need for user-friendly interfaces that scale up to the Web of Data and support end-users in querying this heterogeneous information source.

Consistent with the role played by ontologies in structuring semantic information on the Web, ontology-based Question Answering systems allows us to exploit the expressive power of ontologies and go beyond the usual "keyword-based queries".

Question Answering systems provide concise answers to natural language question posed by users in their own terminology, (Hirschman and Gaizauskas, 2001). Those answers must also be in natural language in order to improve the system and provide a better friendly-user interface.

In this paper we propose a Cooperative Question Answering System that receives queries expressed in natural language and is able to return a cooperative answer, also in natural language, obtained from resources on the Semantic Web (Ontologies and OWL2 Descriptions). The system starts a dialogue whenever there is some question ambiguity or when it detects that the answer is not what user expected. Our proposal includes deep parsing, use of ontologies and other web resources such as the WordNet (Fellbaum, 1998) and the DBpedia (Auer et al., 2007).

Our goal is to provide a system that is independent of prior knowledge of the semantic resources by the user and is able to answer cooperatively to questions posed in natural language. This system maintains the structure of the dialogue and this structure provides a context for the interpretation of the questions and includes implicit content such as temporal-space knowledge, entities and information useful for the pragmatic interpretation like discourse entities used for anaphora resolution.

This paper is organized as follows. First, in Section 2, we introduce the proposed system, describing the main components of its architecture. In parallel, we present an example as an illustration of system


Melo D., Pimenta Rodrigues I. and Beires Nogueira V..
COOPERATIVE QUESTION ANSWERING FOR THE SEMANTIC WEB.
DOI: 10.5220/0003691202580263
In *Proceedings of the International Conference on Knowledge Management and Information Sharing* (KMIS-2011), pages 258-263
ISBN: 978-989-8425-81-2

functionality. Afterwards, in Section 3 we present related work, highlighting the main differences in the proposed system. Finally, in Section 4, we present the conclusions and the future work.

## 2 PROPOSED SYSTEM

Very briefly, the proposed system receives a natural language question and translates into a semantic representation using Discourse Representation Structures (DRS). Then, after consulting the semantics sources of information, provides a natural language answer. If there are multiple answers to the question posed (or to the similar questions for which DBpedia contains answers), they will be grouped according to their semantic meaning, providing a more cooperative and clean answer to the user. Therefore, we consider that our system provides a user friendly interface.

The language chosen for our system was Prolog with several extensions and libraries. Among the reasons for such choice is the fact that there is a wide range of libraries for querying and processing of ontologies OWL2, WordNet has an export for Prolog and there are extensions that allow us to incorporate the notion of context into the reasoning process. Moreover, Wielemaker (Wielemaker, 2005) provides a study for query translation and optimization more specifically the SeRQL RDF query language, where queries are translated to Prolog goals, optimized by reordering literals. Finally, in (Wielemaker et al., 2007) the authors describe how to develop a Semantic Web application entirely in Prolog.

Our system architecture is presented in Figure 1 and to help its understanding in the following subsections we describe the main components.

### 2.1 Semantic Interpretation

Semantic Interpretation, or Semantic Analysis, is built using First-Order Predicate Logic extended with generalized quantifiers. We take special care with the discourse entities in order to have the appropriate quantifier introduced by the determinant interpretation. The semantic representation is supported by Discourse Representation Theory (Kamp and Reyle, 1993; Blackburn and Bos, 2005; Covington, 1988). The semantic analysis rewrites a syntactic structure of the question into a DRS. For us a DRS is a set of referents, universally quantified variables and a set of conditions (First-Order predicates).

