Generating Knowledge Base Queries and Answer Descriptions

Laura Perez-Beltrachini
CNRS/LORIA
laura.perez@loria.fr

Enrico Franconi
Faculty of Computer Science
FUB
franconi@inf.unibz.it

Claire Gardent
CNRS/LORIA
claire.gardent@loria.fr

X. I. Juarez-Castro
Faculty of Computer Science
FUB
Ximena.JuarezCastro@stud-inf.unibz.it

Abstract

We discuss the task of verbalising Knowledge Base (KB) queries and answer descriptions in the context of the Quelo Natural Language Interface (NLI).

1 Introduction

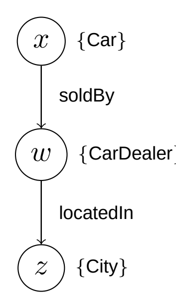
A major concern of research on Natural Language Interfaces (NLIs) to access structured data is the achievement of useful Natural Language (NL) based user-system interfaces (Webber et al., 1983; Benamara and Saint Dizier, 2003; Hastie et al., 2013). The Quelo Tool (Franconi et al., 2010) supports users in the formulation of precise queries against an underlying datasource without previous knowledge of the underlying datasource vocabulary and without the need to learn any formal language nor graphical interface. Quelo provides this intelligent NLI by following a Conceptual Authoring approach (Hallett et al., 2007). This NL Interface relies on an ontology describing the domain, reasoning based query manipulation operations and Natural Language Generation (NLG) techniques. In this paper we outline the generation task and the proposed generation approach.

2 Quelo Natural Language Interface

In Quelo NLI, the user formulates queries using natural language. Queries are formulated by adding, replacing or deleting snippets of English text at different points of the current query. The NLI interface links text spans of the NL query with elements of the underlying formal query (i.e. concepts and relations) and the NL queries are generated automatically using NLG techniques from possible query revisions computed by Quelo on the basis of the underlying formal query. The query formulation process starts from a general request expressed by the query "I am looking for something" which is iteratively refined by the user in interaction with the system graphical interface (Guagliardo, 2009; Franconi et al., 2010). Figure (1) (left) shows an example sequence of query refinements.

Quelo NLI also lets the user see which are those instances satisfying their current request. The interface presents the answer set of tuples resulting from query execution in a tabular format with a natural language description for each of the components of the tuple (Franconi et al., 2014). This description is generated with respect to the current query but from a different perspective as it introduces one or more instances in the tuples answering the query. Figure (1) (right) shows a NL query (Q) where the noun phrases ***"a car dealer"*** and ***"a car"*** are marked meaning that the user wants information about these entities. After the query is executed a NL description of the answer set is generated (A) with respect to the NL query.

3 The Generation Input



The Query Tool formal framework (cf. (Guagliardo, 2009)) defines a query as a labelled tree where edges are labelled with a relation name and nodes are labelled with a variable and a non-empty set of concept names from the ontology. The query tree can be expressed as a concept of Description Logic (DL) $\mathcal{L}$ (i.e. DL query):

$$\mathsf{Car} \sqcap \exists \mathsf{soldBy}.(\mathsf{CarDealer} \sqcap \exists \mathsf{locatedIn}.\mathsf{City})$$

The input to the generator is a linearisation of the query tree given by a depth-first traversal of the tree plus a precedence order defined on the children of and concept labels associated with a node (as defined in (Dongilli, 2008; Guagliardo, 2009)):

$$\mathsf{Car}_0 \sqcap \exists \mathsf{soldBy}_1.(\mathsf{CarDealer}_2 \sqcap \exists \mathsf{locatedIn}_3.\mathsf{City}_4)$$

4 Constraints on the Generation of KB Queries and Answer Descriptions

Generating KB queries and answer descriptions requires satisfying the following constraints.

Incrementality. Generation of KB queries should support the revisions, deletions and additions of text required by incremental specification by the user of her KB query.

Order Constraints. To avoid confusing the user, the revisions (modifications, extensions, deletions) performed by the user should have a minimal effect on the linear order of the NL query.

Fluency. Producing a fluent verbalisation of a KB query involves generating complex sentences or multi-sentence text. This requires the ability to choose the most appropriate syntactic construction for each concept or relation in different contexts.

Versatility. The verbalisation of query answers requires alternative syntactic constructions (i.e. paraphrases) to produce verbalisations from different perspectives.

Portability. The generator should be able to verbalise NL queries and query answer descriptions independently of the given KB.

5 A Grammar-Based generation approach

To provide a NL interface for querying KBs and visualising NL descriptions of query results, we developed a generation system which fits the above requirements. This generation system consists of automatically constructed lexicons, a hand-written unification-based grammar, a hypertagging module and an incremental surface realisation algorithm.

Lexicon Extraction. The lexicons used for generation map KB concepts and relations to words and syntactic rules. To allow for a portable approach, we automatically extract these lexicons from KBs using the approach described in (Trevisan, 2010). That is, lexicon entries are built by first tokenizing and pos tagging[1] concept and relation names. The resulting POS tag sequences are then used to associate the input KB symbol to one or more grammar rules using a pre-defined mapping.

[1]The POS tagger was trained on a corpus of KB symbols.

