Accessing Information About Linked Data Vocabularies with vocab.cc

Steffen Stadtmüller, Andreas Harth, and Marko Grobelnik

Abstract Linked Data vocabulary designers and application developers need means to easily identify relevant vocabularies, to allow them to reuse existing vocabularies and to develop applications making use of Linked Data. We describe a system that provides information about the popularity of classes and properties based on the Billion Triple Challenge data set. The information about classes and properties can be accessed via a web portal or via Linked API resources. We describe both the data analysis process and the architecture of the web portal.

1 Introduction

Providing data in a machine understandable manner—for example, as Linked Data—significantly improves access and integration of such data. Vocabularies provide schema information for Linked Data, i.e., allowing to talk about classes and properties that are used to describe instances. The fourth Linked Data principle[1] implies the reuse of existing vocabulary URIs. Reusing existing URIs improves the interlinkage of hitherto disparate pieces of data. Thus, data publishers should reuse existing vocabulary URIs, rather than minting new URIs, if possible and appropriate [1,4]. However, it is currently not easy to find out which domain existing

[1] http://www.w3.org/DesignIssues/LinkedData.html.

S. Stadtmüller (✉) • A. Harth
Institute AIFB, Karlsruhe Institute of Technology (KIT), Germany
e-mail: steffen.stadtmueller@kit.edu; andreas.harth@kit.edu

M. Grobelnik
Jožef Stefan Institute, Slovenia
e-mail: marko.grobelnik@ijs.si

J. Li et al. (eds.), *Semantic Web and Web Science*, Springer Proceedings in Complexity,
DOI 10.1007/978-1-4614-6880-6_34,

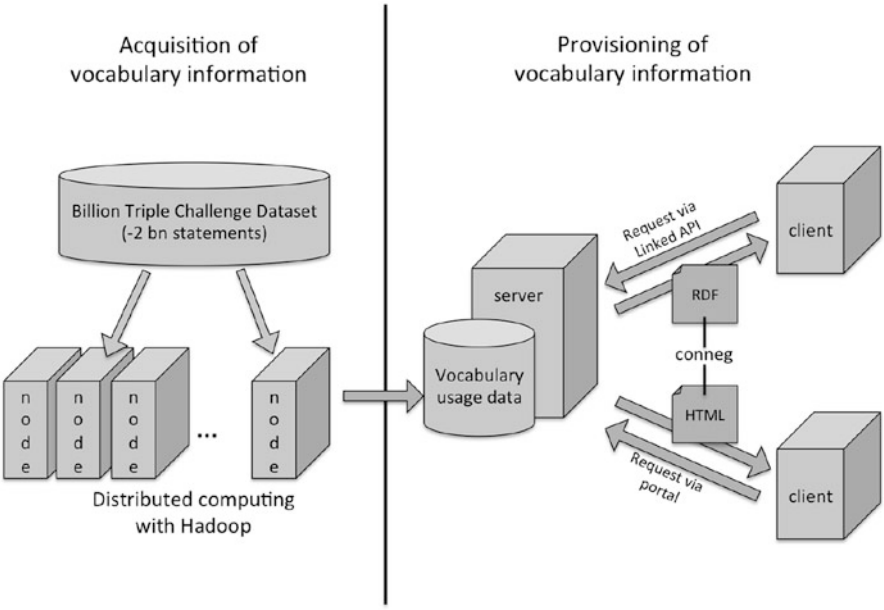

Fig. 1 Architecture overview

vocabularies cover or how relevant existing vocabularies are. Therefore, we see the need for a system that enables data publishers to swiftly acquire information about already available vocabularies and their relevance.

We devise a web portal, called *vocab.cc*,[2] where data publishers and developers can access information about popular classes and properties.

Rather than requiring manual effort, our notion of popularity stems from an analysis of a crawled data set. In other words, *vocab.cc* focuses on the ex post acquisition and provision of information about already existing ontologies. Information about the real use of class URIs and property URIs in the web of data provides an indicator for the relevance of a specific vocabulary URI. In addition to a web interface targeted at human users, *vocab.cc* offers additionally a Linked API, which allows for an easy integration of the data into other applications. The code of *vocab.cc* is available as open source.[3]

Our demonstration will show how to acquire useful information from a large Linked Data corpus and how the information acquired for *vocab.cc* can be accessed via the portal and via the Linked API. Figure 1 illustrates an overview of the architecture. We describe related approaches in Chap. 2. In Chap. 3 we describe how we acquire the necessary statistical information. Next, we explain how the information can be accessed in Chap. 4. Finally, we conclude with Chap. 5.

[2] http://vocab.cc/.

