squall2sparql: a Translator from Controlled English to Full SPARQL 1.1

S. Ferré

SQUALL2SPARQL: a Translator from Controlled English to Full SPARQL 1.1

Sébastien Ferré

IRISA, Université de Rennes 1
Campus de Beaulieu, 35042 Rennes cedex, France
Email: ferre@irisa.fr

Abstract. This paper reports on the participation of the system SQUALL2SPARQL in the QALD-3 question answering challenge for DBpedia. SQUALL2SPARQL is a translator from SQUALL, a controlled natural language for English, to SPARQL 1.1, a standard expressive query and update language for linked open data. It covers nearly all features of SPARQL 1.1, and is directly applicable to any SPARQL endpoint.

1 Introduction

An open challenge of the Semantic Web [7] is *semantic search*, i.e., the ability for users to browse and search semantic data according to their needs. Semantic search systems can be classified according to their *usability*, the *expressive power* they offer, their *compliance* to Semantic Web standards, and their *scalability*. The most expressive approach by far is to use SPARQL [11], the standard RDF query language. SPARQL 1.1[1] features graph patterns, filters, unions, differences, optionals, aggregations, expressions, subqueries, ordering, etc. However, SPARQL is also the least usable approach, as it is defined at a low level in terms of logic (e.g., variables, Boolean conditions) and relational algebra (e.g., UNION, GROUP BY). A more usable approach is *question answering*, where users can express their questions in natural language. Natural language is used in search engines in various forms, going from full natural language (e.g., FREyA [2], Aqualog [9]) to mere keywords (e.g., NLP-Reduce [8]) through controlled natural languages (e.g., Ginseng [1]). Questions in natural language are generally translated to SPARQL queries, but in general, only a small fragment of SPARQL is used. This means that even if full natural language is allowed, expressiveness is in fact strongly limited.

The objective of QALD challenges (Question Answering over Linked Data) is to challenge, evaluate, and compare Semantic Web systems on the task of question answering on large and real linked datasets: DBpedia and MusicBrainz. This paper reports on the participation of SQUALL2SPARQL in the question answering task (in English) for DBpedia in QALD-3 open challenge.

SQUALL (Semantic Query and Update High-Level Language) is a Controlled Natural Language (CNL) for English that has full compliance with Linked Open

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

Data (LOD), and covers nearly all features of SPARQL 1.1, for both queries and updates. The advantage of CNLs is to provide a natural language syntax while retaining the precision and lack of ambiguity of formal languages like SPARQL [6]. The main drawback of CNLs is that users have to learn the language and its disambiguation rules. The system SQUALL2SPARQL[2] is a Web application that supports the translation from SQUALL to SPARQL, as well as the direct querying of SPARQL endpoints, like DBpedia.

The objective of our participation in the QALD-3 question answering task was to evaluate the capability of SQUALL to express English questions in a natural and precise way, and the capability of SQUALL2SPARQL to correctly translate SQUALL questions into SPARQL. Therefore, the measures of precision and recall are not enough to evaluate our approach, and the naturalness of SQUALL questions has also to be assessed.

2 Approach

Our system SQUALL2SPARQL, as its name suggests, is a translator from SQUALL to SPARQL. Given a SQUALL sentence, the system first translates it into an intermediate logical representation using a Montague grammar [3]. The intermediate representation is then translated into SPARQL, simply mapping logical constructs to combinations of SPARQL constructs. The produced query can then be sent to any SPARQL endpoint, and results returned. We have shown that SQUALL covers most features of SPARQL 1.1, including aggregations, expressions, updates, and named graphs. The only missing features are graph-level updates (e.g., `LOAD`), federated queries (i.e., `SERVICE`), and transitive closures of complex property paths (e.g., `(author/^author)+`).

