

Cooperative Question Answering for the Semantic Web

Dora Melo¹, Irene Pimenta Rodrigues², and Vitor Beires Nogueira²

¹ Instituto Politécnico de Coimbra and CENTRIA, Portugal
dmelo@iscac.pt,

² Universidade de Évora and CENTRIA, Portugal
{ipr,vbn}@di.uevora.pt

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 represented in OWL/RDF as knowledge base and WordNet 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 information sources. 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.

Keywords: Natural Language, Ontology, Question Answering, Semantic Web

1 Introduction

Ontologies and the semantic web [1] became a fundamental methodology to represent the conceptual domains of knowledge and to promote the capabilities of semantic question answering systems [2]. By allowing search in the structured large databases and knowledge bases of the semantic web these systems 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.

In this paper we propose a cooperative question answering 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 that provides a context for the interpretation of the questions and includes implicit content such as spatial and temporal

knowledge, entities and information useful for the pragmatic interpretation like discourse entities used for anaphora resolution. 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 considers only the English natural language and includes deep parsing, use of ontologies, lexical and semantic resources such as the WordNet [3] and web resources like the DBpedia [4].

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 the system 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.

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

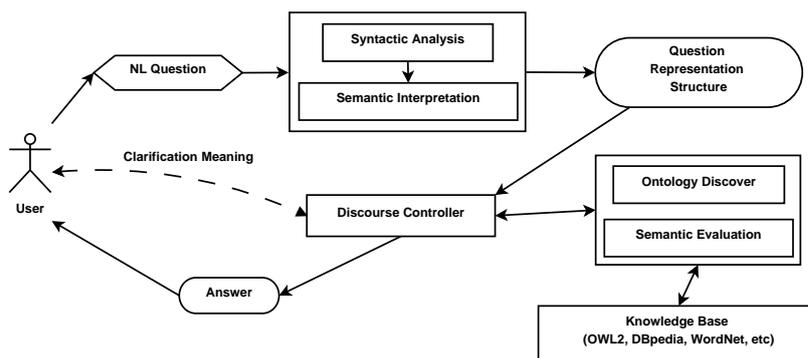


Fig. 1. Question Answering System Architecture.

2.1 Semantic Interpretation

Semantic analysis (or interpretation) is built using first-order logic [5] extended with generalized quantifiers [6]. We take special care with the discourse entities in order to have the appropriate quantifier introduced by the determinant interpretation. At this step, the syntactic structure of the question is rewritten into a DRS³, that is supported by Discourse Representation Theory [7].

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 study, it is stated by the following representation structure:

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

where the referent of the discourse is `all-X`, with `X` an universally quantified discourse entity, the main predication of the question is `died(X)` and the presupposed predications are `writer(Y)`, `french(Y)`, `romantic(Y)`, with `Y` an existential quantified discourse entity. The system has to find all entities `X` that must verify the main predication condition only for those entities `Y` that verify all the question presupposed conditions. The answer to the question will be a list with all french, romantic, writers resource entities who died.

2.2 Ontology Discovery

The Ontology Discovery 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 according to their string-based, taking into account abbreviations, acronyms, domain and lexical knowledge. If an answer is not achieved, each term in the query is extended with its synonyms, hypernyms and hyponyms obtained from WordNet⁴ [8]. 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 classes, properties or instances that have labels matching the search terms 'writer', 'french', 'romantic' and 'died', either exactly or partially. For instance, to represent the concept 'writer', the system finds the DBpedia class `Writer`⁵, with property domain `Work` and domain range `Person`.

³ For us a DRS is a set of referents, universally quantified variables and a set of conditions (first-order predicates). The conditions are either atomic (of the type $P(u_1, \dots, u_n)$ or $u_1 = u_2$) or complex (negation, implication, disjunction, conjunction or generalized quantifiers).

