Linked Open Data Statistics: Collection and Exploitation

Ivan Ermilov[1], Michael Martin[1], Jens Lehmann[1], Sören Auer[2]

[1] AKSW/BIS, Universität Leipzig, Germany, http://aksw.org
{lastname}@informatik.uni-leipzig.de
[2] CS/EIS, Universität Bonn, http://www.iai.uni-bonn.de/~auer
auer@cs.uni-bonn.de

Abstract. This demo presents LODStats, a web application for collection and exploration of the Linked Open Data statistics. LODStats consists of two parts: *the core* collects statistics about the LOD cloud and publishes it on the LODStats web portal, a *front-end* for exploration of dataset statistics. Statistics are published both in human-readable and machine-readable formats, thus allowing consumption of the data through web front-end by the users as well as through an API by services and applications. As an example for the latter we showcase how to visualize the statistical data with the CubeViz application.

1 Introduction

For assessing the state of the Web of Data, for evaluating the quality of individual datasets as well as for tracking the progress of Web data publishing and integration, it is of paramount importance to gather comprehensive statistics on datasets describing their internal structure and external cohesion. We even deem the difficulty to obtain a clear picture of the available datasets to be a major obstacle for a wider usage of Data Web and the deployment of semantic technologies. In order to reuse, link, revise or query a dataset published on the Web, for example, it is important to know the structure, coverage and coherence of the data.

In this demo, we showcase a web application for collection and exploration of the Linked Open Data statistics. The web application is based on *LODStats* – a statement-stream-based approach for gathering comprehensive statistics from resources adhering to the Resource Description Framework (RDF) [4]. One rationale for the development of LODStats is the computation of statistics for resources from the *Comprehensive Knowledge Archive* (CKAN, "The Data Hub"[3]) on a regular basis. Data catalogs such as CKAN enable organizations to upload or link and describe data sources using comprehensive meta-data schemes. Similar to digital libraries, networks of such data catalogs can support the description, archiving and discovery of data on the Data Web. Recently, we have seen a rapid

[3] http://thedatahub.org

growth of data catalogs being made available on the Web. The data catalog registry *datacatalogs.org*, for example, already lists 336 data catalogs worldwide. Examples for the increasing popularity of data catalogs are Open Government Data portals (which often include data sources about energy networks, public transport, environment etc.), data portals of international organizations and NGOs, scientific data portals as well as master data catalogs in large enterprises.

Datasets from CKAN are available either serialised as a file (in RDF/XML, N-Triples and other formats) or/and via SPARQL endpoints. Serialised datasets containing more than a few million triples (i.e. data items) tend to be too large for most existing analysis approaches as the size of the dataset or its representation as a graph exceeds the available main memory, where the complete dataset is commonly stored for statistical processing. LODStats' main advantage when compared to existing approaches is that the tradeoff between performance and accuracy can be controlled, in particular it can be adjusted to the available main memory size. Furthermore, LODStats is also easily extensible with novel analytical criteria. It comes with a set of 32 different statistics, amongst others are those covering the statistical criteria defined by the *Vocabulary of Interlinked Datasets* [1] (VoID). Examples of available statistics are property usage, vocabulary usage, datatypes used and average length of string literals. LODStats implementation is written in *Python* and available as a module for integration with other projects.

This paper is organized as follows. In Section 2 we describe the architecture of the developed web platform. We give an overview on the gathered statistics in Section 3. We describe how we exposed the statistics in RDF format using the Sparqlify RDB2RDF conversion tool in Section 4. Finally, in Section 5 we show how to explore the collected statistics with the example of the CubeViz application. We conclude our work in Section 6.

2 LODStats Architecture

Figure 1 shows an overview of the general architecture of LODStats. LODStats consists of two parts: *LODStats Core* collects statistics about LOD cloud and publishes it on the LODStats web portal, the *front-end* for the data exploration.

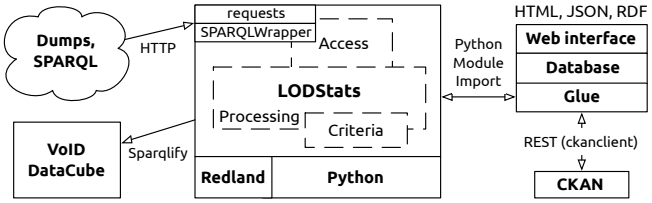

Fig. 1. Architecture of the LODStats reference implementation.

