A Text-based Query Interface to OWL Ontologies

Danica Damljanovic, Valentin Tablan, Kalina Boncheva

Department of Computer Science
University of Sheffield
Regent Court, 211 Portobello Street
S1 4DP, Sheffield, UK
{d.damljanovic, v.tablan, k.boncheva}@dcs.shef.ac.uk

Abstract content

1. Introduction

One of the key preconditions for applying Semantic Web technologies to different domains is structuring the data, mostly in the form of ontologies. Ontologies enable presenting data in a machine-readable form, thus providing knowledge reuse and sharing between applications, reasoning and semantic search. Data created or annotated with regards to ontologies are usually stored in a repository. As this repository contains knowledge formalisms from the particular domain, it is usually referred to as a *knowledge store*. Tools for creating, editing and querying the knowledge store are widely developed up to date (e.g., Sesame, Jena). However, most of them - although giving a great power of expressiveness - require background knowledge and expertise in the field.

There are many existing Semantic Web query languages (e.g., SPARQL, SeRQL) for querying knowledge stores. These languages are usually complex and each has a specific syntax. For query construction and understanding of the results users must understand not only the language itself, but also ontologies and ontology languages. Even for an expert in this field, writing queries is an error-prone task whilst the syntax is not easy to learn.

In this paper we present an approach which enables users to formulate queries using natural language. However, the problem with natural language queries is that these tend to be ambiguous. One way to overcome this problem is by defining a *controlled language* for querying knowledge stores. A controlled language is a subset of a natural language that includes certain vocabulary and grammar rules that have to be followed. With a controlled language users can then create queries in the form of questions such as: "list hotels in Paris located by the river".

Similar or even the same query can be used as input for Web search engines, such as Google. Due to the way search engines work, almost equal results would be given for a much shorter query comprising only the most important concepts, namely: 'hotel Paris river'. This behavior has had a great impact on users, who can be thought of as being 'Googleized', i.e., they are expecting the same simple search-box interface and the same behaviour from any other language-based search interface.

In this paper we present our system, called CLOnE QL, for querying knowledge stores in natural language. As a knowledge store we consider a set of ontologies and the knowledge base containing instances of classes from ontologies and relations between them. Our approach supports both controlled language search and keyword search. Ambiguities in the queries are resolved by using reasoning over the ontology, in order to derive all valid possible interpretatons, which are then presented to the user.

CLOnE QL was inspired by CLOnE (Controlled Language for Ontology Editing): CLOnE provides users to edit ontologies using natural language (Tablan et al., 2006). Similarly, CLOnE QL enables users to query the existing knowledge by transforming input Natural Language queries into SeRQL queries using GATE for natural language processing.

2. Related work

To simplify the semantic search process, some knowledge management platforms such as KIM (Popov et al., 2003) provide an interface that enables querying knowledge bases by either instantiating a query from a set of given templates, or by constructing a SeRQL query using a form-based interface. Consequently, users are either restricted in what they can search for, or they need to be familiar with the query language and the underlying ontology. CLOnE QL differs from KIM in that it saves users not only from the complexity of the query language, but also from the necessity to be familiar with ontologies and semantic search.

AquaLog (Lopez and Motta, 2004) is perhaps the system closest to ours, as it also uses a controlled language for querying ontologies. It is also coupled with a learning mechanism, so that its performance improves over time, in response to the vocabulary used by the end users. This system heavily relies on language processing (Lei et al., 2006) and requires syntactically correct sentences. Our approach differs in that it is much more robust with respect to mistakes in the controlled language and, in addition to question-based queries, it also supports concept-based ones such as 'accommodation Rome'.

(Lei et al., 2006) present SemSearch - a concept-based system which claims to have Google-like Query Interface. It requires a list of concepts (classes or instances) as an input query separated by colon (e.g., 'news:PhD Students' is a query that results in all instances of class News that are in relation with PhD Students). The idea of having a simple form for semantic search queries is very good. On the other side, SemSearch does not consider properties, and neither does it disambiguate in cases when there is more than one relation between the two concepts. It also does not accept