⁴ <http://wordnet.princeton.edu/>

⁵ <http://dbpedia.org/ontology/Writer>

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 constrain the discourse entities. Moreover, this extra information (regarding the question interpretation) can help the Discourse Controller to formulate a more objective answer.

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

Back to our example, to solve the constraint problem the Dialogue Controller generates and poses the questions such "Who are the French romantic writers?" to the question answering system, whose representation structure is

```
drs([wh-X,exist-Y],[writer(Y), french(Y), romantic(Y),
is(X,Y)]).
```

First and according to the domain knowledge, the interpreter will transform the conditions of the DRS into OWL. For instance, the condition `ontology_writer` will represent the DRS condition `writer`. Therefore, the new representation structure⁶ for the question is

```
drs([wh-X,exist-Y],[ontology_writer(Y), ontology_french(Y),
ontology_romantic(Y), is(X,Y)])
```

After obtaining this new set of DRS, the terms of the ontology will be interpreted as usual Prolog predicates. Then, by applying the unification mechanism of Prolog the system will obtain the answer to the question. Therefore, the answer to initial question will be

```
Francois-Rene de Chateaubriand (1768-1848)
Alphonse de Lamartine (1790-1869)
Alfred de Musset (1810-1857)
Victor Hugo (1802-1885)
Henri-Marie Beyle, Stendhal (1783-1842)
```

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

⁶ The condition `ontology_term` represents the class, property or instance in the ontology that is the meaning of the term. If the interpreter has more than one possible ontology conditions for each term then will get several DRS rewritten with the terms of the ontology.

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.

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. 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.

Our dialoguing system has as main objective the use of interaction 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 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 [9] 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 [10]. The author proposes 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.

In [11] we find 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 [12] 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.

Our proposal is a friendly, simple and cooperative question answering system. The main difference is the cooperative way on answering 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, whenever this is possible to do by the reasoner. We also use 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. The system is able of dialoguing when the question has some ambiguity or when it detects that the answer is not what the user expected. Our proposal includes deep parsing, the use of ontologies, lexical and semantic resources such as the WordNet and web resources like the DBpedia.

As future work, we intend to answer more complex questions and extend it to 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 contemplate about 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

1. Horrocks, I.: Ontologies and the semantic web. *Communications of the ACM* **51** (2008) 58
2. Guo, Q., Zhang, M.: Question answering based on pervasive agent ontology and Semantic Web. *Knowledge-Based Systems* **22** (2009) 443–448
3. Fellbaum, C.: *WordNet: An electronic lexical database*. The MIT press (1998)
4. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J.: Dbpedia: A nucleus for a web of open data. *The Semantic Web* **4825** (2007) 722–735
5. Hodges, W.: *Classical logic I: first-order logic*. The Blackwell guide to philosophical logic (2001) 9–32
6. Barwise, J., Cooper, R.: Generalized quantifiers and natural language. *Linguistics and philosophy* **4** (1981) 159–219
7. Kamp, H., Reyle, U.: *From Discourse to Logic*. Volume 42 of *Studies in Linguistics and Philosophy*. Kluwer (1993)
8. Witzig, S., Center, A.: *Accessing wordnet from prolog*. Artificial Intelligence Centre, University of Georgia (2003) 1–18
9. Rodrigues, I., Quintano, L., Ferreira, L.: Nl database dialogue question-answering as a constraint satisfaction problem. In: *18th Intl. Conf. on Applications of Declarative Programming and Knowledge Management (INAP'09)*, Univ. Évora (2009)
10. Almendros-Jiménez, J.M.: A Prolog-based Query Language for OWL. *Electronic Notes in Theoretical Computer Science* **271** (2011) 3–22
11. Nogueira, V., Abreu, S.: Temporal contextual logic programming. *Electronic Notes in Theoretical Computer Science* **177** (2007) 219–233
12. Lopez, V., Motta, E.: *Poweraqua: Fishing the semantic web*. *Semantic Web: Research and Applications* (2006)