Towards Portable Controlled Natural Languages for Querying Ontologies

Danica Damljanovic

Department of Computer Science
University of Sheffield
Regent Court, 211 Portobello Street
S1 4DP, Sheffield, UK
{D.Damljanovic}@dcs.shef.ac.uk

Abstract. Natural Language Interfaces (NLIs) to structured data allow users to interact with a system using written or spoken language to perform tasks that require knowledge of a formal language. Due to natural language complexity and ambiguity, such interfaces usually support a Controlled Natural Language (CNL): a subset of a natural language that includes certain *vocabulary* and *grammar* rules that have to be followed. Building vocabulary differs from one system to another, and the way this is performed significantly affects *portability*: *portable* CNLs for querying ontologies are those that can be adapted easily to new domains without sacrificing performance. In this paper we describe the approach for dynamically extending the vocabulary supported by such systems, through a dialog with the user.

Key words: Controlled Languages, ontology, user interaction

1 Introduction

Natural Language Interfaces (NLIs) for querying structured data allow users to interact with a system using written or spoken language (e.g., English) to perform tasks that usually require knowledge of a formal query language. Due to Natural Language (NL) complexity and ambiguity, such interfaces usually support a Controlled Natural Language (CNL): a subset of a NL that includes certain *vocabulary* and *grammar* rules that have to be followed.

Building vocabulary for CNLs which are used for querying ontologies differs from one system to another, and the way this is performed significantly affects *portability*: *portable* NLIs to ontologies are those that can be adapted easily to new domains. Although portable NLIs are considered as potentially much more useful than domain-specific systems, constructing them poses a number of technical and theoretical problems as many of the techniques preclude automatic adaptation of the systems to new domains [1]. A recent trend in developing NLIs for querying ontologies includes building the domain vocabulary (lexicon) automatically from the ontology lexicalisations. However, when ontologies are

built using automated methods such as ontology learning or by automatic transition from relational databases, many ontology concepts have artificial names and do not include human-understandable lexicalisations. For example, artificial constructs such as a property with a local name *hasEmail* and no labels, would need to be preprocessed in order to be useful. Triples have a form of *(Subject, Predicate, Object)* and many existing tools make an assumption that the Predicate should be related to a verb in the sentence (or sometimes noun for datatype properties). However, this is often not the case, for many reasons. Firstly, requirement of having a unique URI means that for two object properties such as *hasEmail* and *hasAddress*, the URIs would be equal under the same namespace. In addition, it seems unlikely that users would use the verb *to have* when enquiring about an email, as they would probably use questions such as *What is your email?* instead of *What email you have?*. Informal rules which are often applied when naming ontology concepts indirectly affect the task of extracting the lexicon from the ontology. Many portable systems solve this problem by the means of customisation. As it is stated in [2], manual customisation increases recall. This is in line with the statement from [3] that "there is no free lunch" and that the customisation is mandatory in order to achieve reasonable performance.

In this paper, we propose a novel approach for minimizing the customisation of CNLs for querying ontologies without sacrificing performance, when porting them to work with another domain (ontology). We achieve this by modeling a dialog for the user in order to enrich already available vocabulary extracted from the ontology. Along the same lines, the dialog is modeled for any ambiguities which arise from the user's question, and the user is asked to disambiguate the specific meaning, before the question is translated to the formal language.

2 Context

Several CNL systems for querying ontologies have been developed recently, which extract the domain vocabulary (lexicon) from the ontology itself by extracting and processing the lexicalisations such as labels and datatype property values. Examples of such systems are ORAKEL [3], AquaLog [2], QuestIO [4] and many others. In case of ORAKEL, a part of the *domain-specific lexicon* is created automatically from the domain ontology, while another part is created manually and contains mappings of subcategorisation frames (e.g., verbs and nouns) to ontology properties.

Another way to generate/enrich the lexicon for CNLs for querying ontologies is by using the CNLs for knowledge representation, such as ACE [5] or CPL [6]. ACE is probably the most powerful, not only because of the maturity but also due to many support tools, such as OWL Verbaliser, which can be used to generate the lexicon from the ontology which is built externally; the lexicon can be updated/enriched by changing/adding new ACE sentences. While neither of the CNLs are tailored to a specific domain, porting them to a different domain requires knowledge of these CNLs in order to generate/update the domain knowledge.

Our approach generates the initial lexicon automatically from the ontology lexicalisations. When the user starts using the system, if a question term is not found in the lexicon, the combination of syntactic parse and ontology reasoning rules is used to generate the dialog. The user is then asked to map the unknown term into the ontology concept and following his selection, the new term is added to the lexicon. In addition, the lexicon carries the semantics which is related to the context in which certain word appeared.

3 Building lexicon through the user interaction

Generation of the user-defined lexicon is broken down into the following steps:

1. **Generate lexicon** by extracting lexicalisations attached to the Ontology Concepts[1]. This step includes extracting fragment identifiers, labels, and values of datatype properties.
2. **Perform lexicon-based lookup**. This would find the links between question terms and the logical form in the ontology. For example, in *What is the population of New York?*, *New York* would be identified as Ontology Concept (OC) referring to both `geo:newYork`[2] and `geo:newYorkNY`, because it is matched with the labels of these URIs which is *new york*.
3. **Analyse grammar** and identify the *candidate words* which could be referring to an Ontology Concept. We call these Potential Ontology Concepts (POCs). For example, for the above question, *population* would be identified as POC, however, as there is no such lexicalisation in the ontology, we do not know to which Ontology Concept this noun refers, and therefore, we ask the user.
4. **Generate the dialog** (if a POC cannot be mapped to the logical form automatically) and ask the user to map the unknown term (POC) into the specific concept in the ontology (OC). In addition, if POC refers to more than one OC, generate the dialog and ask the user to disambiguate. For example, in *What is the population of New York?* the question is ambiguous as it can be translated to two interpretations, where the first one is the *state population of New York state* and the other one is *city population of New York city*.
5. **Add the POC to the lexicon** as a description of the OC. This description includes the *context* in which the term appears so that it can be reused in similar contexts. Figure 1 illustrates an example of mapping *population* to the `geo:cityPopulation` whenever it appears together with *New York as a city*[3]. If the same word (population) is used together with *New York state*, then it will need to be mapped to a different OC such as `geo:statePopulation`.

