LinDA: A Service Infrastructure for Linked Data Analysis and Provision of Data Statistics

Nicolas Beck, Stefan Scheglmann, and Thomas Gottron

WeST – Institute for Web Science and Technologies
University of Koblenz-Landau, 56070 Koblenz, Germany
{nico1510,schegi,gottron}@uni-koblenz.de

Abstract We present LinDA: an extensible, data driven platform where analytical tools can be integrated to process, analyse and describe data sets of Linked Open Data. To date, we have integrated four different tools: the computation of a VoID description, the computation of schema-level index (SchemEX), an information theoretic analysis of the data and the construction of a formal concept lattice. A demo prototype of LinDA is publicly available and allows for uploading data sets for analysis. A web frontend allows users to interact with the system, access the results of previously analysed data sets or to upload new data sets for the analysis tools to operate on.

1 Introduction

The Linked Open Data (LOD) Cloud is growing fast. More and more data sets are published in an open, accessible and machine readable fashion. To make use of this data, end users, developers, data integrators and data engineers need good descriptions and statistics about the available data. Several languages (e.g. VoID), methods (e.g. SchemEX) and tools (e.g. RDFstats) have been proposed to analyse, describe or summarize LOD data on different levels. The uptake of these approaches, however, is far behind its potential. Reasons might be the work overhead in obtaining the description of the data, the installation or usage of research prototypes or simply the lack of awareness about the approaches. Therefore, for many linked data sets there is no good and descriptive meta data available [3].

To alleviate this situation we propose the LinDA platform. LinDA (an acronym for LINked Data Analysis) provides a central platform where we provide analytical tools as a service to the LOD community. Users can upload snapshots of their data sets to this platform. Subsequently, the platform's data driven design enables the user to process the genuine data set with the tools available on the platform, to use output generated by these tools for further analysis, or both. Dependencies between the analytical tools can be modelled to allow for running the tools incrementally, e.g. the computation of a schema-level index precedes an information theoretic analysis using the schema index as input. Furthermore, all the results can be downloaded for offline use, e.g. by the data owner who wants to provide the descriptions together with the original data.

In this paper we present the general idea and setup of LinDA together with a first prototype implementation. As initial services the prototype includes four tools to analyse and describe Linked Data:

1. The generation of a VoID description [1,4] for the data set.
2. The computation of a formal concept lattice based on a formal concept analysis [8].
3. The computation of a SchemEX index [6].
4. An information theoretic analysis on the interdependencies between explicit and implicit schema information [5].

2 Architecture

The aim of LinDA is to enable users to create and share derived analytical and descriptive meta data about LOD data sets in a simple and effective way. To this end LinDA provides services that combine the computation and storage of meta data about data sets. The services are intended to wrap existing or novel analytical tools operating on LOD. The output and results of this analysis are provided for download as bulk files as soon as computation is completed. This approach has been chosen for three reasons: (a) storage of the data enables re-use of the results by other users, (b) in some cases the output of one analysis serves as basis for the execution of another analytical tools and (c) the analysis and its computation might require a long time which demands for asynchronous response to the user. LinDA is an extensible framework. New analytical tools, which build on or integrate existing results, can be developed and deployed.

2.1 Implemented Service

The current prototype version of LinDA includes four services. The general architecture—as mentioned—is extensible to include further services and also allows for the combination of services in workflow sequences. A light-weight XML-based configuration language allows to model dependencies among services. These dependencies specify which computations can be run in parallel and which ones need to be run in a sequential order. This configurations allows for a flexible description of the services, including parameter settings, execution information as well as input and output data format. Therefore, tools are not bound to be implemented in a specific programming language but we can integrate tools of different breed very easily.

Our services operate on RDF data sets. These data sets can be provided in different serializations: RDF/XML, N3, Turtle, N-Triples and N-Quads. As such serializations of RDF can be of quite large size quickly, we allow for the data to be provided in a compressed format: to date we support tgz, gz and zip compression. Along with the data sets we store some basic meta data about the data set. We use basic Dublin core meta data such as a title or name for the data set (dcterms:title) and its publisher (dcterms:publisher). Furthermore, we also ask for meta data used in the VoID vocabulary. This covers information like a textual description of the data set(void:DatasetDescription) or hand picked representative examples resources of the data set (void:exampleResource). This meta data can be provided when a data set is uploaded to LinDA and can help to identify and describe the data set at a very high level and for human users.

