Applying Semantic Parsing to Question Answering over Linked Data: Addressing the Lexical Gap

Sherzod Hakimov, Christina Unger, Sebastian Walter, Philipp Cimiano
{shakimov, cunger, swalter, cimiano}@cit-ec.uni-bielefeld.de

Semantic Computing Group
Cognitive Interaction Technology – Center of Excellence (CITEC)
Bielefeld University
33615 Bielefeld, Germany

Abstract. Question answering over linked data has emerged in the past years as an important topic of research in order to provide natural language access to a growing body of linked open data on the Web. In this paper we focus on analyzing the lexical gap that arises as a challenge for any such question answering system. The lexical gap refers to the mismatch between the vocabulary used in a user question and the vocabulary used in the relevant dataset. We implement a semantic parsing approach and evaluate it on the QALD-4 benchmark, showing that the performance of such an approach suffers from training data sparseness. Its performance can, however, be substantially improved if the right lexical knowledge is available. To show this, we model a set of lexical entries by hand to quantify the number of entries that would be needed. Further, we analyze if a state-of-the-art tool for inducing ontology lexica from corpora can derive these lexical entries automatically. We conclude that further research and investments are needed to derive such lexical knowledge automatically or semi-automatically.

1 Introduction

The topic of question answering over linked data has started to receive substantial attention in the Semantic Web community [8], and benchmarking campaigns such as QALD[1] [7] have been organized in order to support the systematic comparison of different approaches on the same task, on a shared dataset and using the same evaluation protocol.

The main task in question answering over linked data can be framed as finding a mapping of natural language questions to SPARQL[2] queries which can then be executed over an RDF dataset. As an example, consider the question in 1 together with the given SPARQL query that can be executed over DBpedia in order to retrieve the answer.

[1] http://www.sc.cit-ec.uni-bielefeld.de/qald/

[2] http://www.w3.org/TR/sparql11-query/

1. Who was the first to climb Mount Everest?

```
PREFIX res: <http://dbpedia.org/resource/>
PREFIX dbo: <http://dbpedia.org/ontology/>
SELECT DISTINCT ?uri
WHERE {
       res:Mount_Everest dbo:firstAscentPerson ?uri .
}
```

The benchmarking challenges organized so far identified the *lexical gap* as one of the main problems in developing question answering approaches to linked data. The lexical gap refers to the problem that the vocabulary used by a user and the vocabulary used to formally represent the data can differ substantially. In the above example, for instance, the natural language question uses the expression *the first to climb*, while the corresponding property in the SPARQL query is called `firstAscentPerson`.

In order to develop a system that is successful and robust in mapping natural language questions to corresponding SPARQL queries, substantial lexical knowledge is needed, such as the knowledge that the property `firstAscentPerson` can be expressed as *the first to climb*. In this paper we analyze what lexical knowledge is needed for a question answering approach over linked data in order to be able to correctly interpret questions such as the above one. To this end, we have implemented the semantic parsing approach proposed by Zettlemoyer & Collins [14] and adapted it for the task of question answering over linked data.

As first contribution, we show that a vanilla implementation of this approach achieves poor results on the task. The main reason for this is that it was designed for a scenario in which the vocabulary used in the training data largely overlaps with the vocabulary used in the test data. This assumption, however, does not hold in open ended question answering systems over linked data in which the questions on which the system is trained can be rather different to the questions that actual users will ask, with respect to both wording and structure.

As second contribution, we investigate how much lexical knowledge would need to be added so that a semantic parsing approach can perform well on unseen data. We manually add a set of lexical entries on the basis of analyzing the test portion of the QALD-4 dataset. Further, we analyze if a state-of-the-art tool for inducing ontology lexica from corpora can derive these lexical entries automatically.

2 Semantic Parsing for Question Answering over Linked Data

In order to apply semantic parsing for question answering over linked data, we adapt Zettlemoyer & Collins' approach [14] (ZC05). This approach relies on Combinatory Categorial Grammar (CCG) [9, 10] for consituent-based syntactic representations, and typed-lambda calculus expressions [3] for semantic

Lexical item	Syntactic category	Semantic representation
Barack Obama	NP	`Barack_Obama`
is	(S\NP)/(S\NP)	$\lambda f.\lambda x.f(x)$
married to	(S\NP)/NP	$\lambda y.\lambda x.\texttt{spouse}(x,y)$
Michelle Obama	NP	`Michelle_Obama`

Table 1. Example CCG lexicon.

representations. A simple example of a CCG lexicon for the sentence *Barack Obama is married to Michelle Obama* is given in Table 1. The forward and backward application rules of CCG are applied to these lexical items in order to construct the parse tree of the sentence and its semantic representation $\texttt{spouse}(\texttt{Barack\_Obama}, \texttt{Michelle\_Obama})$.