The implementation of this component follows an approach similar to the one for constructing of a Question Answering system over documents

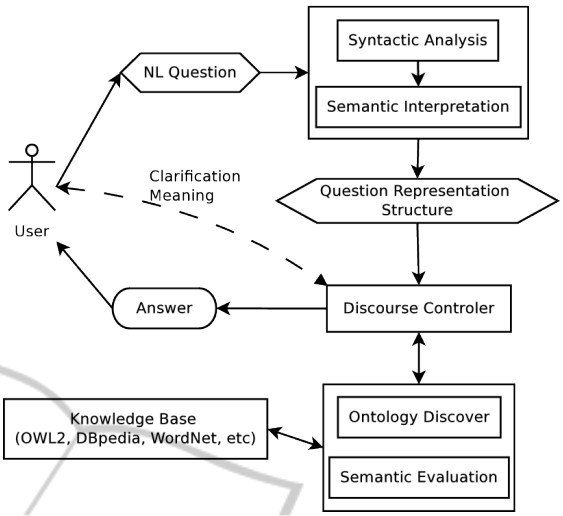


Figure 1: Question Answering System Architecture.

databases proposed in (Quaresma et al., 2006). The system consists of two separate modules: preliminary analysis of the documents (information extraction) and processing questions (information retrieval). The system is looking for processing the corpus and the questions, supported by theories of computational linguistics: syntactic analysis (grammatical restrictions), followed by semantic analysis using the theory of discourse representation and finally the semantic/pragmatic interpretation using ontology and logical inference.

As an illustration, consider the question ”All French romantic writers have died?”. The syntactic analysis generates a tree that is rewritten according to a set of rules and integrated into a DRS. In our system, it is stated by the Prolog fact:

```
drs([all-X],[(writer(X), french(X),
                romantic(X)), died(X)]).
```

where `X` is a universally quantified (`all`) discourse entity that must verify all the question conditions (`writer(X), french(X), romantic(X)), died(X)`).

### 2.2 Ontology Discover

The Ontology Discover is guided by the Discourse Controller to obtain the extension of sentence representation along with the reasoning process. The reasoning context and the question meaning will change whenever the Discourse Controller reaches a dead end.

This system module looks for similarities between labels by means of string-based, taking into account

abbreviations, acronyms, domain and lexical knowledge. To maximize recall, the ontology search looks for classes or instances that have labels matching a search term either exactly or partially and, if an answer is not achieved, each term in the query is extended with its synonyms, hypernyms and hyponyms obtained from WordNet (Witzig and Center, 2003). Afterwards we extract a set of semantic resources which may contain the information requested.

Continuing the example of the previous section, in order to obtain the extension of sentence representation along the reasoning process, the system has to find the answers to the following questions:

- Which Classes or Properties represent the concept 'writer'?

  The system finds the DBpedia[1] class `Writer` [2] with property domain *Work* and domain range *Person*;

- Which Classes or Properties represent the concept 'french'?

  The DBpedia has a class `birthPlace` [3] (an entity of type *ObjectProperty*, with property domain *Person* and domain range *Place*) that represents the place where some person was born. The term 'french' is also interpreted as a "person of France" (obtain from WordNet), so the system also has to find the Classes or Properties of all similar meanings to the initial term that could lead the system to the correct answer;

- Which Classes or Properties represent the concept 'romantic'?

  The system finds the DBpedia resource `Romanticism` [4] (an entity of type *Thing*, an instance of property `movement` [5]);

- Which Classes or Properties represent the concept 'died'?

  The Dpbedia has a class `deathDate` [6] (an entity of type *DatatypeProperty*, with property domain *Person* and domain range *date*) that represents the death date of a person.

The next step is the construction of query(ies) needed to verify the initial question. If the question don't have an answer, a set of similar questions is constructed. Querying the WordNet, the system obtains similar terms to those that compose the initial question. This set of similar questions will enrich the knowledge domain and helps the interpretation of the original question or in the construction of its answer. If this set of new questions leads the system to different answers, we are in the presence of an ambiguity and the user is invoked to clarify it. If the system did not find any correspondence to a word and its derivatives, the user is informed and can clarify the system by reformulating the question or presenting others query(ies).

## 2.3 Semantic Evaluation

Semantic Evaluation is intended to be the pragmatic evaluation step of the system, where the question semantic is transformed into a constraint satisfaction problem. This is achieved by adding conditions that constraint the discourse entities. Moreover, this extra information (regarding the question interpretation) can help the Discourse Controller to formulate a more objective answer.

The Semantic/Pragmatic evaluation must reinterpret the semantic representation of the sentence based on ontology considered in order to obtain the set of facts that represent the information provided by the question.

