Tool Support for Ontology Design and Quality Assurance

Ian HORROCKS [a] and Jiaoyan CHEN [a] Jaehun LEE [b]
[a] *Department of Computer Science, University of Oxford, UK*
[b] *Samsung Electronics*

Abstract. We briefly describe some recent work where we have used a combination of machine learning and large-scale knowledge resources to support ontology design and quality assurance, including some ongoing work where we have identified a significant number of possible errors in the FoodOn ontology.

1. Introduction

Interest in semantic technologies, and in particular ontologies and knowledge graphs, continues to grow in both science and industry. In the industry case at least this is being fuelled by the increasing range, robustness and scalability of tools for *deploying* ontologies [5]. However, tool support for the development and maintenance of ontologies has advanced remarkably little over the last 10 years, with Protégé (or similar) still being the first choice for most ontology developers [7].

Development environments such as Protégé ensure that the resulting ontology is syntactically valid, and they invariably provide integrated reasoning support for identifying implicit subsumption relationships and logical inconsistencies. However, this is of only limited value in supporting design and quality assurance (QA): ontologies are invariably highly underspecified, so design inconsistencies may not manifest themselves in logical inconsistencies, and reasoning does little to address common errors such as omission and repetition.

A range of tools and methodologies have been developed to address this issue, many of them now available as Protégé plugins. These include, e.g., the use of foundational ontologies [11], the use of design patterns [12], taxonomy and visualisation-based tools such as OAF [8], and approaches based on modularisation [3] and alignment [4].

Notwithstanding all this effort, the development of (good quality) ontologies still heavily depends on highly skilled experts. The ontologies they develop are often large and complex, making it difficult to avoid and/or identify errors. Errors of omission can be particularly difficult to identify as monotonicity means that missing information won't lead to any inconsistency or any unexpected inferred subsumptions. The only certain way to identify missing information is via inspection by experts, but searching a large ontology for missing relationships can be difficult and time consuming.

Recent advances in machine learning, particularly in the way graph structures are handled, and the increasing number and size of available knowledge resources,[1] suggests

[1] AgroPortal (`http://agroportal.lirmm.fr/`), e.g., lists 124 ontologies in the agri-food domain alone.

a possible way to address this problem: we might be able to use these resources, possibly in combination with some learning process, as a surrogate for domain experts. Of course this kind of technique is itself imprecise (by its very nature), but if we can use it to identify possible errors, or even areas of interest, then we can optimise the use of precious ontology engineer and domain expert time.

In this short abstract we will briefly describe some of our recent work exploring this idea, including some ongoing work where we have identified a significant number of possible errors in the FoodOn ontology.

2. Ontology Design and Quality Assurance

Strictly speaking, an OWL ontology can contain both a conceptual model (sometimes called a TBox) and an instatiation of the model (sometimes called an ABox); in practice, however, the term ontology is often used to mean a conceptual model. Recently the term *knowledge graph* (KG) has become increasingly common, and has been used to describe everything from conceptual models with formal semantics through to rather informally specified graph structured knowledge. Our work targets all kinds of knowledge resources, and in the following we often won't distinguish between ontologies and knowledge graphs.

2.1. Literal Canonicalization

Although not so much of an issue in carefully crafted conceptual models, many large scale knowledge graphs contain large numbers of string literals instead of semantically typed entities; for example, in the RDF triple ⟨*River_Thames*, *passesArea*, *"Port Meadow, Oxford"*⟩. This weakens the KG as it does not capture the semantics of such literals, and can be seen as a form of repetition when the KG also includes a corresponding entity. This problem is pervasive and hence results in a significant loss of information: according to statistics from Gunaratna et al. [6], in 2016, the DBpedia property *dbp:location* has over 105,000 unique string literals that could be matched with entities.

We have developed a framework that can automatically *canonicalize* such literals, i.e., replace the literal with an existing entity from the KG or with a new entity that is typed using classes from the KG. Our framework combines both reasoning and machine learning in order to predict the relevant entities and types, and has been shown to perform well in comparison to state-of-the-art baselines for both semantic typing and entity matching [2].

2.2. Correcting Erroneous Assertions

Another issue that is prevalent in KGs such as Wikidata and DBpedia is the presence of erroneous assertions [9]. These may be due to the knowledge source (e.g., Wikipedia is estimated to have an error rate of 2.8% [13]), or may be introduced by knowledge extraction processes. There has already been quite a lot of work on detecting such errors, e.g., via statistical methods [10], but relatively little work on correcting them (it is typically assumed that erroneous assertions are simply discarded).

We have developed a a general framework for correcting assertions whose objects are either erroneous entities or literals. In the latter case we can exploit our work on canonicalization; in the former case we use a combination of related entity estimation, link prediction and constraint-based validation [1].

2.3. Identifying Conceptual Modelling Errors

Ontology alignment is a well known problem that has been widely studied; see, e.g., `http://oaei.ontologymatching.org/`. This usually involves identifying a set of *mappings* between two ontologies, where each mapping is a semantic relationship between a pair of concepts, one from each of the ontologies. Mappings are typically equivalence or subclass relationships, and are often rendered as axioms; this has the advantage that we can merge the two ontologies simply by combining them along with the mappings in a single merged ontology.

As well as merging ontologies, the alignment of two ontologies covering similar domains can be used for quality assurance; effectively, we compare the domain knowledge captured in the two ontologies and identify apparent disagreements. Our initial experiments using the LogMap tool[2] with FoodOn[3] and HeLiS[4] suggest that this is a useful technique for identifying errors of both omission and duplication. To aid intuitive understanding, in the following we will render class names as *ont:label*, where *ont* indicates the ontology (FoodOn or HeLiS in this case) and *label* is the label string in camel case; e.g., we will refer to the FoodOn class *FOODON:03305289* (labelled “soybean milk”) as *foodon:SoybeanMilk*.