Input to the ZC05 algorithm is a set of training examples (S_i, L_i) with $i = 1 \ldots n$, where each S_i is a sentence and each L_i is a corresponding semantic representation (*logical form*). The output is a pair (ϕ, θ), where ϕ is a set of features and θ is a vector of weights for those features.

At the heart of the algorithm is the method GENLEX(S,L). It takes as input a sentence S and a corresponding logical form L, and generates a set of potential lexical items with syntactic categories and semantics, and finally pairs them with all possible substrings of S using rules defined in [14]. The resulting lexical items are then used in the actual semantic parsing step, together with initially defined lexical items for domain-independent expressions, such as wh-words, prepositions, determiners, etc.

The actual semantic parsing step returns the highest scoring parses that derive the expected logical form L using all possible lexical items. Parsing itself is an iterative process: The first step uses all possible lexical items generated by GENLEX, and only those lexical items that were used in the successful parses are then passed to the second step of parsing, with newly estimated parameter values.

We re-implemented the algorithm following the description in [14], using CKY-style parsing and a stack decoder, and changing the parameter estimation step into perceptron updates as in [15]. In Table 2 we show the updated GENLEX rules to apply ZC05 semantic parsing approach. Newly added input triggers are highlighted in boldface. Domain-independent expressions were specified manually, based on the domain-independent expressions used in [14]. These expressions and 200 training examples from QALD-4 [11], used as input to the ZC05 algorithm, can be found at http://pub.uni-bielefeld.de/data/2715997.

In order to evaluate semantic parsing on the QALD-4 dataset, the provided SPARQL queries are automatically converted to semantic representations using the following translation rules:

- Every resource in the query is translated into a constant.
- Every property in the query is translated into a binary function.
- Every `COUNT` solution modifier is translated into the function constant *count*.

Input Trigger	Output Category and Example
Constant c	NP : c NP : $\texttt{dbr:Brooklyn\_Bridge}$
Arity-two predicate p	(S\NP)/NP : $\lambda x.\lambda y.p(y,x)$ (S\NP)/NP : $\lambda x\lambda y.\texttt{dbo:author}(y,x)$
Arity-two predicate p	(S\NP)/NP : $\lambda x.\lambda y.p(x,y)$ (S\NP)/NP : $\lambda x.\lambda y.\texttt{dbo:starring}(x,y)$
Arity-two predicate p	(S\NP)/NP : $\lambda g.\lambda x.\lambda y.p(y,x) \wedge g(y)$ (S\NP)/NP : $\lambda g.\lambda x.\lambda y.\texttt{dbo:crosses}(x,y) \wedge g(y)$
Arity-two predicate p	N/NP : $\lambda x.\lambda y.p(x,y)$ N/NP : $\lambda x.\lambda y.\texttt{dbo:officialColor}(x,y)$
Arity-two predicate p	N/NP : $\lambda g.\lambda x.\lambda y.p(y,x) \wedge g(y)$ N/NP : $\lambda g.\lambda x.\lambda y.\texttt{dbo:capital}(y,x) \wedge g(y)$
Arity-two predicate p **and constant c**	N : $\lambda x.p(x,c)$ N : $\lambda x.\texttt{rdf:type}(x,\texttt{dbo:River})$
Arity-two predicate p	(N\N)/NP : $\lambda x.\lambda g.\lambda y.p(y,x) \wedge g(y)$ (N\N)/NP : $\lambda x.\lambda g.\lambda y.\texttt{dbo:crosses}(y,x) \wedge g(y)$
Arity-two predicate p and constant c	N/N : $\lambda g.\lambda y.p(y,c) \wedge g(y)$ N/N : $\lambda x.\texttt{dbo:country}(x,\texttt{dbr:Germany}) \wedge g(x)$
argmax/min with second argument arity-two function f	NP/N : $\lambda g.\lambda x.\mathit{argmax/min}(g(x), f(x))$ NP/N : $\lambda g.\lambda x.\mathit{argmax}(g(x), \lambda d.\texttt{dbo:birthDate}(x,d))$

Table 2. GENLEX rules from Zettlemoyer & Collins [14] adapted to question answering over linked data.