The process responsible for the Semantic/Pragmatic interpretation receives the DRS of the question and interprets it in a knowledge base with rules derived from the ontology and the information contained in the databases like as DBpedia and WordNet.

Back to our example, to solve the constraint problem the Dialogue Controller generates and poses the following questions to the Question Answering system:

- Who are the French writers?
- Who are the French romantic writers?
- Who are the French romantic writers who died?

To answer this questions the system has to find all entities $[X1, X2]$ such that $X1 : writer(X1)$, $X2 : french(X2)$, where $X1 \cap X2$ represents the entities

[1]The DBpedia Ontology is a shallow, cross-domain ontology, which has been manually created based on the most commonly used infoboxes within Wikipedia. The ontology currently covers over 272 classes which form a subsumption hierarchy and are described by 1,300 different properties. Each class is identified by a URI reference http://dbpedia.org/ontology/Name and each property with http://dbpedia.org/property/property_name URI reference. The DBpedia database provid more than 3.5 million resources, each of them is identified by a URI reference http://dbpedia.org/resource/Name, where `Name` is removed from the URL whose source is the Wikipedia article, which has the form http://en.wikipedia.org/wiki/Name.

[2]http://dbpedia.org/ontology/Writer

[3]http://dbpedia.org/ontology/birthPlace

[4]http://dbpedia.org/page/Romanticism

[5]http://dbpedia.org/property/movement

[6]http://dbpedia.org/ontology/deathDate

that are "French writers". Afterwards the system has to find the entities $[X3]$ such that $X3 : romantic(X3)$ and the evaluation of the expression $X1 \cap X2 \cap X3$ gives the entities that are "French romantic writers".

Finally, the system has to find all entities $[X4]$ belongs to the set $A = X1 \cap X2 \cap X3$ such that $X4 : died(X4)$ and evaluation the resulting set $B = \{X4 \in A : died(X4)\}$ gives the entities that are "French romantic writers who died".

The interpretation of the relations between the sets $A$ and $B$ guides the system to the final answer in the following way:

- If $A \cap B = \emptyset$, then we can conclude that all French romantic writers have died;
- Otherwise, the expression $A \cap B$ gives the set of entities that are French romantic writers still alive.

After querying and searching in knowledge base, the system concludes that the answer to initial question "All French romantic writers have died?" is "Yes, all French romantic writers have died.".

## 2.4 Discourse Controller

The Discourse Controller is a core component that is invoked after the natural language question has been transformed into its semantic representation. Essentially the Discourse Controller tries to make sense of the input query by looking at the structure of the ontology and the information available on the Semantic Web, as well as using string similarity matching and generic lexical resources such as WordNet.

The Dialogue Controller deals with the set of discourse entities and is able to compute the question answer. It has to verify the question presupposition, choose the sources of knowledge to be used and decide when the answer has been achieved or to iterate using new sources of knowledge. The decision of when to relax a question in order to justify the answer and when to clarify a question and how to clarify it also taken by in this module.

Whenever the Discourse Controller isn't sure how to disambiguate between two or more possible terms or relations in order to interpret a query, it starts a dialogue with the user and asks him for disambiguation. The clarification done by the user will be essential for the Discourse Controller, this way obtaining the right answer to the query posed by the user. For instance, the question "Where is the Taj Mahal?", 'Taj Mahal' could be mapped into the name of a Mausoleum, a Casino Hotel or an Indian Restaurant and only the user can clarify about the intended meaning. The more cooperative and interactive the Discourse Controller is, the closer it will be to the correct answer.

Another important aspect of the Discourse Controller is to provide a friendly answer to user. The answer should be as closest to the natural language as possible. For instance, the Question Answering system has to respond "yes" or "no" when the user posed the query "Is Barack Obama the President of the USA?". In this case, the answer will be "yes". However, the answer must be more informative for the user. Some concepts are defined in the temporal context, even if implicitly, and the answer should be more clear and informative. For instance, the term 'President', in the context of the question, is defined as the title of head of state in some republics and has an associated duration for the mandate, a start date (date of election, date on taking office), and an end date of the mandate. So the answer to the question "Is Barack Obama the President of the USA?" should be "Yes, Barack Obama is the actual President of USA", that is more cooperative and informative.