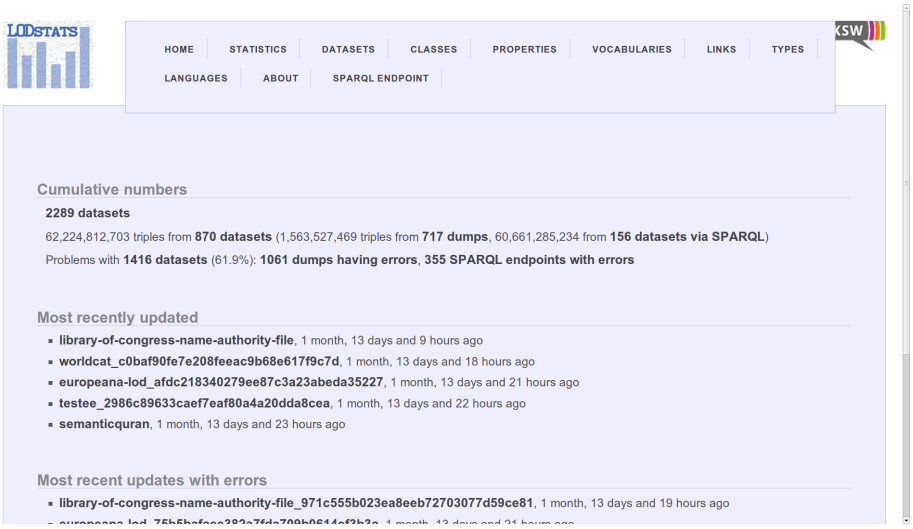

Fig. 2. LODStats front-end at `http://stats.lod2.eu`. General overview.

LODStats Core. LODStats Core is written in Python and uses the Redland library [3] and its bindings to Python [2] to parse different RDF serializations and process them statement by statement. Resources reachable via HTTP, contained in archives (e.g. zip, tar) or compressed with gzip or bzip2 are transparently handled by LODStats. *SPARQLWrapper* [5] is used for augmenting LODStats with support for SPARQL endpoints.

LODStats core has been implemented as a Python module with simple calling conventions, aiming at general reusability. It is available for integration with the Comprehensive Knowledge Archiving Network (CKAN), a widely used dataset metadata repository, either as a patch or as an external web application using CKAN's API.[4] Integration with other (non Python) projects is also possible via a command line interface and a RESTful web service.

LODStats front-end. We implemented a web interface[5] (cf. Figure 2, Figure 3), which provides continuously updated information about RDF datasets from the Data Hub (CKAN).[6] Beyond the various statistics made, the interface allows to search for common classes or properties, helping to encourage vocabulary reuse. Class and property search functionality as well as the statistics are also available via a RESTful web service for integration with other projects. The interface supports the functionality listed below.

– General LOD cloud statistics,

[4] `https://github.com/AKSW/LODStats`
[5] `http://stats.lod2.eu`
[6] Our implementation is open-source and is accessible at `https://github.com/AKSW/LODStats_WWW`

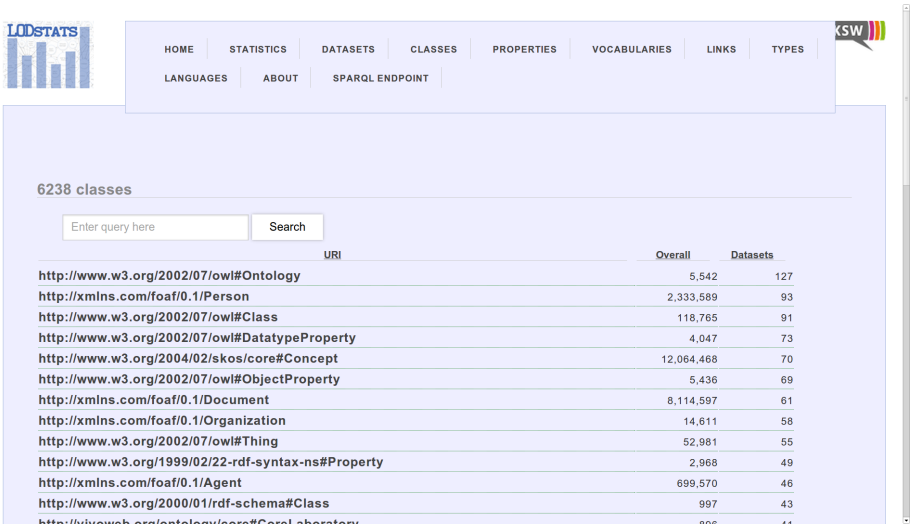

Fig. 3. LODStats front-end at `http://stats.lod2.eu`. Classes usage statistics.

- report of warnings and errors for each dataset,
- report on statistical criteria for each individual dataset,
- export as VoID and DataCube statistical metadata,
- dataset linkage explorer,
- search function for datasets, vocabularies, classes, properties, languages, datatypes,
- REST interface for the above search functions,
- Linked Data publication of statistics,
- SPARQL endpoint to query all extracted statistics,
- CubeViz installation for facet-based browsing and visualisation of the statistical metadata.