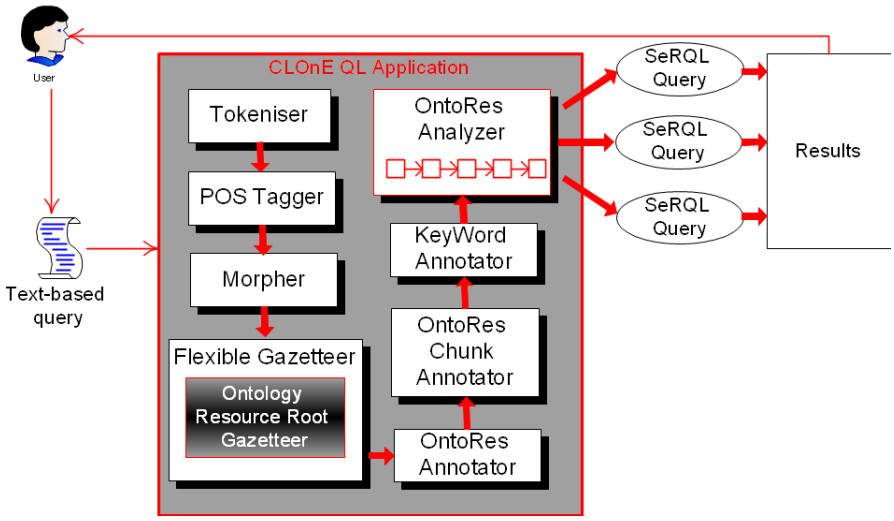

Figure 1: CLOnE QL Application

natural language questions such as 'how many PhD Students are registered at the University of Sheffield?'. Retrieving relevant properties is a very important task in our approach, resulting in SeRQL queries that are of a high accuracy.

The main advantage of our system in comparison to previous work is in its emphasis on robustness, i.e., it gives users the freedom to enter queries of any length and form. It uses heavily reasoning over the ontology, in order to disambiguate and interpret the user queries. Last but not least, the query interface is very simple (a Google-like search box), so it can be used without any prior training.

3. CLOnE QL Implementation

CLOnE QL is an Information Extraction application (see figure 1), based on the GATE language processing framework (Cunningham et al., 2002). This application is a pipeline of some generic GATE language processing resources such as tokeniser, sentence splitter, morphological analyser, part-of-speech tagger and gazetteer and several newly implemented ones:

OntoRes Annotator - for identifying ontology resources in the query.

OntoResChunk Annotator - for identifying chunks: part(s) of the query between identified ontology resources.

KeyPhrase Annotator - for identifying keywords and keyphrases from the controlled language.

OntoRes Analyser - analyses the identified ontology resources and chunks and uses reasoning in order to interpret and disambiguate the queries into the requeried SeRQL syntax.

The first step is identifying key concepts. We are matching all morphological inflections of the relevant terms by using a morphological analyzer in the dynamic construction of the gazetteer lists from the ontologies.

After key concepts are identified we investigate to find any potential relations between them stored inside the knowledge store. To retrieve these relations we use reasoning provided by the reasoning component of the knowledge store. Retrieved relations are then scored according to two factors. One of them is similarity of the relation's name with the part of the query (a chunk) between identified concepts. We compare them and give the highest score to the relation that is the most similar to the chunk. For this comparison we use *Levenshtein distance metrics*. The Levenshtein distance between two strings is the minimum number of operations needed to transform one string into the other, where an operation is an insertion, deletion, or substitution of a single character. The other relevant factor for scoring the properties is more complex and is based on its the position in the property hierarchy (if such exists).

After the retrieved relations are scored, SeRQL queries are constructed for the highest ranked ones. They are then executed and the results are shown to the user.

CLOnE QL is currently supporting queries with one to three recognized concepts. For example, for the knowledge store comprising a GATE domain ontology (http://gate.ac.uk/ns/gate-ontology) and a knowledge base containing instances from this ontology, query 'List Processing Resources' would result in listing all known instances of class 'Processing Resource'. 'List Processing Resources in ANNIE' would result in listing all processing resources (i.e. instances of class Processing Resource) that are in a relation with an instance 'ANNIE': in the GATE domain ontology, ANNIE is a plugin consisting of several Processing Resources. As CLOnE QL does not require strict adherence to syntax, the same results would be given for the query 'Processing Resources ANNIE'.

Additionally, CLOnE QL supports a limited number of relative clauses. An example is shown on figure 2 where in a given query three concepts are identified: 'parameters' - referring to the *ResourceParameter* class, 'PR' - referring to the *ProcessingResource* class, and 'ANNIE' - referring to the instance of a GATE plugin. Potential relations are iden-

tified between these resources and the appropriate SeRQL queries are constructed.

