The Manchester OWL Repository: System Description

Nicolas Matentzoglu, Daniel Tang, Bijan Parsia, and Uli Sattler

The University of Manchester
Oxford Road, Manchester, M13 9PL, UK
{matentzn,bparsia,sattler}@cs.manchester.ac.uk

Abstract. Tool development for and empirical experimentation in OWL ontology research require a wide variety of suitable ontologies as input for testing and evaluation purposes and detailed characterisations of real ontologies. Findings of surveys and results of benchmarking activities may be biased, even heavily, towards manually assembled sets of "somehow suitable" ontologies. We are building the Manchester OWL Repository, a resource for creating and sharing ontology datasets, to push the quality frontier of empirical ontology research and provide access to a great variety of well curated ontologies.

Keywords: Repository, Ontologies, Empirical

1 Introduction

Empirical work with ontologies comes in a wide variety of forms, for example surveys of the modular structure of ontologies[1], surveys of modelling patterns to inform design decisions of engineering environments [4] and benchmarking activities for reasoning services such as Description Logic (DL) classification [2]. Since it is generally difficult to obtain *representative* datasets, both due to technical reasons (lack of suitable collections) and conceptual reasons (lack of agreement on what they should be representative of), it is common practice to manually select a somewhat arbitrary set of ontologies that usually supports the given case. On top of that, few authors ever publish the datasets they used, often for practical reasons (e.g. size, effort), which makes reproducing experiment results often impossible. The currently best option for ontology related research is the BioPortal repository [5], which provides a web based interface for browsing ontologies in the biomedical domain and a REST web service to programmatically obtain copies of all (public) versions of a wide range of biomedical ontologies. There are, however, certain problems with this option. First, the repository is limited to biomedical ontologies, which makes BioPortal unsuitable for surveys that require access to ontologies of different domains. The second problem is the technical barrier of accessing the web service: It requires a good amount of work to download all interesting ontologies, for example due to a range of ontologies published in a compressed form or the logistical hurdle of recreating new snapshots over and over again. The third problem is due to the fact that there is

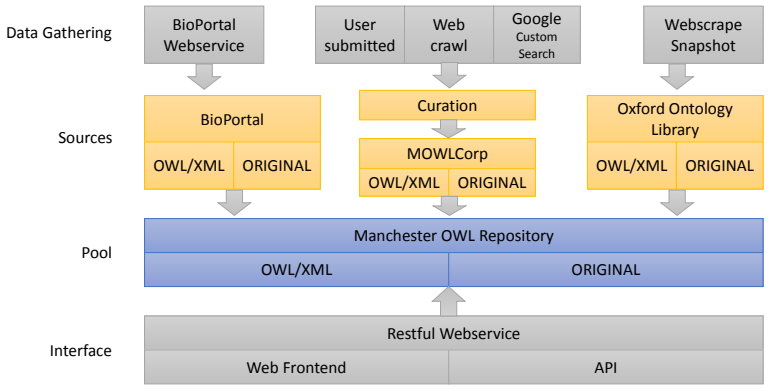

Fig. 1. The repository architecture.

no shared understanding of what it means to "use BioPortal". Different authors have different inclusion and exclusion criteria, for example they only take the ones that are easily parseable after download, or the ones that were accessible at a particular point in time. The Manchester OWL Respository aims to bridge that gap by providing a framework for conveniently retrieving some standard datasets and allowing users to create, and share, their own.

2 Overall architecture

The Manchester OWL repository can be divided into four layers (see Figure 1). The first layer represents the data gathering. Through web crawls, web scrapes, API calls, and user contributions ontologies are collected and stored in their respective collections. The second layer represents the three main data sources of the repository, each providing ontologies in their original and curated (OWL/XML) form. The third layer, the pool, represents a virtual layer in which access to the ontologies is unified, providing some means for de-duplification because of the possibility of corpora intersection. Lastly, the interface layer provides access to the repository through a REST service and a web-based front end.

3 Data Gathering

The main component of the data gathering layer is a web crawl based on crawler4j, a java-based framework for custom web crawling and daily calls to the Google Custom Search API that fills the MOWLCorp, which makes up the bulk of the repository's data. An ongoing BioPortal downloader creates a snapshot of BioPortal once per month using the BioPortal web services, whilst retaining copies of all available versions so far. The third (minor) component of the repository is a web scrape of the Oxford Ontology Library (OOL), a hand curated set of ontologies which features some particularly difficult, and thus in-

teresting to reasoner developers, ontologies. Ontologies are downloaded in their raw form and thrown in the curation pipeline.