[3] http://code.google.com/p/vocab/.

2 Related Work

LODStats offers information according to 32 statistical criteria about data sets published in the CKAN repository.[4] To do so it accesses dump files and the SPARQL end points of the registered data sets. In its current release, the tool covers 226 data sets with a volume of 1, 211, 878, 106 triples. The analysis of the underlying ontologies covers 14, 433 unique vocabulary elements. *SchemaCache*[5] and *Linked Open Vocabularies (LOV)*[6] operate as registries for ontologies used by Linked Data publishers. The documentation of these ontologies is provided by the developers or submitted by (registered) users. *Schema-Cache* covers 9, 489 unique vocabulary elements. With a more limited coverage (3,714 vocabulary elements), *LOV* focuses on the classification of vocabularies and the provisioning of detailed metadata information about them.

Cupboard is an approach to support ontology engineers to publish ontologies in a way that users can assess and reuse ontologies [2].

vocab.cc offers with 261,119 unique vocabulary elements a significantly larger coverage of existing vocabularies than previous approaches.

3 Analysis of Existing Linked Data Vocabularies

Our demonstration will provide an introduction in the methods used to extract information from a large data set.

As basis for our analysis, we use the Billion Triple Challenge 2011 data set,[7] which contains over 2.1 bn statements in N-Quads[8] format, collected from 7.4 m documents. We extract all URIs from the BTCD that are used as predicates (a total of 47,681) and all URIs that represent a class (a total of 213,438), thus covering 261,119 unique vocabulary elements. URIs are identified as classes if they are in object position in a triple with *rdf:type* as predicate.

Considering the size of the corpus, we use Apache Hadoop[9] to analyse the data. Hadoop allows for the parallel and distributed processing of large data sets across clusters of computers. We run the analysis on the KIT OpenCirrus[10] Hadoop cluster. OpenCirrus is a collaboration of several organizations to provide an open cloud-computing research test bed designed to support research. For our analysis we used

[4] http://stats.lod2.eu/.

[5] http://schemacache.com/.

[6] http://labs.mondeca.com/dataset/lov/index.html.

[7] http://km.aifb.kit.edu/projects/btc-2011/.

[8] http://sw.deri.org/2008/07/n-quads/.

[9] http://hadoop.apache.org/.

[10] https://opencirrus.org/.

Table 1 Top vocabulary URIs by overall occurrence

(a) Top 10 classes			(b) Top 10 properties		
#	URI	Overall frequency	#	URI	Overall frequency
1	foaf:Person	365 623 021	1	rdf:type	579 095 292
2	cube:Observation	6 783 306	2	rdfs:seeAlso	369 286 912
3	rdf:Statement	5 767 380	3	foaf:nick	366 167 925
4	mo:MusicArtist	3 979 450	4	foaf:knows	365 522 760
5	cc:Work	3 055 547	5	rdfs:label	25 755 421
6	foaf:OnlineAccount	2 930 600	6	foaf:weblog	21 814 705
7	foaf:PersonalProfileDocument	2 593 101	7	foaf:member_name	19 146 708
8	foaf:Agent	2 535 723	8	foaf:tagLine	19 146 699
9	owl:Class	2 096 025	9	foaf:image	18 133 652
10	swrc:Person	1 850 559	10	owl:sameAs	8 552 727

Table 2 Top vocabulary URIs by count of documents

(a) Top 10 classes			(b) Top 10 properties		
#	URI	Document frequency	#	URI	Document frequency
1	foaf:Person	1 633 434	1	rdf:type	6 694 991
2	foaf:Document	814 800	2	rdfs:label	2 867 107
3	freebase:common.topic	572 382	3	rdfs:seeAlso	2 381 790
4	owl:Thing	468 387	4	foaf:primaryTopic	2 099 555
5	mo:MusicArtist	346 728	5	owl:sameAs	1 778 210
6	dc:IMT	330 971	6	foaf:weblog	1 590 806
7	frbr:Manifestation	330 946	7	foaf:nick	1 496 280
8	frbr:Expression	330 943	8	foaf:knows	1 469 700
9	metalex:BibliographicManifestation	330 943	9	foaf:img	1 341 111
10	metalex:BibliographicExpression	330 943	10	foaf:page	1 194 188

54 work nodes, each with a 2.27 GHz 4-Core CPU and 100GB RAM, a setup which completes a scan over the entire corpus in about 15 min.

During a scan over the data, for each identified class URI and property URI, we derive two frequency measures (results in Tables 1 and 2):

- We count how often each identified class and property is used in the BTCD overall. An overall count regards classes and properties more important even if they appear often in just a few large documents.
- We also count for every identified URI how many of the original data sources (i.e., documents) make use of the URI. A count per document regards vocabularies more important that are used by many different documents, even if they are small.