We now briefly describe SQUALL as a controlled natural language. Content words are nouns, verbs, and proper nouns. Nouns (e.g., *Person*) and intransitive verbs are interpreted as class URIs or built-in unary predicates. Relation nouns (e.g., *child*, *birthPlace*) and transitive verbs (e.g., *matches*) are interpreted as property URIs or built-in binary predicates. Proper nouns (e.g., *res:Germany*) are interpreted as entity URIs. Grammatical words are determiners (e.g., *a. the*, *every*, *at least 10*), auxilliary verbs (e.g., *is*, *has*), predefined verbs (e.g., *shares*, *relates*), imperative verbs (e.g., *give me*, *return*), comparative and superlative adjectives (e.g., *higher*, *later*, *most*), aggregation nouns and adjectives (e.g., *number*, *average*), interrogative determiners and pronouns (e.g., *what*, *which*, *how many*), coordinations (e.g., *and*, *or*, *not*), and others. Boolean coordinations can be applied to most types of phrases: noun phrases, verb phrases, relative propositions, and sentences. Every proposition has a subject, a verb, and also an object if the verb is transitive. A sentence can be an open question (e.g., starting with *Which* or *What*), a closed question (e.g., starting with *Whether* or using auxiliary verbs and inversion), an imperative-style question (e.g., starting with *Give me* or *Return*), or an assertion (for updates).

[2] Web forms, examples, and source code can be found from the SQUALL homepage: `http://www.irisa.fr/LIS/softwares/squall`.

Complete examples of SQUALL questions are given in the following sections. More details and examples about the SQUALL language and its translation to SPARQL can be found in previous papers [4,5].

3 Resources

The use of SQUALL2SPARQL in QALD-3 assumes that English questions are reformulated in SQUALL, i.e. Controlled English. Its syntax is regular and sufficiently similar to English so that it can be learned without too much effort. Many examples are available on the SQUALL's Web page. Its vocabulary (i.e., nouns and verbs) is made of URIs because there is so far no lexical treatment in SQUALL2SPARQL. This has the obvious drawback that SQUALL queries look less natural, and that URIs have to be known or discovered manually. However, the advantage is that SQUALL2SPARQL is directly applicable to any LOD dataset, because no linguistic resource is required. If such linguistic resource is available, like those produced by the lexicon task of the QALD-3 challenge, it could be combined in SQUALL, using words instead of URIs.

From the training phase, we already learned some of the DBpedia vocabulary, and other URIs were found manually with Google searches and DBpedia browsing. We spent on average a few minutes per question for the reformulation phase. The automatic translation to SPARQL takes much less time than SPARQL query evaluation, and is therefore not an issue.

For illustration purposes, we list below a few original questions along with their SQUALL reformulation. The full list of SQUALL questions can be found in the official results of the QALD-3 open challenge.

1 *Which German cities have more than 250000 inhabitants?*
Which Town that has country res:Germany has a populationTotal greater than 250000?

2 *Who was the successor of John F. Kennedy?*
Who is the successor of res:John_F._Kennedy?

3 *Who is the mayor of Berlin?*
Who is the leader of res:Berlin?

4 *How many students does the Free University in Amsterdam have?*
What is the numberOfStudents of res:Vrije_Universiteit?

5 *What is the second highest mountain on Earth?*
Which Mountain has the 2nd highest elevation?

7 *When was Alberta admitted as province?*
What is the dbp:admittancedate of res:Alberta?

9 *Give me a list of all trumpet players that were bandleaders.*
Give me all Person-s whose instrument is res:Trumpet and whose occupation is res:Bandleader.

12 *Give me all world heritage sites designated within the past five years.*
Give me all WorldHeritageSite whose dbp:year is between 2008 and 2013.

15 *What is the longest river?*
Which River has the highest dbp:length?

21 *What is the capital of Canada?*
What is the capital of res:Canada?
23 *Do Prince Harry and Prince William have the same mother?*
Has 'Prince Harry' the same dbp:mother as 'Prince William'?
26 *How many official languages are spoken on the Seychelles?*
How many officialLanguage-s of res:Seychelles are there?
28 *Give me all movies directed by Francis Ford Coppola.*
Give me all Film-s whose director is res:Francis_Ford_Coppola.
32 *How often did Nicole Kidman marry?*
How many spouse-s of res:Nicole_Kidman are there?
74 *When did Michael Jackson die?*
What is the deathDate of res:Michael_Jackson?

