SQID: Towards Ontological Reasoning for Wikidata

Maximilian Marx and Markus Krötzsch

Center for Advancing Electronics Dresden (cfaed), TU Dresden, Dresden, Germany
{maximilian.marx, markus.kroetzsch}@tu-dresden.de

1 Introduction

Modern data management has re-discovered the power and flexibility of graph-based representation formats, and so-called *knowledge graphs* are now used in many practical applications. The term originates from the *Google Knowledge Graph* [3], which uses a graph-structured knowledge base to deliver answers in Web search, but it has since been generalised to a wide range of applications not only at Google, Microsoft, and Facebook, but also in many other companies that employ graph databases. Large, freely avialable knowledge graphs include Bio2RDF [1], Freebase [2], Wikidata [9], and YAGO2 [4].

Knowledge graphs appear as an ideal application for semantic web technologies, which support declarative data management and knowledge modelling. Indeed, we find some use of RDF (e.g., in Bio2RDF) and SPARQL (e.g., in Wikidata's popular query service), but a significant part of applications relies on *ad hoc* data models and tool chains. At best, some projects rely on shared libraries such as Apache Tinkerpop to establish some compatibility. Ontological modelling is hardly used at all.

It has been argued that this is in part due to a mismatch between the capabilities of RDF (and, based on it, OWL) and the demands of knowledge graphs [5]. In particular, most applications require some form of *enriched* graph model, where edges are extended with *annotations*, used to capture many forms of auxiliary information that don't quite fit into the highly normalised graph model. Even RDF-based projects such as Bio2RDF use reification to express more complex, n-ary relationships, and it has been observed that this makes it impossible to use common ontology languages on such datasets [7].

In several recent works, we have therefore proposed more flexible ontology languages for knowledge graphs, based on the concept of *attributed logics* [8,6]. In this demonstration, we present a prototype for applying ontological reasoning with attributed logics to the Wikidata knowledge graph. We are facing the familiar "chicken or egg" problem: while attributed logics are a promising new ontology language, there is currently neither tool support (ontology reasoners, editors, file formats, parsers, ...) nor any data (actual ontologies) to motivate the development of such tools. To break this deadlock, we propose a lightweight approach that incentivises users to create ontological rules by providing useful examples and some (incomplete) reasoning support. These features are integrated into our Wikidata ontology and data browser and editor *SQID*, which has many useful features that are unrelated to its inferencing capabilities.

We envision that ontological reasoning will be of great utility for quality control on Wikidata, e.g., by ensuring that property constraints such as symmetry are maintained, and look forward to see interesting ontological axioms being created by users of Wikidata.

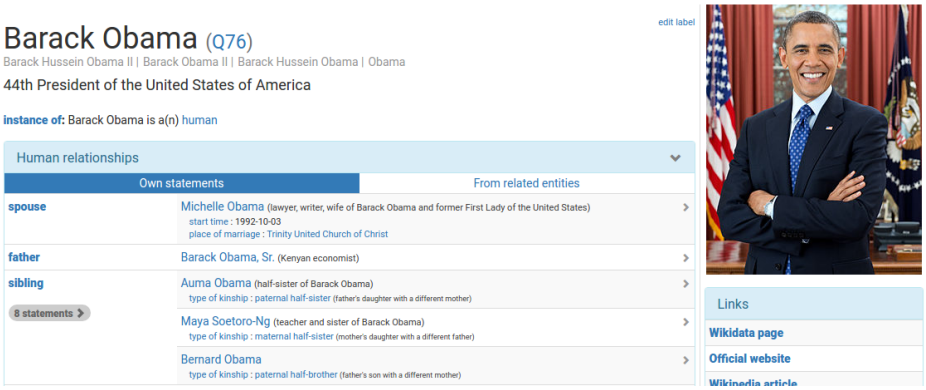

Figure 1. SQID view of Barack Obama

2 SQID: A Browser and Editor for Wikidata

Wikidata is a sister project of Wikipedia that manages factual data used in Wikipedia or any other Wikimedia project [9]. As of July 2017, Wikidata stores information about almost 29 million entities,[1] and has received contributions from over 175,000 registered contributors. Content from Wikidata is widely used in other applications and on the Web, ranging from interactive query views on specific subjects (e.g., the Academy Awards portal of the major German newspaper FAZ online[2]) to general purpose question answering tools (e.g., Apple's Siri search engine in iOS 11 beta returns data from Wikidata[3]).