LinDA currently provides four services for analysing RDF data sets:

VoiD We compute a VoID description for a given RDF data set. To this end we incorporate a Shell script[1] provided in the context of [4]. The script operates on RDF data sets and generates a VoID description in Turtle notation.

Concept Lattice Construction We have developed a novel analysis service employing a formal concept analysis [8] to extract a concept lattice from a given data set of RDF. The resulting lattice is encoded in RDF and describes the type and property composition of the analysed LOD resources. Additionally to providing the computed concept lattice as bulk download, the service also allows for browsing the descriptions and publishes them as LOD themselves. To this end we implemented a REST API to dereference the URIs of concepts.

SchemEX The definition and construction of a schema based index for LOD is described in [6]. The index uses equivalence classes over schema-level information to segment the analysed data in distinct groups. The groups form an index which allows for the lookup of meta data about the resources being in the group, e.g. number or resources, relevant data sources on the LOD cloud or examples resources. We allow for the computation of such an index using the scalable stream based approach. The index itself is encoded in RDF as well.

Information Theoretic Analysis Using a SchemEX index as computed in the previous service allows for an information theoretic analysis of the data [5]. In this way it is possible to judge the homogeneity or diversity of the explicit and implicit schema information extracted from the data. The results come in the form of a few aggregated metrics, such as Conditional Entropy or Mutual Information.

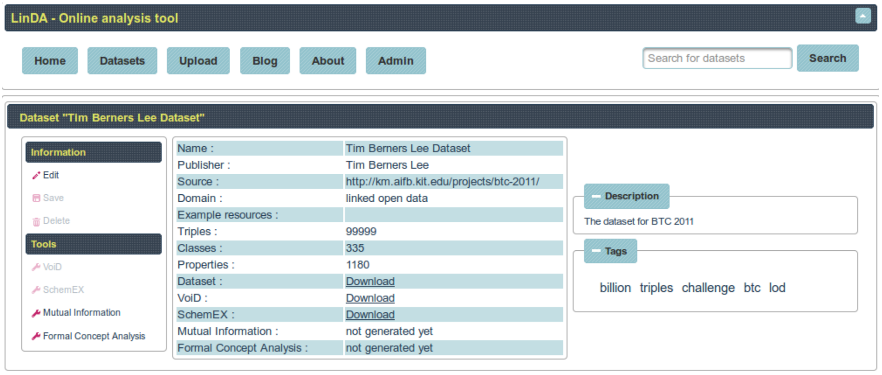
2.2 Data Management and Processing

The processes in LinDA follow a data driven approach. All data sets and results are cached in the system. In this way the results of an analysis do not need to be computed from scratch whenever they are requested, but can be retrieved immediately. This is of importance not only for end users accessing the results but also for services which operate on the outcome of other services. These workflow can then be executed temporally separated and do not need to be executed immediately one after the other.

All services are triggered by events. These events can be issued explicitly by users interacting with the system or by other services which depend on the prior computation of another service. The events triggering the analysis jobs are deserialized into messages and sent to a JMS queue. Message consumers can subscribe to this queue, complete the jobs and store the results. In the current prototype system we have a setup with a single message consumer but the design itself allows for job execution on multiple machines. Making use of a JMS queue also provides a fail safe feature for incompleted jobs. For instance, on huge data sets a small scale consumer machine might run out of resources and fail to complete the requested analysis. These requests are not discarded but stored in the message queue until a consumer is eventually able to process them.

A dedicated services takes care of user notification. This service provides feedback to the end user who has triggered a service as soon as its computation is completed.

[1] The script is available online at `http://code.google.com/p/rdffederator/source/browse/trunk/scripts/generate_void_description.sh`



Dataset "Tim Berners Lee Dataset"

Name :	Tim Berners Lee Dataset
Publisher :	Tim Berners Lee
Source :	http://km.aifb.kit.edu/projects/btc-2011/
Domain :	linked open data
Example resources :	
Triples :	99999
Classes :	335
Properties :	1180
Dataset :	Download
VoiD :	Download
SchemEX :	Download
Mutual Information :	not generated yet
Formal Concept Analysis :	not generated yet

Description

The dataset for BTC 2011

Tags

billion triples challenge btc lod

Figure 1. Information about a data set as presented in the web frontend in LinDA

This is useful in particular for larger data sets, processes with a longer runtime or during peak times when the system has many requests in the queue. In these cases, the users can issue their request and receive an e-mail once their job has been completed.