For the cases where the answer to a question of type Yes/No is "No", the Discourse Controller will return a complete answer, clarifying the negation. If we consider the question "All the capitals of Europe have more than 200,000 inhabitants?" that have a "No" as an answer, the system will construct the proper answer that clarify the user and will return "No, 9 capitals of Europe have less than or equal to 200,000 inhabitants".

If there are multiple answers to the question posed by the user (or to the similar questions for which DBpedia contains answers), they will be grouped according to their semantic meaning, providing a more cooperative and clean answer to the user. To do so, the discourse controller has to reason over the question and construct the answer, well constructed questions have always the right words that help in the answer construction. For the question "Where is the Taj Mahal?" consider that the user is not able to clarify the system about the ambiguity of the question: Taj Mahal is a Mausoleum, a restaurant or Casino Hotel; or that the user simply wants that the system returns all possible answers. So when the system has all the answers to all possible interpretations for the question posed by the user, the Discourse Controller will not list the answer in a random way, but will list the answer according to their semantic mean:

```
Mausoleum Taj Mahal is in Agra, India
Casino hotel Taj Mahal is in Atlantic City, NJ, USA
Indian Restaurant Taj Mahal is in New Farm, Brisbane, Australia
Indian Restaurant Taj Mahal is in 7315 3rd Ave. - Brooklyn, NY, USA
```

Our dialoguing system has as main objective the

use of interactive mechanism to obtain more objective and concrete answers. It is not used only to clarify the problems of ambiguity, but also to help finding the path to obtain the correct answer. Making the dialogue system more cooperative makes one able to get closer to the answer desired by the user. In many cases, the user is the only one who can help the system in the deduction and interpretation of information.

## 3 RELATED WORK

The representation of questions with generalized quantifiers as in (Rodrigues et al., 2009) allows the use of various natural language quantifiers like all, at least 3, none, etc. Moreover, the question evaluation also resorts to logic programming with constraints.

A query language for OWL based on Prolog is presented in (Almendros-Jiménez, 2011). The authors propose a way of defining a query language based on a fragment of Description Logic and a way of mapping it into Prolog by means of logic rules.

An illustration of a Question Answering system for Portuguese language that uses the Web as a database, through meta-search on conventional search engines can be seen in (Rabelo and Barros, 2004). This system uses surface text patterns to find answers in the documents returned by search engines. Another example of a Question Answering system where domain knowledge is represented by an ontology can be found in (Guo and Zhang, 2008): it is presented an Interface System for Question Answering Chinese Natural Language, that runs through a natural language parser.

The paper (Nogueira and Abreu, 2007) describes a declarative approach to represent and reason about temporal contextual information. In this proposal each question takes place in a temporal context and that context is used to restrict the answer.

PowerAqua (Lopez and Motta, 2006; Lopez et al., 2007a) is a multi-ontology-based Question Answering system that takes as input queries expressed in natural language and is able to return answers drawn from relevant distributed resources on the Semantic Web. PowerAqua evolved from the earlier AquaLog system (Lopez et al., 2007b). PowerAqua allows the user to choose an ontology and then ask natural language queries related to the domain covered by the ontology. The system architecture and the reasoning methods are completely domain-independent, relying on the semantics of the ontology, and the use of generic lexical resources, such as WordNet.

Our proposal is a friendly, simple and cooperative Question Answering system. The main difference is the cooperative way that answers the natural language questions posed by the user. We interact with the user in order to disambiguate and/or to guide the path to obtain the correct answer to the query posted, whenever this is possible to do by the reasoner. We also use the cooperation to provide more informed answers. The answers have to clarify what the system can infer about the question from the knowledge domain.

## 4 CONCLUSIONS AND FUTURE WORK

We presented a Cooperative Semantic Web Question Answering system that receives queries expressed in natural language and is able to return a cooperative answer, also in natural language, obtained from Semantic Web resources (Ontologies and OWL2 Descriptions). The system is able of dialoguing when the question has some ambiguity or when it detects that the answer is not what user expected. Our proposal includes deep parsing, use of ontologies and other web resources such as the WordNet and the DBpedia.