Hand-Written Grammar. To model the various syntactic constructions required for fluent verbalisations of KB queries and answer descriptions, we manually developed a Feature Based Lexicalised Tree Adjoining Grammar (FB-LTAG, (Vijay-Shanker and Joshi, 1988)) equipped with unification based compositional semantics ((Gardent, 2008)).

The grammar consists of 135 trees and captures the syntax of KB queries and is in this sense generic i.e., domain independent. When combined with an (automatically extracted) KB-specific lexicon, it permits generating NL queries and result descriptions specific to that knowledge base.

Hypertagger. To help identify the most fluent output and to prune the initial search space, we trained a CRF (Conditional Random Field) model to predict the sequences of grammar trees which yield the most fluent sentences.

Surface Realisation. The realisation algorithm is a chart-based algorithm adapted as described in (Perez-Beltrachini et al., 2014) for *Incremental generation* (The verbalisation of each query refinement is based on the current query's associated chart state and a local change fired by a query refinement step –add, remove or delete some content to the current verbalisation.) and *Order preserving generation* (Verbalisations whose surface elements are as close in order as possible to the underlying query linearisation are favoured during generation).

6 Conclusion

The generation approach we developed uses standard NLP techniques (computational grammar, chart based surface realisation algorithm, hypertagging) and adapts them to satisfies the particular requirements imposed by this specific setting. The grammar is tailored to the language of KB queries. The lexicon extraction process relies on the shape of KB relation names. And the surface realisation algorithm is modified to support incremental processing and order preservation. One main feature of the approach is that it is portable in that it permits querying arbitrary KBs independently of the domain being considered. This is made possible by a strict separation between lexicon and grammar rules, by a fully automated lexicon extraction process, and by a generic grammar describing the syntactic shape of KB queries.

I am looking for something. (Initial request)
· · · *for a new car.* (Substitution, Add Concept)
· · · *for a new car sold by a car dealer.* (Add Reln)
· · · *for a new car, a coupé sold by a car dealer.* (Add Concept)
· · · *for a new car sold by a car dealer.* (Deletion)
· · · *for a car sold by a car dealer.* (Deletion)
· · · *for a car sold by a car dealer located in a country.* (Add Reln)

(Q) *I am looking for a car sold by* [*a car dealer*] *located in* [*a city*].

(A)	*A car dealer:*	*The city in which he is located:*
	Jegla AG	*Rickenbach*
	Auto Center Wetzikon	*Wetzikon*

Figure 1: LEFT: Sequence of query refinements. RIGHT: A NL query and the NL descriptions generated for the answer set thereof.

References

Farah Benamara and Patrick Saint Dizier. 2003. Webcoop: A cooperative question-answering system on the web. In *Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics-Volume 2*, pages 63–66. Association for Computational Linguistics.

Paolo Dongilli. 2008. Natural language rendering of a conjunctive query. *KRDB Research Centre Technical Report No. KRDB08-3). Bozen, IT: Free University of Bozen-Bolzano*, 2:5.

Enrico Franconi, Paolo Guagliardo, and Marco Trevisan. 2010. An intelligent query interface based on ontology navigation. In *Proceedings of the Workshop on Visual Interfaces to the Social and Semantic Web (VISSW 2010)*, volume 565. Citeseer.

Enrico Franconi, Claire Gardent, Ximena I. Juarez-Castro, and Laura Perez-Beltrachini. 2014. Quelo Natural Language Interface: Generating queries and answer descriptions. In *ISWC2014, Natural Language Interfaces for Web of Data Workshop, NLI-WoD 2014*, Riva del Garda, Trentino, Italy, October.

Claire Gardent. 2008. Integrating a unification-based semantics in a large scale Lexicalised Tree Adjoining Grammar for French. In *COLING'08*, Manchester, UK.

Paolo Guagliardo. 2009. Theoretical foundations of an ontology-based visual tool for query formulation support. Master's thesis, KRDB Research Centre, Free University of Bozen-Bolzano, October.

C. Hallett, Donia Scott, and Richard Power. 2007. Composing questions through conceptual authoring. *Computational Linguistics*, 33(1):105–133.

Helen Hastie, Marie-Aude Aufaure, Panos Alexopoulos, Heriberto Cuayáhuitl, Nina Dethlefs, Milica Gasic, James Henderson, Oliver Lemon, Xingkun Liu, Peter Mika, et al. 2013. Demonstration of the parlance system: a data-driven, incremental, spoken dialogue system for interactive search. In *Proc SIGDIAL*, pages 154–156.

Laura Perez-Beltrachini, Claire Gardent, and Enrico Franconi. 2014. Incremental query generation. In *EACL 2014*, Gothenburg, Sweden, April.

Marco Trevisan. 2010. A portable menuguided natural language interface to knowledge bases for querytool. Master's thesis, Free University of Bozen-Bolzano (Italy) and University of Groningen (Netherlands).

K Vijay-Shanker and Aravind K Joshi. 1988. Feature structures based tree adjoining grammars. In *Proceedings of the 12th conference on Computational linguistics-Volume 2*, pages 714–719. Association for Computational Linguistics.

Bonnie Webber, Aravind Joshi, Eric Mays, and Kathleen McKeown. 1983. Extended natural language data base interactions. *Computers & Mathematics with Applications*, 9(1):233 – 244.