The service permanently regularly crawls data portals such as TheDataHub.org for new and updated datasets as well as (re-)computes the respective statistics.

3 LOD Statistics Overview

We evaluated every dataset available via CKAN in one of the formats LODStats can handle (i.e. N-Triples, RDF/XML, Turtle, N-Quads, N3). We excluded datasets for which we know that they duplicate fractions of already analysed data such as the LOD cloud cache[7]. Results (see Table 1) show that a significant amount of datasets is solely available via SPARQL endpoints (22.3%), with 23.7% of those having errors. Problems most likely originate from either limitations set up on queries to mitigate abuse of endpoints that made gathering

[7] `http://lod.openlinksw.com/sparql`

	2011-12-09	2012-04-25	2013-03-01	2013-05-15
Datasets	452	506	699	2289
SPARQL-only	198	215	200	511
Triples	950M	1,174M	7,4B	11B
Dumps	235M	534M	1,3B	1.5B
SPARQL	714M	640M	6,1B	9.5B
Errors	248	276	366	592
Unreachable	45	38	121	374
SPARQL issues	139	153	150	121
Parse errors	57	66	94	91
Archive issues	3	2	1	6
Warnings	2334	5029	3801	5870

Table 1. Aggregated LODStats results at different points in time.

statistics infeasible or from not supporting necessary extensions to SPARQL for counting triples on the endpoint side. Complete description and analysis of gathered statistics is described in [4].

4 Exposing Statistics as a Linked Data

Originally LODStats statistics are stored in a PostgreSQL relational database. In addition we publish the collected statistics as RDF data through a SPARQL endpoint[8]. In this section we describe how to establish a mapping to connect the relational database with the RDF DataCube vocabulary using the RDB2RDF mapping tool Sparqlify [6].

Mappings for Sparqlify are written using Sparqlification Mapping Language (SML). In general[9], a mapping comprises one or several views, each one consisting of *Construct*, *With* and *From* parts. *From* part specifies a SQL query for a relational database. *With* part binds Sparqlify variables to the query result. *Construct* part utilizes Sparqlify variables to build RDF triples. The RDF data is obtained row-wise.

To represent the LODStats statistics in RDF we choose the W3C RDF Data Cube Vocabulary[10]. For the data transformation from the relational database to the RDF Data Cube we define a mapping comprised of two views: `Static` and `StatResults`.

- **Static view** describes the static elements of the RDF Data Cube Model: `qb:DataSet`, `qb:DataStructureDefinition` and several `qb:component` elements. There are five `qb:component` elements in `qb:DataStructureDefinition` with one associated `qb:Dataset`. Three `qb:dimension` elements (time of

[8] `http://stats.lod2.eu/sparql`

[9] Detailed description of the SML as well as the examples are located at the Sparqlify wiki `http://sparqlify.org/wiki/Sparqlify_mapping_language`.

[10] `http://www.w3.org/TR/vocab-data-cube/`

measure, statistical criterion, source dataset) explicitly identify any particular observation. Our data model includes exactly one `qb:measure` and one `qb:attribute` element.

– **StatResults view** operates on the statistical data from the relational database of LODStats application. It retrieves available statistical criteria from the LODStats database and constructs an observation for each. Observation URIs are defined as a concatenation of a *ls-qb* namespace URI with a unique hash.

An overview of the mapping process is depicted in Figure 4. The complete listing of the mapping is available at LODStats GitHub repository.[11]

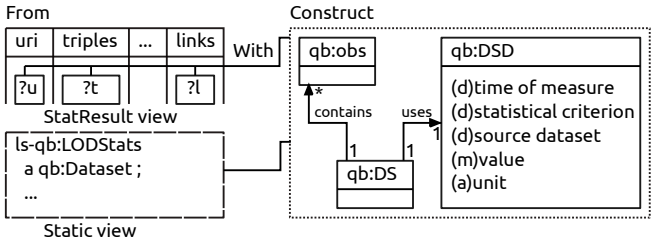

Fig. 4. Overview of an SML mapping. Abbreviations used In the figure: qb:obs - `qb:observation`, qb:DS - `qb:DataSet`, qb:DSD - `qb:DataStructureDefinition`. Inside qb:DSD: (d) stands for dimension, (m) for measure and (a) for attribute.

5 Visualizing Statistics

Well-formed RDF Data Cubes can be visualized as is by existing applications, thus providing the new insights on the data without additional effort. In this section we demonstrate the visualization of the LODStats statistics with *CubeViz*[12].

CubeViz visualizes RDF Data Cube datasets with different chart types. By defining the SPARQL endpoint inside the application, it is possible to identify the statistical data structured according to Data Cube RDF inside this endpoint. A faceted browser allows to choose from available dimensions and measures and thus select the data. The selected components and retrieved data are displayed as charts as well as plain text in the selected configuration and retrieved data tabs.