As future work, we intend to answer questions that are more elaborate and/or more difficult. Moreover, we also plan to extend to the Portuguese Natural Language. For this purpose it will be necessary to enrich the knowledge domain with concepts that may be deduced from the initial domain. Although the system is intended to be domain independent it will be tested in a number of domains, with special relevance to the wine and the cinema since for these fields there are many resources available in the Semantic Web. We also plan to build a DRS generator, that builds the question semantics and retains additional information that allows the Discourse Controller to build a more adequate cooperative answer. We contemplate enlarging the knowledge base with other ontologies in order to support open domain Question Answering and take advantage of the vast amount of heterogeneous semantic data provided by the Semantic Web.

## REFERENCES

Almendros-Jiménez, J. M. (2011). A Prolog-based Query Language for OWL. *Electronic Notes in Theoretical Computer Science*, 271:3–22.

Auer, S., Bizer, C., Kobilarov, G., and Lehmann, J. (2007). Dbpedia: A nucleus for a web of open data. *The Semantic Web*, 4825(Springer):722–735.

Blackburn, P. and Bos, J. (2005). *Representation and inference for natural language: A first course in computational semantics*. Center for the Study of Language and Information.

Covington, M. A. (1988). From English to Prolog via Discourse Representation Theory ACMC Research Report 01-0024 Introduction / Abstract. *Representation Theory*, pages 1–35.

Fellbaum, C. (1998). *WordNet: An electronic lexical database*. The MIT press.

Guo, Q. and Zhang, M. (2008). Question answering system based on ontology and semantic web. In *Proceedings of the 3rd international conference on Rough sets and knowledge technology*, pages 652–659. Springer-Verlag.

Guo, Q. and Zhang, M. (2009). Question answering based on pervasive agent ontology and Semantic Web. *Knowledge-Based Systems*, 22(6):443–448.

Horrocks, I. (2008). Ontologies and the semantic web. *Communications of the ACM*, 51(12):58.

Kamp, H. and Reyle, U. (1993). *From Discourse to Logic*, volume 42 of *Studies in Linguistics and Philosophy*. Kluwer.

Lopez, V., Fernández, M., Motta, E., Sabou, M., and Uren, V. (2007a). Question Answering on the Real Semantic Web. In *6th International and 2nd Asian Semantic Web Conference (ISWC 2007+ ASWC 2007)*, pages 2–4.

Lopez, V. and Motta, E. (2006). Poweraqua: Fishing the semantic web. *Semantic Web: Research and Applications*.

Lopez, V., Motta, E., and Uren, V. (2007b). AquaLog: An ontology-driven Question Answering System as an interface to the Semantic Web. *Journal of Web Semantics*.

Nogueira, V. and Abreu, S. (2007). Temporal contextual logic programming. *Electronic Notes in Theoretical Computer Science*, 177:219–233.

Quaresma, P., Rodrigues, I., Prolo, C., and Vieira, R. (2006). Um sistema de Pergunta-Resposta para uma base de Documentos. *Letras de Hoje*, 41(2):43–63.

Rabelo, J. and Barros, F. (2004). Pergunte! uma interface em português para pergunta-resposta na web. *Master's thesis, Informatics Center, Federal University of Pernambuco, Brazil*, pages 1114–1117.

Rodrigues, I. P., Ferreira, L., and Quintano, L. (2009). NL Database Dialogue Question-Answering as a Constraint Satisfaction Problem. In Abreu, S. and Seipel, D., editors, *Proceedings of the 18th International Conference on Applications of Declarative Programming and Knowledge Management (INAP 2009)*.

Wielemaker, J. (2005). An optimised Semantic Web query language implementation in Prolog. *Logic Programming*, pages 128–142.

Wielemaker, J., Hildebrand, M., and van Ossenbruggen, J. (2007). Using Prolog as the fundament for applications on the semantic web. *Proceedings of ALPSWS2007*, pages 84–98.

Witzig, S. and Center, A. (2003). Accessing wordnet from prolog. *Artificial Intelligence Centre, University of Georgia*, pages 1–18.