4 Results

Out of the 99 questions, we got the right answers for 80 questions (including the three OUT OF SCOPE questions), and partial answers for 13. Recall is 0.88, precision is 0.93, and the F-measure is 0.90. Errors come:

- from heterogeneity in data (12 errors, questions 1, 6, 17, 19, 29, 33, 39, 60, 63, 72, 93, 96),
- from the user reformulation in SQUALL (2 errors, questions 14, 43),
- from SQUALL2SPARQL (2 errors, questions 49, 59),
- from the gold standard (2 errors, question 16, 75),
- from the endpoint (1 error, question 92).

Looking at heterogeneity errors in detail, it appears that most of them could be solved simply by: either adding generic super-properties in the DBpedia ontology, or by expanding common words (e.g., location, date) into UNION graph patterns. For example, in question 39 *"Give me all companies in Munich."*, the implicit relation *"has location"* can be translated in any of the three RDF properties: `dbo:location`, `dbo:headquarter`, `dbo:locationCity`. This explains why our reformulation in SQUALL *"Give me all Company-es whose location is res:Munich."* has recall 0.6 only (the default prefix was used for DBpedia ontology, so that `location` stands for `dbo:location`). If `location`, or another property, was defined as a super-property of the other properties, the same SQUALL question would have recall 1. Alternatively, assuming linguistic knowledge, the word *"location"* could be mapped to the graph pattern

```
{ ?x dbo:location ?y }
 UNION { ?x dbo:headquarter ?y }
 UNION { ?x dbo:locationCity ?y }
```

where `?x` and `?y` respectively stand for the subject and object of the relation. Such graph patterns could easily be exploited in the translation from the intermediate representation to SPARQL without the need to change the SQUALL language and its parsing.

Another problem related to heterogeneity is that some expected domain and range axioms are not verified in some cases. For example, in question 19 *"Give me all people that were born in Vienna and died in Berlin."*, 2 out of the 6 expected answers are not instances of the class `Person`. This is why our reformulation *"Give me all Person-s whose birthPlace is res:Vienna and whose deathPlace is res:Berlin."* missed 2 answers, even though it is arguably equivalent to the original formulation.

The errors coming from the user reformulation of questions are due to misspelling or misunderstanding of URIs. In question 14, *"res:Prodigy"* was used instead of *"res:The_Prodigy"*. In question 43, the property *"dbp:breed"* was used in the wrong direction.

The errors coming from SQUALL2SPARQL are due to an incorrect translation of the special verb *"share"*. For example, Question 49 *"Which other weapons did the designer of the Uzi develop?"* was reformulated as *"Which Weapon shares the dbp:designer with res:Uzi?"*, which returns *"Uzi"* itself as an answer. Another possible reformulation is *"Which Weapon has the same dbp:designer as res:Uzi?"*, but it exhibits the same error.

The error from the endpoint is because the BIND construct of SPARQL is not (yet) supported by the QALD-3 endpoint. It is possible to write the SPARQL query to avoid it, but SQUALL2SPARQL relies on it to simplify the translation from SQUALL. Note that the correct answers are returned when using the official DBpedia endpoint.

Regarding the naturalness of SQUALL sentences, most of them are not much longer than the original ones, and can be understood without learning SQUALL. Most differences fall into three categories:

1. reformulating the question to make it agree with SQUALL's grammar,
2. replacing a word by another (e.g., *movie* → *Film*),
3. making explicit some relations (e.g., *"is a chemist"* → *"has profession res:Chemist"*).

5 Discussion

We here discuss a few directions to go in order to improve the usability and performance of our approach.