Direct read/write access to raw Wikidata content is provided through Wikidata's web interface, using a very simple, general-purpose layout. For a more visually attractive view, we have created the SQID data browser,[4] a screenshot of which is shown in Fig. 1. This example also illustrates the enriched graph structure of Wikidata. We find, e.g., the statement that Obama's spouse is Michelle Obama (easily expressed in RDF), enriched with annotations for *start time* and *place of marriage* (not so easily expressed in RDF). In general, Wikidata allows arbitrary finite sets of attribute-value pairs to be used for annotating statements.

SQID is implemented as a browser application that integrates data obtained from Wikidata's live SPARQL query service (https://query.wikidata.org), full data dumps analysed offline, Wikidata's Web API, and other Wikimedia sources (e.g., for images). The basic data view already displays incoming relations (under "From related entities" in the figure) to improve browsing. In addition, SQID features class and property browsers, and query views that show larger lists of SPARQL results. Logged-in users moreover are offered some basic editing capabilities, e.g., for labels. We have extended this with the

[1] This should be compared to the 5.5 million articles found in English Wikipedia.
[2] http://www.faz.net/aktuell/feuilleton/kino/academy-awards-die-oscar-gewinner-auf-einen-blick-12820119.html
[3] https://lists.wikimedia.org/pipermail/wikidata/2017-July/010919.html
[4] https://tools.wmflabs.org/sqid/

Figure 2. Inferred statement for Barack Obama as displayed in SQID

ability to suggest new statements (via inferencing) that a logged-in user may approve to store them permanently in Wikidata.

3 Ontological Reasoning for Wikidata

We encode ontological knowledge using a notational variant of the recently-proposed rule language *MARPL* [8]. MARPL rules correspond to logical implications, where rule bodies are conjunctions of atomic conditions that refer to statements in a knowledge graph, including their annotation sets (such as start and place for Obama's marriage). MARPL therefore includes *object variables* that may stand for entities and values (as usual) but also *set variables* that may stand for such annotation sets. For example, we can express that the spouse relationship is symmetric, where all annotations are preserved (i.e., Michelle is married to Barack with *the same start time and place*):

$$\mathsf{spouse}(x, y)@S \rightarrow \mathsf{spouse}(y, x)@S \tag{1}$$

Here, the variables x, y and S are implicitly univerally quantified. Rather than simply copying all annotations, it is also often necessary to create new annotation sets for the conclusion. MARPL has a powerful mechanism for supporting this, but here we only show a simplified case using notation as for attributed description logics [6]. Using modelling similar to the statements in Fig. 1, we can, e.g., express that a person is related to a male parent of one of their parents, where the type of kinship is grandfather:

$$\begin{aligned}\mathsf{gender}(g, \mathsf{male})@S \wedge \mathsf{hasChild}(g, x)@T \wedge \mathsf{hasChild}(x, y)@U \\ \rightarrow \mathsf{relatedTo}(y, g)@[\mathsf{kinship} : \mathsf{grandfather}],\end{aligned} \tag{2}$$

Wikidata does not distinguish items and properties on the schema level – both can be subject and object in statements and annotations [5]. To capture this, we model Wikidata properties as individuals (not as predicates) and treat Wikidata statements as annotated ternary relations $\mathsf{statement}(s, p, o)@Q$ relating subject s, predicate p, and object o with annotation set Q (note that Wikidata refers to annotations as statement *qualifiers* [9]).