3 Evaluation

After having trained the Zettlemoyer & Collins algorithm on the QALD-4 training set, the learned model was tested on the QALD-4 test set, comprising 50 questions. We excluded questions that require YAGO classes, `UNION`s, `ORDER BY` statements and `FILTER`s, leaving 37 questions with respect to which the results produced by the semantic parsing approach were compared to the QALD-4 gold standard results. For each question q, precision and recall were computed as follows:

$$\mathit{Recall}(q) = \frac{\text{number of correct system answers for } q}{\text{number of gold standard answers for } q}$$

$$\mathit{Precision}(q) = \frac{\text{number of correct system answers for } q}{\text{number of system answers for } q}$$

In addition, F1-measure is computed as the harmonic mean of precision and recall. Since the QALD-4 training queries cover only a small part of the DBpedia vocabulary, we decided to increase lexical coverage of the system by adding a

	Precision	Recall	F1	Correct
Learned lexicon + ontology labels	0.66	0.05	0.09	2
Learned lexicon + ontology labels + handcrafted items	0.93	0.70	0.80	26
Learned lexicon + ontology labels + M-ATOLL	0.70	0.18	0.30	7

Table 3. Results on the QALD-4 test dataset in terms of precision, recall and F-measure, together with the number of correctly answered questions (out of 37).

Expression	Syntax	Semantics
first to climb	N/NP	$\lambda x \lambda y.\texttt{dbo:firstAscentPerson}(x, y)$
artistic movement	N	$\lambda x \lambda y.\texttt{dbo:movement}(x, y)$
launched from	(S\NP)/NP	$\lambda x \lambda y.\texttt{dbo:launchPad}(y, x)$
extinct	N	$\lambda x.\texttt{dbo:conservationStatus}(x, \texttt{'EX'})$
German	N/N	$\lambda g \lambda x.g(x) \wedge \texttt{dbo:country}(x, \texttt{dbr:Germany}))$
taikonauts	N	$\lambda x.\texttt{rdf:type}(x, \texttt{dbo:Astronaut})$ $\wedge \texttt{dbo:nationality}(x, \texttt{dbr:China})$

Table 4. Example lexical items created for the QALD-4 test data.

lexical item for each DBpedia predicate and class on the basis of their label, according to the GENLEX rules in Table 2.

The test results are given in the first row of Table 3, where *correct* specifies the number of correctly answered questions (out of 37). Most prominently, recall turns out to be very low. This is because most of the expressions in the test questions appear neither in the training data nor among the DBpedia labels. Thus, the system lacks a great deal of lexical knowledge of expressions that were not seen during training.

For example, to answer the question *Who was the first to climb Mount Everest*, the system would need a lexical item such as shown in the first row of Table 4. Such an item is not present in the induced lexicon, neither is it contained among the ontology labels. In such cases we therefore need external lexical resources to bridge the lexical gap. In order to test how much additional lexical knowledge is needed, we manually handcrafted lexical items for the test data. Some examples are given in Table 4.

In total we created 54 lexical items. The results using those additional lexical items are presented in the second row in Table 3, showing that recall significantly increased, from 5% to 70%.

The system thus shows remarkable improvements by using the handcrafted lexical items. However, for large domains the required manual effort is not always feasible. Therefore we ran M-ATOLL [12, 13], a system that automatically extracts lexicalizations for ontology elements from a text corpus, on the predicates used in the training dataset. It managed to find 10 of the required 54 lexical items. Results using lexical items per predicate that were automatically extracted by M-ATOLL are shown in the third row in Table 3.

Despite the range of automatically and manually created lexical items, the system still failed to answer questions such as *What was Brazil's lowest rank in the FIFA World Ranking*. This is mainly due to the n-gram size used to match vocabulary elements with expressions occuring in the natural language question. Currently we consider only 4-grams, in order restrict the number of parse trees produced during semantic parsing, whereas 7-grams would be needed to map *lowest rank in the FIFA World Ranking* to the corresponding property `fifaMin`.