Lexicons. The data independence of SQUALL is valuable as it allows to query all LOD with neither preparation nor linguistic resources. However, when such linguistic resources are available [12], it is a shame not to use them, as they could improve recall, and make SQUALL sentences much more natural at the lexical level. The useful format of lexicons for SQUALL2SPARQL would be mappings from words to graph patterns. Nouns and intransitive verbs would be mapped to RDF classes or mono-dimensional graph patterns (one free variable), and relational nouns and transitive verbs would be mapped to RDF properties or bi-dimensional graph patterns (two free variables). Such lexicons may be extended to adjectives, adverbs, and prepositions by corresponding extensions of

SQUALL's syntax. A candidate format of lexicons is *lemon* (lexicon model for ontologies) [10].

Multilinguality. This is an aspect of the QALD-3 challenge that we did not address. A priori, it suffices to define a different concrete syntax for each language, keeping unchanged the intermediate representation. However, for some languages, it may be more difficult than for English whose morphology is less complex than many other languages.

Guidance. Writing questions in a controlled natural language is easier than in a formal language like SPARQL. However, it is still error-prone, and may be frustrating for users. A possible solution that has already been used for CNLs is an auto-completion mechanism that suggests possible completions for the sentence, based on the grammar (e.g., Ginseng [1]). However, this is mostly useful to avoid grammatical errors, but not so helpful to find the right content words. Another participant of the challenge, Scalewelis, proposes a content-based guided approach, where users can build in a flexible way complex queries without the need to know the grammar or the content words.

References

1. Bernstein, A., Kaufmann, E., Kaiser, C.: Querying the semantic web with Ginseng: A guided input natural language search engine. In: Work. Information Technology and Systems (WITS) (2005)
2. Damljanovic, D., Agatonovic, M., Cunningham, H.: Identification of the question focus: Combining syntactic analysis and ontology-based lookup through the user interaction. In: Language Resources and Evaluation Conference (LREC). ELRA (2010)
3. Dowty, D.R., Wall, R.E., Peters, S.: Introduction to Montague Semantics. D. Reidel Publishing Company (1981)
4. Ferré, S.: SQUALL: a controlled natural language for querying and updating RDF graphs. In: Kuhn, T., Fuchs, N. (eds.) Controlled Natural Languages. pp. 11–25. LNCS 7427, Springer (2012)
5. Ferré, S.: SQUALL: a controlled natural language as expressive as SPARQL 1.1. In: Métais, E. (ed.) Int. Conf. Application of Natural Language to Information Systems (NLDB). pp. 114–125. LNCS 7934, Springer (2013)
6. Fuchs, N.E., Kaljurand, K., Schneider, G.: Attempto Controlled English meets the challenges of knowledge representation, reasoning, interoperability and user interfaces. In: Sutcliffe, G., Goebel, R. (eds.) FLAIRS Conference. pp. 664–669. AAAI Press (2006)
7. Hitzler, P., Krötzsch, M., Rudolph, S.: Foundations of Semantic Web Technologies. Chapman & Hall/CRC (2009)
8. Kaufmann, E., Bernstein, A.: Evaluating the usability of natural language query languages and interfaces to semantic web knowledge bases. J. Web Semantics 8(4), 377–393 (2010)
9. Lopez, V., Uren, V., Motta, E., Pasin, M.: Aqualog: An ontology-driven question answering system for organizational semantic intranets. Journal of Web Semantics 5(2), 72–105 (2007)

10. McCrae, J., Spohr, D., Cimiano, P.: Linking lexical resources and ontologies on the semantic web with lemon. In: Extended Semantic Web Conference (ESWC). pp. 245–259. LNCS 6643, Springer (2011)
11. Pérez, J., Arenas, M., Gutierrez, C.: Semantics and complexity of SPARQL. In: *et al*, I.F.C. (ed.) Int. Semantic Web Conf. pp. 30–43. LNCS 4273, Springer (2006)
12. Walter, S., Unger, C., Cimiano, P.: A corpus-based approach for the induction of ontology lexica. In: Int. Conf. Applications of Natural Languages to Information Systems (NLDB). pp. 102–113. LNCS 7934, Springer (2013)