In Figure 5 we choose the LODStats SPARQL endpoint and compare the number of triples, entities, literals and links (statistical criterion dimension) in

[11] `https://github.com/AKSW/LODStats_WWW/blob/master/LODStats-Sparqlify/lodstats.sml`
[12] `http://cubeviz.aksw.org/`

HeBIS[13], Library of Congress NAF[14] and World Bank Linked Data[15] datasets (source dataset dimension) in May 2013 (time of measure dimension). The visualization is represented as a column chart. The user can easily adapt the visualization to her needs by changing visualization parameters. For instance, changing column chart to bar chart or swapping axes. The visualization parameters are configured in the configure visualization window (under a chart type icon). The online demo of this example is available at `http://cubeviz.aksw.org/example`.

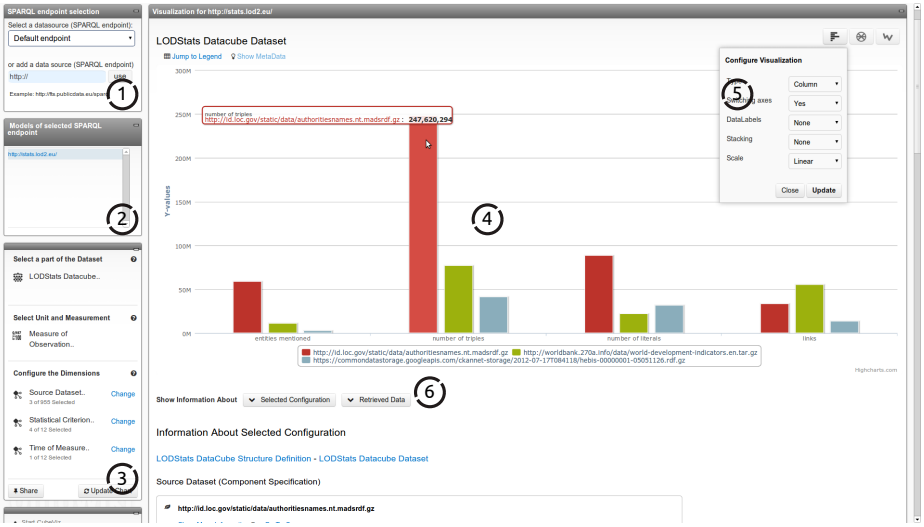

Fig. 5. Comparing the number of triples in different datasets with CubeViz. The numbers in circles refer to: 1 – SPARQL endpoint selection window, 2 – models of selected SPARQL endpoint, 3 – faceted browser for the RDF Data Cube, 4 – the visualization of the selected Data Cube, 5 – configuration for the particular chart type, 6 – area with selected configuration (i.e. dimensions) and retrieved data in plain text.

6 Conclusions

In this demo, we showcased a web application for collection and exploration of the Linked Open Data statistics. We presented LODStats – the base of the web application, an extensible and scalable approach for large-scale dataset analytics on the Web of Data. The web interface of LODStats enables the LOD statistics exploration thus allowing the users to navigate through collected statistical data. We described and performed the transformation of the collected data to RDF and showed how easily it is to visualize such a data with CubeViz.

[13] `http://datahub.io/dataset/hebis-bibliographic-resources`
[14] `http://datahub.io/dataset/library-of-congress-name-authority-file`
[15] `http://datahub.io/dataset/world-bank-linked-data`

Acknowledgment

This work was supported by grants from the EU's 7th Framework Programme provided for the projects LOD2 (GA no. 257943) and GeoKnow (GA no. 318159).

References

1. K. Alexander, R. Cyganiak, M. Hausenblas, and J. Zhao. Describing linked datasets. In *2nd WS on Linked Data on the Web*, Madrid, Spain, April 2009.
2. D. Beckett. Redland librdf language bindings. http://librdf.org/bindings/.
3. D. Beckett. The design and implementation of the redland rdf application framework. In *Proc. of 10th Int. World Wide Web Conf.*, pages 449–456. ACM, 2001.
4. I. Ermilov, J. Demter, M. Martin, J. Lehmann, and S. Auer. LODStats – Large Scale Dataset Analytics for Linked Open Data. Under review in ISWC, 2013.
5. I. Herman, S. Fernández, and C. Tejo. SPARQL endpoint interface to python. http://sparql-wrapper.sourceforge.net/.
6. C. Stadler, J. Unbehauen, J. Lehmann, and S. Auer. Connecting crowd-sourced spatial information to the data web with sparqlify, 2013. http://sparqlify.org/downloads/documents/2013-Sparqlify-Technical-Report.pdf.