4 Related Work

A very prominent work on learning grammars for semantic parsing is Zettlemoyer & Collins [14], who proposed lexical induction and parameter estimation using pairs of questions and logical forms. Our learning algorithm is based on their approach but differs in the parameter estimation step, using perceptron-style updates (as in [15]) instead of gradient updates. Kwiatkowski et al. [5] proposed an approach for lexicon induction without using handcrafted domain-independent lexical items. The approach is based on an iterative splitting of the sentence and the logical form, such that the approach learns which splitting operation produces the most accurate lexical items. Preceding work by Kwiatkowski et al. [6] leverages the same splitting strategy but generalizes better by using templates for lexical items. Other work on semantic parsing with CCG is Artzi & Zettlemoyer [1, 2], and Krishnamurthy et al. [4] who apply semantic parsing to open-domain question answering.

Research on applying semantic parsing to Freebase has also gained a lot of attention, examples are Cai & Yates (2013); Kwiatkowski et al. (2013); Berant et al. (2013); Berant & Liang (2014); Reddy et al. (2014). Like our system, these systems need external lexical knowledge to parse unseen expressions during the test phase.

5 Conclusion

We have implemented the semantic parsing approach by Zettlemoyer & Collins [14] and adapted it to question answering over linked data. In order to quantify the effort needed to address the lexical gap, we have analyzed the amount of entries that would be needed in order to get acceptable results on the QALD-4 benchmark. By manually adding 54 lexical entries to the seed lexicon of the semantic parser we achieve a precision of 93% and a recall of 70%. We have further analyzed whether these lexical entries can be induced automatically from a corpus using the state-of-the-art ontology induction system M-ATOLL. While these preliminary results bear some promise, they also clearly show that automatic methods still leave a large part of the lexical gap open, that until now can only be filled manually, and that further research and investments are needed in techniques that induce lexical entries from corpora or by crowd-sourcing in order to build successful question answering systems over linked data.

References

1. Artzi, Y., Zettlemoyer, L.: Bootstrapping semantic parsers from conversations. In: Proceedings of the conference on empirical methods in natural language processing. pp. 421–432. Association for Computational Linguistics (2011)
2. Artzi, Y., Zettlemoyer, L.: Weakly supervised learning of semantic parsers for mapping instructions to actions. TACL 1, 49–62 (2013)
3. Carpenter, B.: Type-logical semantics. MIT press (1997)
4. Krishnamurthy, J., Mitchell, M.T.: Joint syntactic and semantic parsing with combinatory categorial grammar. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 1188–1198 (2014)
5. Kwiatkowski, T., Zettlemoyer, L., Goldwater, S., Steedman, M.: Inducing probabilistic CCG grammars from logical form with higher-order unification. In: Proceedings of the 2010 conference on empirical methods in natural language processing. pp. 1223–1233. Association for Computational Linguistics (2010)
6. Kwiatkowski, T., Zettlemoyer, L., Goldwater, S., Steedman, M.: Lexical generalization in CCG grammar induction for semantic parsing. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing. pp. 1512–1523. Association for Computational Linguistics (2011)
7. Lopez, V., Unger, C., Cimiano, P., Motta, E.: Evaluating question answering over linked data. Web Semantics Science Services And Agents On The World Wide Web 21, 3–13 (2013)
8. Lopez, V., Uren, V., Sabou, M., Motta, E.: Is Question Answering fit for the Semantic Web? A Survey. Semantic Web 2, 125–155 (2011)
9. Steedman, M.: Surface structure and interpretation. MIT press (1996)
10. Steedman, M.: The syntactic process, vol. 35. MIT Press (2000)
11. Unger, C., Forascu, C., Lopez, V., Ngonga Ngomo, A.C., Cabrio, E., Cimiano, P., Walter, S.: Question Answering over Linked Data (QALD-4). In: Cappellato, L., Ferro, N., Halvey, M., Kraaij, W. (eds.) Working Notes for CLEF 2014 Conference (2014)
12. Walter, S., Unger, C., Cimiano, P.: ATOLL - a framework for the automatic induction of ontology lexica. Data & Knowledge Engineering (2014)
13. Walter, S., Unger, C., Cimiano, P.: M-ATOLL: A framework for the lexicalization of ontologies in multiple languages. In: The Semantic Web ISWC 2014, Lecture Notes in Computer Science, vol. 8796, pp. 472–486. Springer International Publishing (2014)
14. Zettlemoyer, L.S., Collins, M.: Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. arXiv preprint arXiv:1207.1420 (2005)
15. Zettlemoyer, L.S., Collins, M.: Online learning of relaxed CCG grammars for parsing to logical form. In: In Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL-2007. Citeseer (2007)