4 Data curation

Ontology candidates from all three sources undergo a mild form of repair (undeclared entity injection, rewrite of non absolute IRIs) and are exported into OWL/XML, with their imports closure merged into a single ontology, while retaining information about the axiom source ontology through respective annotations. Metrics and files for both the original and the curated versions of the ontologies are retained and form part of the repository. The data curation looks slightly different for all three data sources, especially with respect to filtering. Apart from the criterion of OWL API [3] parse-ability, BioPortal and the OOL are left unfiltered because they are already deemed curated. This means that some ontologies in the corpus may not contain any logical axioms at all. In MOWLCorp, on the other hand, we filter out ontologies that 1) have an empty TBox (root ontology) and 2) have byte-identical duplicates after serialisation into OWL/XML. The reason for the first step is our focus on ontologies (which excludes pure collections of RDF instance data) and the fact that the imports closure is part of the repository, i.e., they are downloaded and evaluated independently of the root ontology.

5 Accessing the repository

There are currently three different means to access the repository: 1) A web frontend[1] provides access to preconstructed datasets and their descriptions, 2) an experimental data set creator allows users to create custom datasets based on a wide range of metrics and 3) an experimental REST-based web service that allows users to create a dataset using the REST API. Since 2) is based on 3), we now describe the query language that allows users to create their own datasets and access the web service.

The query language allows the user to construct statements that represent filter criteria for ontologies based on some essential metrics such as axiom and entity counts, or profile membership. It roughly conforms to the following grammar:

$$\begin{aligned} \text{q} &= \text{comp } \{(\text{"\&\&"}|\text{"||"})\ \text{comp}\} \\ \text{comp} &= \text{metric } (\text{">="} \mid \text{"<="} \mid \text{"="})\ n \\ \text{metric} &= \text{"axiom\_count"} \mid \text{"class\_count"} \mid ... \end{aligned}$$

where "metric" should be a valid metadata element. The query language parser was built with open-source parser generator Yacc and Lex.

The repository web services are built using the PHP framework Laravel. Laravel is an advanced framework which implements the REST protocol, so that users can get access to the services using a REST client, or simply using a web

[1] `http://mowlrepo.cs.manchester.ac.uk/`

Table 1. The REST service parameters.

service	url	method	param	return
query	/api/	POST	query	JSON array with fields:status, count,size, message, progress
check status	/api/checkStatus/	GET	id	JSON array with fields: status, progress
download	/api/resource	GET	id	file stream

browser and web-based tools such as Curl. For now, we have implemented three services: query, checkStatus and download. The query service accepts a query string that complies to the query language and returns an id string. Afterwards, users can use the id string to check the status of their query, and to download the final dataset using checkStatus and download services.

The usage of the services are listed in the Table 1; note that urls should be appended to mowlrepo.cs.manchester.ac.uk which has been omitted.

6 Next steps

We have presented the Manchester OWL Repository and a range of prototype interfaces to access pre-constructed datasets and create custom ones. We believe that the repository will help pushing the quality frontier of empirical ontology-related research by providing access to shareable, well curated datasets. We are currently working on the REST services, the dataset creator and improved dataset descriptions. In the near future, we are aiming to 1) integrate the repository with Zenodo, a service that allows hosting large datasets that are citable via DOIs, 2) extend our metadata to capture even more ontology properties (in particular consistency and coherence) and 3) improving the curation pipeline by implementing extended yet save fixes for OWL DL profile violations.

References

1. C. Del Vescovo, P. Klinov, B. Parsia, U. Sattler, T. Schneider, and D. Tsarkov. Empirical study of logic-based modules: Cheap is cheerful. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 8218 LNCS, pages 84–100, 2013.
2. R. S. Gonçalves, S. Bail, E. Jiménez-Ruiz, N. Matentzoglu, B. Parsia, B. Glimm, and Y. Kazakov. OWL Reasoner Evaluation (ORE) Workshop 2013 Results: Short Report. In *ORE*, pages 1–18, 2013.
3. M. Horridge and S. Bechhofer. The OWL API: A Java API for OWL ontologies. *Semantic Web*, 2:11–21, 2011.
4. M. Horridge, T. Tudorache, J. Vendetti, C. Nyulas, M. A. Musen, and N. F. Noy. Simplified OWL Ontology Editing for the Web: Is WebProt{é}g{é} Enough? In *International Semantic Web Conference (1)*, pages 200–215, 2013.
5. N. F. Noy, N. H. Shah, P. L. Whetzel, B. Dai, M. Dorf, N. Griffith, C. Jonquet, D. L. Rubin, M. A. Storey, C. G. Chute, and M. A. Musen. BioPortal: Ontologies and integrated data resources at the click of a mouse. *Nucleic Acids Research*, 37, 2009.