The basic idea is to align the two ontologies, and to examine new subsumption relationships in the resulting aligned ontology. In particular, we focus on subsumptions between pairs of classes in one of the ontologies that hold only after alignment with the other ontology — this suggests some disagreement in the way the two ontologies model the domain. For example, when we align FoodOn with HeLiS, we find a large number of alignments, including *helis:SoyProducts* with *foodon:SoybeanFoodProduct* and *helis:SoyMilk* with *foodon:SoybeanMilk*. As a result, *foodon:SoybeanMilk* (implicitly) becomes a subClass of *foodon:SoybeanFoodProduct* in the merged ontology, a subsumption relationship that does not hold in FoodOn alone. Such additional relationships may indicate errors in the ontology, and indeed this does seem to be an error of omission in FoodOn, and one that could have potentially serious consequences for those with soy allergy if the ontology were used for dietary advice.

This process identified more than 500 new subsumption relationships between FoodOn classes. We did not have the time or domain expertise to examine all of these, but several of those that we did examine seemed to identify possible errors, or at least issues that are worth investigation. For example, *foodon:Mushroom* is a subclass of *foodon:MushroomVegetableFoodProduct* in the aligned ontology, but not in FoodOn alone. If we investigate this further we can see that foodon classifies *foodon:Mushroom* as a kind of *foodon:FoodSource*, but it classifies specific kinds of mushroom (such as *foodon:BoletusMushroom*) under *foodon:MushroomVegetableFoodProduct*.

Other new subsumption relationships were due to the fact that FoodOn includes two separate models of vitamins that are not at all integrated, with duplicates such as

[2] `http://www.cs.ox.ac.uk/isg/tools/LogMap/`

[3] `http://purl.obolibrary.org/obo/foodon/releases/2020-01-29/foodon.owl`

[4] `https://horus-ai.fbk.eu/helis/`

foodon:BVitamin and *foodon:VitaminB*, and *foodon:vitamin* and *foodon:Vitamin* (the former with a lower case "v").

3. Future Work

So far we have only been exploring various techniques, and haven't attempted to build user-friendly tools. The experiment with FoodOn, in particular, involved several manual steps using LogMap, Protégé, and various Linux command line tools. However, the potential seems to be clear, and to justify some effort in developing a fully-fledged tool, perhaps as a Protégé plugin.

References

[1] J. Chen, X. Chen, I. Horrocks, E. B. Myklebust, and E. Jiménez-Ruiz. Correcting knowledge base assertions. In *WWW*, pages 1537–1547. ACM / IW3C2, 2020.

[2] J. Chen, E. Jiménez-Ruiz, and I. Horrocks. Canonicalizing knowledge base literals. In *Proc. of the 18th International Semantic Web Conference (ISWC 2019)*, volume 11778 of *Lecture Notes in Computer Science*, pages 110–127. Springer, 2019.

[3] B. Cuenca Grau, I. Horrocks, Y. Kazakov, and U. Sattler. Modular reuse of ontologies: Theory and practice. *J. of Artificial Intelligence Research*, 31:273–318, 2008.

[4] J. Euzenat, C. Meilicke, H. Stuckenschmidt, P. Shvaiko, and C. T. dos Santos. Ontology alignment evaluation initiative: Six years of experience. *J. of Data Semantics*, 15:158–192, 2011.

[5] K. Evers, T. Liebig, A. Maisenbacher, M. Opitz, J. R. Seyler, G. Sudra, and J. Wissmann. An overview of the festo semantic platform. In *ESWC (Satellite Events)*, volume 11762 of *Lecture Notes in Computer Science*, pages 43–46. Springer, 2019.

[6] K. Gunaratna, K. Thirunarayan, A. Sheth, and G. Cheng. Gleaning types for literals in rdf triples with application to entity summarization. In *European Semantic Web Conference*, pages 85–100, 2016.

[7] M. A. Musen. The protégé project: a look back and a look forward. *AI Matters*, 1(4):4–12, 2015.

[8] C. Ochs, J. Geller, Y. Perl, and M. A. Musen. A unified software framework for deriving, visualizing, and exploring abstraction networks for ontologies. *J. Biomed. Informatics*, 62:90–105, 2016.

[9] H. Paulheim. Knowledge graph refinement: A survey of approaches and evaluation methods. *Semantic web*, 8(3):489–508, 2017.

[10] H. Paulheim and C. Bizer. Improving the quality of linked data using statistical distributions. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 10(2):63–86, 2014.

[11] H. Paulheim and A. Gangemi. Serving dbpedia with DOLCE - more than just adding a cherry on top. In *International Semantic Web Conference (1)*, volume 9366 of *Lecture Notes in Computer Science*, pages 180–196. Springer, 2015.

[12] M. G. Skjæveland, D. P. Lupp, L. H. Karlsen, and H. Forssell. Practical ontology pattern instantiation, discovery, and maintenance with reasonable ontology templates. In *International Semantic Web Conference (1)*, volume 11136 of *Lecture Notes in Computer Science*, pages 477–494. Springer, 2018.

[13] G. Weaver, B. Strickland, and G. Crane. Quantifying the accuracy of relational statements in wikipedia: A methodology. In *Proceedings of the 6th ACM/IEEE-CS Joint Conference on Digital Libraries*, volume 6, pages 358–358. Citeseer, 2006.