3 LinDA Prototype

For demo purposes we have implemented a prototype web interface for LinDA (cf. Figure 1). The system is available at `http://linda.west.uni-koblenz.de/`. The web interface lists all data sets which have been uploaded so far along with the descriptive high level meta data provided during upload. In a detailed view, the users can see for each data set which analytical services have already been computed on the data, which services are currently being computed and which results are available. By this means the users do also get an overview of the available services.

When uploading a new data set, users are asked to provide meta data on the title, publisher and a description of the data set. When issuing the computation of an analytical service on the data, users can provide their e-mail address to receive the notification about the completion of the job. Viewers interested in the result of a currently running computation can subscribe to jobs and receive the completion notification as well.

4 Related Work

Analysis of Linked Open Data or RDF data sets in general is addressed by many works. The services we integrated in our framework cover some of these analysis. SchemEX [6], for instance, extracts schema information from RDF to construct a schema-based index structure. This index allows for answering schema-oriented information needs. In a different setting the same index structure is used as basis for an information theoretic analysis of RDF data [5]. VoID is a vocabulary specifically designed for providing meta data about RDF data sets [1]. RDFStats [7] combines several analysis and provides statistical information about RDF data sets as well as SPARQL endpoints.

LODstats [2] is an extensible framework which focusses on statistical analysis of LOD data sets. It integrates RDFStats and makes use of VoID descriptions for computing further statistical metrics. With LinDA we aim for a more general approach which allows for computation of index structures and derived or descriptive meta data as well.

5 Summary

We presented LinDA, a web based service for computing various analytics on Linked Open Data. Data sets can be uploaded and analysed by different services. These services can also be combined into workflows to perform analysis on top of the results of another service. To date, the platform provides four types of analysis. However, the platform is extensible to include additional analysis in the future. In a next step we plan to extend the infrastructure to an actually distributed setup of several machines consuming requests for computing analytical services. Furthermore, we want to extend the dereferencing functionality implemented for the formal concept lattice to the results of the schema-level index in SchemEX. Then, it is possible to directly access the schema itself in the form of LOD. Finally, we plan to integrate visualizations for some of the results where such visualizations are applicable. This will include at least plots of histograms or distributions obtained from the statistical analysis. The visualizations will then be displayed in the web frontend together with the basic meta data available already.

Acknowledgements. The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 257859, ROBUST.

References

1. Alexander, K., Cyganiak, R., Hausenblas, M., Zhao, J.: Describing linked datasets with the void vocabulary. http://www.w3.org/TR/void/, accessed: 11 April, 2013
2. Auer, S., Demter, J., Martin, M., Lehman, J.: Lodstats - an extensible framework for high-performance dataset analytics. In: Knowledge Engineering and Knowledge Management. Lecture Notes in Computer Science, vol. 7603, pp. 353–362. Springer Berlin Heidelberg (2012)
3. Bizer, C., Jentzsch, A., Cyganiak, R.: State of the lod cloud. http://www4.wiwiss.fu-berlin.de/lodcloud/state/, accessed: 11 April, 2013
4. Görlitz, O., Staab, S.: SPLENDID: SPARQL Endpoint Federation Exploiting VOID Descriptions. In: Proceedings of the 2nd International CoLD Workshop. Bonn, Germany (2011)
5. Gottron, T., Knauf, M., Scheglmann, S., Scherp, A.: A Systematic Investigation of Explicit and Implicit Schema Information on the Linked Open Data Cloud. In: ESWC'13: Proceedings of the 10th Extended Semantic Web Conference (2013), to appear
6. Konrath, M., Gottron, T., Staab, S., Scherp, A.: Schemex—efficient construction of a data catalogue by stream-based indexing of linked data. Web Semantics: Science, Services and Agents on the World Wide Web 16(5), 52 – 58 (2012), the Semantic Web Challenge 2011
7. Langegger, A., Woss, W.: Rdfstats - an extensible rdf statistics generator and library. In: Proceedings of the 2009 20th International Workshop on Database and Expert Systems Application. pp. 79–83. DEXA '09, IEEE Computer Society, Washington, DC, USA (2009)
8. Wille, R.: Restructuring lattice theory: An approach based on hierarchies of concepts. In: Ferré, S., Rudolph, S. (eds.) Formal Concept Analysis, Lecture Notes in Computer Science, vol. 5548, pp. 314–339. Springer Berlin Heidelberg (2009)