Furthermore we extract all labels of class URI and property URI to allow for keyword search functionality. We also extract the local names of identified URIs and add them to the set of labels. A web application, described next, provides the statistics and the keyword search functionality to users.

4 Access to the Information

We provide access to the data derived from the BTC corpus via *vocab.cc*. The portal provides a minimal interface with a central input field, where users can specify a URI or type in a keyword query. Users can input URIs with their common namespace rather than their fully qualified name. *vocab.cc* makes use of *prefix.cc*[11] which also inspired name and layout of the web portal. Figures 2 and 3 show the portal and how results are represented.

Users can define an arbitrary query (i.e., a string of words) for their domain of interest to search for existing vocabularies. *vocab.cc* matches the words in a query with the labels found for the URIs. The response details the classes and properties, which labels contain all of the specified words. Words in the query are disregarded, if they do not appear in any label, thus increasing the number of potential result sets.

vocab.cc also allows users to specify a URI directly. Returned information includes the number of overall appearances in the BTC data set as well as the number of documents the URI appeared in. Additionally *vocab.cc* returns the positions in the rankings.

A Linked API allows access to the information, beyond the human readable way to access *vocab.cc*. The Linked API allows for an easy integration of the functionalities in other applications, fostering the Linked Data principles. Linked

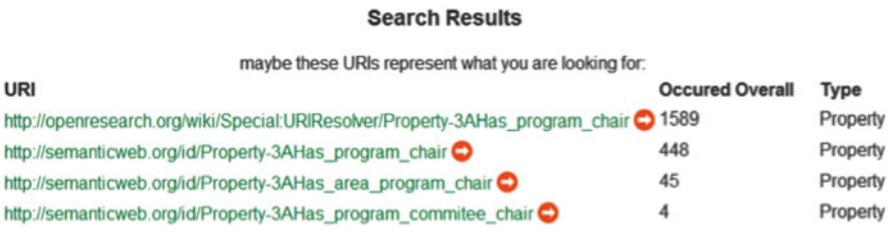

Fig. 2 Query results

Fig. 3 Usage information for a URI

[11] http://prefix.cc/.

APIS [5, 6] offer web service functionalities as RDF prosumers by combining LD technologies with RESTful services [3].

The demonstration will illustrate the different methods to make use of *vocab.cc*.

The resources of the Linked API allow to submit queries to *vocab.cc* in an HTTP POST request. The HTTP response contains RDF data, detailing the usage information of the found URIs. Accessing the output RDF as resources is also possible directly via content negotiation: A Client can perform an HTTP GET on the corresponding URI of the portal asking for an RDF content type. This direct access adheres to a RESTful architecture style.

5 Conclusion and Outlook

vocab.cc provides the means to search for RDF vocabularies based on labels and URIs and decides on the relevance of the vocabularies based on usage information.

To improve the *vocab.cc*, accounting for subclass and subproperty hierarchies could lead to a refined definition of the popularity of a URI. Furthermore, aggregating the usage information can lead to an understanding of the relevance of a vocabulary itself, rather than just of the individual classes and properties. Possible synergies can be achieved by linking *vocab.cc* with other vocabulary catalogues. Finally, we intend to allow users to contribute data about vocabularies.

Acknowledgements The research leading to this paper was partially supported by the Network of Excellence PlanetData,[12] funded by the European Community's Seventh Framework Programme FP7/2007-2013 under contract 257641.

References

1. Bizer, C., Heath, T., Berners-Lee, T.: Linked data - the story so far. Int. J. Semantic Web Inf. Syst. **5**(3), 122 (2009)
2. d'Aquin, M., Lewen, H.: Cupboard - a place to expose your ontologies to applications and the community. In: ESWC. LNCS, vol. 5554, pp. 913–918. Springer, New York (2009)
3. Fielding, R.: Architectural styles and the design of network-based software architectures. Ph.D. thesis, University of California, Irvine (2000)
4. Heath, T., Bizer, C.: Linked Data: Evolving the Web into a Global Data Space, 1st edn. Synthesis Lectures on the Semantic Web, Morgan & Claypool (2011)
5. Krummenacher, R., Norton, B., Marte, A.: Towards linked open services. In: 3rd Future Internet Symposium, September 2010
6. Speiser, S., Harth, A.: Integrating linked data and services with linked data services. In: Proceedings of 8th Extended Semantic Web Conference, ESWC 2011. pp. 170–184 (2011)

[12] http://planet-data.eu/.