To make this special form of MARPL rules usable in software, we further introduce a customised syntax that allows rules to be expressed using only the restricted ASCII character set. For example, in our implementation, the rule (2) is written as follows (we use readable labels instead of the numeric ids Wikidata acutally uses for properties):

```
(?g.gender=male)@?S, (?g.hasChild=?x)@?T, (?x.hasChild=?y)@?U
   -> (?y.relatedTo=?g)@[kinship: grandfather]
```
(3)

Using this rule, we may, e.g., infer that Barack Obama has Stanley Armour Dunham as a grandfather, as shown in Fig. 2. Note that this is really how the grandfather relationship is encoded in Wikidata: there are no dedicated properties for most types of human relations.

For each item page in SQID, we are only interested in inferences that have the currently displayed item as a subject. Since any variable in the head predicate must also appear in the rule body, we may eliminate rules from consideration if the body requires an outgoing statement that is not present on the current item (in some cases, we can do this for incoming statements as well). For any rule not eliminated in such a fashion, we construct a SPARQL query that matches if the rule is applicable (but may match if the rule is not applicable), i.e., we query for an underapproximation of the rule body. For each query result, we then check if it gives rise to a match of the rule body, by verifying additional conditions on annotation sets that are not easily expressed in SPARQL. This yields a non-recursive and therefore incomplete, but nonetheless sound reasoner implementation that can work on the current version of the hundreds of millions of assertions in Wikidata.

4 Demonstration

In our demonstration, we will show the workings of our inference mechanism and the related user interface, but we will also give some general insights into the content, technical infrastructure and modelling approach of Wikidata as a whole. We will then demonstrate reasoning using a variety of inference rules. The SQID data browser is available at https://tools.wmflabs.org/sqid/. Reasoning support in SQID is still under active development and all source code is freely available at https://github.com/Wikidata/SQID/.

Acknowledgements. This work is partly supported by the German Research Foundation (DFG) in CRC 912 (HAEC), CoE cfaed, and in Emmy Noether grant KR 4381/1-1.

References

1. Belleau, F., Nolin, M., Tourigny, N., Rigault, P., Morissette, J.: Bio2RDF: Towards a mashup to build bioinformatics knowledge systems. J. of Biomedical Informatics 41(5), 706–716 (2008)
2. Bollacker, K., Evans, C., Paritosh, P., Sturge, T., Taylor, J.: Freebase: A collaboratively created graph database for structuring human knowledge. In: Proc. 2008 ACM SIGMOD Int. Conf. on Management of Data. pp. 1247–1250. ACM (2008)
3. Google Inc.: Knowledge – Inside Search. https://www.google.com/intl/es419/insidesearch/features/search/knowledge.html, retrieved July 2017 (2017)
4. Hoffart, J., Suchanek, F.M., Berberich, K., Weikum, G.: YAGO2: A spatially and temporally enhanced knowledge base from Wikipedia. J. of Artif. Intell. 194, 28–61 (2013)
5. Krötzsch, M.: Ontologies for knowledge graphs? In: Proc. 30th Int. Workshop on Description Logics (DL'17). CEUR Workshop Proceedings, vol. 1879. CEUR-WS.org (2017)
6. Krötzsch, M., Marx, M., Ozaki, A., Thost, V.: Attributed description logics: Ontologies for knowledge graphs. In: Proc. 16th Int. Semantic Web Conf. (ISWC'17). Springer (2017)
7. Krötzsch, M., Thost, V.: Ontologies for knowledge graphs: Breaking the rules. In: Groth, P.T., Simperl, E., Gray, A.J.G., Sabou, M., Krötzsch, M., Lécué, F., Flöck, F., Gil, Y. (eds.) Proc. 15th Int. Semantic Web Conf. (ISWC'16). LNCS, vol. 9981, pp. 376–392 (2016)
8. Marx, M., Krötzsch, M., Thost, V.: Logic on MARS: Ontologies for generalised property graphs. In: Sierra, C. (ed.) Proc. 26th Int. Joint Conf. on Artificial Intelligence (IJCAI'17). pp. 1188–1194. International Joint Conferences on Artificial Intelligence (2017)
9. Vrandečić, D., Krötzsch, M.: Wikidata: A free collaborative knowledgebase. Commun. ACM 57(10) (2014)