[1] Note that we use the term Ontology Concept to refer to all types of ontology resources such as classes, instances, properties and literals.

[2] We use `geo:` instead of the full namespace *http://www.mooney.net/geo* for brevity.

[3] Note that the dialog before this one was asking the user to disambiguate whether *New York is a city or a state*

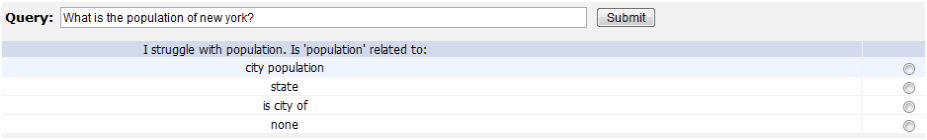

Fig. 1. A dialog where the user needs to select what population refers to

A dynamically enriched lexicon from the user-defined vocabulary is used in the system called FREyA[4] which serves as an NLI for querying ontologies. FREyA translates an NL query such as *What is the capital of California?* into the formal SPARQL query in order to find the answer. For more details about FREyA see [7]. However, the lexicon can be easily used by any CNL system for querying ontologies. Currently, its format is in JSON, and for the above example of 'mapping' *population* to the `geo:cityPopulation`, whenever this term is used in combination with `geo:City`, JSON would look similar to the following:

```
"Key:
    population
    http://www.mooney.net/geo#City",
"identifier":
    "http://www.mooney.net/geo#cityPopulation","function":""
```

The field *function* is used to indicate whether mapping certain words into OCs requires applying additional functions on their values (such as applying maximum or minimum function on the values of datatype properties of type number). For example, if in the question *What is the largest state in the US?*, there is *state* in the lexicon, which refers to `geo:State`, but there is no *largest*. The dialog is modeled and the user can map *largest* to the maximum value of the `geo:stateArea`, whenever it is used as a modifier of the lexicalisation of the `geo:State`:

```
"Key:
    largest
    http://www.mooney.net/geo#State",
"identifier":
    "http://www.mooney.net/geo#stateArea", "function":"max"
```

Translating this JSON format into the knowledge representation such as OWL in a way which can be used by any CNL system is straightforward. For example, the format of the OWL file could be such that ACE OWL Verbaliser generates proper ACE sentences so that the lexicon (content words) of ACE can be enriched.

[4] http://gate.ac.uk/freya

4 Evaluation

We experimented with 250 questions from the Mooney GeoQuery dataset[5] and the ontology covering the USA geography domain. The system without any customisation (domain lexicon generated automatically from the ontology) could automatically answer 72 questions (28.8%). For the remaining questions, FREyA generated at most 2 dialogs. When running it in the automatic mode (the lexicon is generated without engaging the user, but by selecting the first suggestion generated by FREyA) the precision and recall were 81.2%. Finally, by engaging the user for up to 15 minutes these values increased to 92.4%.

Acknowledgments We would like to thank A. Bernstein and E. Kaufmann from the University of Zurich, for sharing with us Mooney OWL ontology, and J. Mooney from University of Texas for making this dataset publicly available.

References

1. Grosz, B.J., Appelt, D.E., Martin, P.A., Pereira, F.C.N.: TEAM: An experiment in the design of transportable natural-language interfaces. Artificial Intelligence **32**(2) (1987) 173 – 243
2. Lopez, V., Uren, V., Motta, E., Pasin, M.: Aqualog: An ontology-driven question answering system for organizational semantic intranets. Web Semantics: Science, Services and Agents on the World Wide Web **5**(2) (June 2007) 72–105
3. Cimiano, P., Haase, P., Heizmann, J., Mantel, M., Studer, R.: Towards Portable Natural Language Interfaces to Knowledge Bases – the Case of the ORAKEL System. Data and Knowledge Engineering **65**(2) (May 2008) 325—354
4. Damljanovic, D., Tablan, V., Bontcheva, K.: A text-based query interface to owl ontologies. In: 6th Language Resources and Evaluation Conference (LREC), Marrakech, Morocco, ELRA (May 2008)
5. Fuchs, N.E., Kaljurand, K., Kuhn, T.: Attempto Controlled English for Knowledge Representation. In Baroglio, C., Bonatti, P.A., Małuszyński, J., Marchiori, M., Polleres, A., Schaffert, S., eds.: Reasoning Web, Fourth International Summer School 2008. Number 5224 in Lecture Notes in Computer Science, Springer (2008) 104–124
6. Clark, P., Harrison, P., Jenkins, T., Thompson, J., Wojcik, R.H.: Acquiring and Using World Knowledge Using a Restricted Subset of English. In Russell, I., Markov, Z., eds.: Proceedings of the 18th International FLAIRS Conference (FLAIRS'05), AAAI Press (2005) 506–511
7. Damljanovic, D., Agatonovic, M., Cunningham, H.: Natural Language Interfaces to Ontologies: Combining Syntactic Analysis and Ontology-based Lookup through the User Interaction. In: Proceedings of the 7th Extended Semantic Web Conference (ESWC 2010). Lecture Notes in Computer Science, Heraklion, Greece, Springer-Verlag (June 2010)

[5] `http://userweb.cs.utexas.edu/users/ml/nldata/geoquery.html`