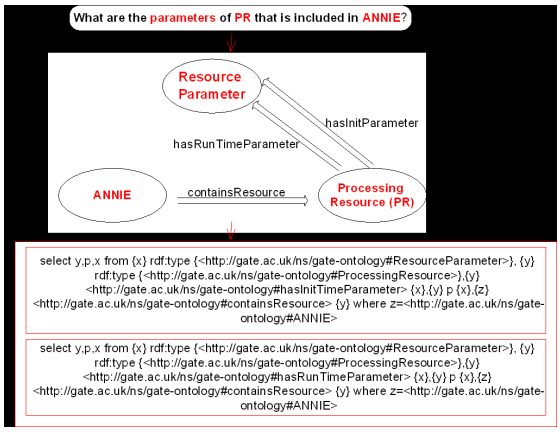

Figure 2: Supporting relative clauses with CLOnE QL

Last but not least, our system supports queries including conjunction and disjunction (see figure 3). These type of queries are processed so that first, those concepts connected with 'and' or 'or' are grouped. Further on, relations with other identified concepts are found for each member of the group separately.

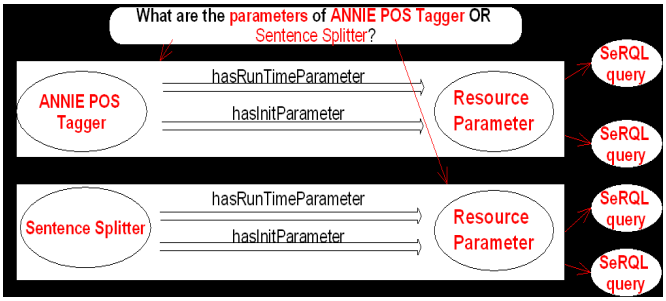

Figure 3: Supporting queries expressing conjunction/disjunction with CLOnE QL

In this given example, recognized concepts are 'parameters' - referring to the class *ResourceParameter*, 'ANNIE POS Tagger' - referring to the instance with this label and 'Sentence Splitter' - referring to the class with this label. *ANNIE POS Tagger* and *Sentence Splitter* are first grouped. Further on, potential relations between *ResourceParameter* and each member of the previously created group are found, and SeRQL queries are created accordingly.
An in-depth description of all components and algorithms will be provided in the complete paper.

4. Evaluation

In the full paper we will also present the results of a task-based black-box evaluation of the tool on a given domain ontology. We are using the GATE software ontology[1], developed as a part of the TAO project, which consists of concepts describing GATE's architecture, components, code, documentation, and publications. The populated knowledge base contains instances of classes from that ontology and relations between them. We will present results of quantitative (e.g., time taken to achieve task X) and qualitative measures (e.g., easy-to-use interface) taken based on questionnaires filled by the evaluation subjects.

5. Conclusion and future work

To summarise, CLOnE QL serves as a layer for transforming natural language queries into formal ontology query languages. This transformation eases the process of creating queries for expert users, whereas for non-experts it gives an opportunity to query the knowledge store without first having to learn its specialised query language. The main advantages of CLOnE QL over other similar systems are in its simple interface and support for queries of any length and form.
In future work we plan to explore language personalisation using a learning mechanism. We also plan to support user assistance and dialogue-based interaction for ambiguous queries, which will enable the system to ask users which of the possible interpretations is the one they require.

6. References

Hamish Cunningham, D. Maynard, K. Bontcheva, and V. Tablan. 2002. GATE: A framework and graphical development environment for robust NLP tools and applications. In *Proceedings of the* 40^{th} *Anniversary Meeting of the Association for Computational Linguistics (ACL'02)*, Philadelphia, USA, Jul. `http://gate.ac.uk/sale/acl02/acl-main.pdf`.

Y. Lei, V.S. Uren, and E. Motta. 2006. Semsearch: a search engine for the semantic web. In *Managing Knowledge in a World of Networks*, pages 238–245. Springer Berlin / Heidelberg.

Vanessa Lopez and Enrico Motta. 2004. Ontology driven question answering in AquaLog. In *NLDB 2004 (9th International Conference on Applications of Natural Language to Information Systems)*, Manchester, UK.

B. Popov, A. Kiryakov, D. Ognyanoff, D. Manov, A. Kirilov, and M. Goranov. 2003. Towards Semantic Web Information Extraction. In *Human Language Technologies Workshop at the 2nd International Semantic Web Conference (ISWC2003)*, Florida, USA.

V. Tablan, T. Polajnar, H. Cunningham, and K. Bontcheva. 2006. User-friendly ontology authoring using a controlled language. In *5th Language Resources and Evaluation Conference (LREC)*, Genoa, Italy, May. ELRA.

[1] http://gate.ac.uk/ns/gate-ontology