Discovering Issues in Datasets Using LODSight Visual Summaries

Marek Dudáš and Vojtěch Svátek

Department of Information and Knowledge Engineering,
University of Economics, W. Churchill Sq.4, 130 67 Prague 3, Czech Republic,
{marek.dudas|svatek}@vse.cz

Abstract. Quality-checking of linked data is a hot topic nowadays. As complement to fully automated quality analysis we propose issue discovery via manual exploration of dataset summary graphs. Our LODSight summary visualizer has been extended with new features, ontology/predicate filtering, instance picking and multi-dataset summarization, so as to better support this task. Three scenarios of dataset issue discovery have been investigated with the help of the extended tool.

1 Introduction

Quality checking of RDF datasets is a widely researched topic with diverse approaches being applied [7]. Automated error detection might be test-driven, where a set of tests implemented, e.g., as SPARQL queries might be run to search for incorrect predicate usage or typing. However, the tests have to be prepared in advance. This may work well for checking the usage of entities from a single ontology; datasets however often refer to several ontologies, and preparing and maintaining a set of tests for every possible combination of ontologies that might be used in the dataset does not seem feasible. Using reasoning and checking for inconsistencies is an obvious option, but requires the ontologies to be systematically equiped with axioms, including, e.g., the class disjointness ones, which is not always the case (e.g., in the DBpedia ontology). Manual, user-driven evaluation of facts in the dataset has also been proposed, but its scalability is obviously limited.

As a novel approach we propose to first *summarize* the dataset graph/s and then to apply specifically tailored *visualization* over the summary, allowing for manual discovery of issues. Depending on context, the visual exploration of summaries may either precede the automated quality analysis (indicating, e.g., on which predicates the tests are to be run), or, conversely, focus on parts of the dataset already indicated as problematic by automated analysis; for smaller datasets the analysis in visualizer might even be sufficient. In the paper we present several possibilities how a previously developed dataset summary visualization tool, *LODSight* [2], enriched with several new features, can be used as a complement to existing error detection systems.

Related research The visualization in LODSight is similar to *maps of ontology usage* [4] and Explod [3]. Both tools could also be used for error detection in a similar way as presented with LODSight. We are unaware of any research focused on exploiting visualization for dataset error detection. However, many non-visual approaches to error detection exist. Atencia et al. [1] proposes finding pseudo-keys in the dataset and using them to detect errors such as a person with the same death and birth date. Detection of such errors cannot be performed nor supported with LODSight type of visualization – it is too general to allow comparison of values linked to specific instance. Outlier detection implemented by Paulheim [6] could be supported by LODSight: combinations of classes and properties detected as outliers could be e.g. highlighted in the visualization. A simpler form of outlier detection can be even performed in LODSight by looking at type-property combinations with lower frequency in the dataset represented by link thickness. Error detection done by Péron et al. [8] uses domain/range axioms from ontologies. Kontokostas et al. [5] implemented versatile error detection based on SPARQL queries automatically created from patterns. Complex approaches like the last two mentioned obviously can reveal errors that cannot be seen in the simplified visualization. However, the general overview of the dataset contents provided by the visualization might still help to determine which approaches to error detection should be used for the specific dataset. Datasets can be also checked for errors manually, as shown by Zaveri et al. [9].

2 LODSight

LODSight[1] is a dataset summary visualization tool. It uses SPARQL to find all type-property and datatype-property paths in the dataset. Type-property path is a sequence `type1 - property - type2`. `Type1` and `type2` are the types of instances from the dataset that are connected by the property. We use the term *path frequency* to denote the number of triples `?s ?property ?o` in the dataset where `?s` is an instance of `type1` and `?o` of `type2`. Datatype-property paths are analogous sequences of `type - datatype property - datatype`. All paths are merged into one graph and visualized in one view allowing the user to see generalized structure of the dataset and usage of ontologies in it. The visualization is interactive and the user can also filter the displayed paths to show only those with lower or higher frequency. The summarization is run offline as it might be prohibitively time-consuming in case of larger datasets. The results of the summarization are stored in a database. A list of previously summarized datasets is offered to view in the LODSight web application. To support error detection, we implemented several new features.

Ontology Filter Whenever dataset visualization is loaded, a list of ontology IRIs used in the dataset is shown. Users can select any subset of the IRIs to limit the visualization to entities from the selected ontologies and entities linked directly to them. This way users can analyze usage of selected ontology in the context of the dataset.

[1] Available at `http://lod2-dev.vse.cz/lodsight-v2`

Predicate Filter Similarly to the ontology list, a list of all properties used in the dataset is displayed. When a subset of the properties is selected, only the entities linked with them are shown. Users can thus analyze their usage without other links cluttering the view.

Analyzing Example Instances Users can select a subset of class nodes in the graph and retrieve their example instances that are linked with the properties shown in the generalized graph. The labels or URIs of the instances are displayed above the class nodes. The user can click on any of them to open a new browser tab where the resource description is retrieved. This makes manual checking of the facts related to the instances easier.

Merging Summarizations of Several Datasets Any number of the available dataset summarizations can be selected in the list and then visualized in one view. The paths from all the selected summarizations are simply merged into one graph and displayed. This feature can be useful in conjunction with ontology filter – see Section 3.3 for more details.

3 Preliminary Tests in Example Usage Scenarios

3.1 Analyzing Large Dataset with Predicate Filter

As an example of a large dataset, we used Greek DBpedia. Visualizing the whole summarization is simply impossible in this case as it contains thousands of paths and the resulting visualization is too cluttered and thus unreadable. A way to get an overview of possibly erroneous parts of the structure would be to limit the maximum path frequency to a very low number. In this case, that still leads to too many results and unreadable visualization. So does filtering the visualization by ontology. A feasible option is to filter by predicate. We can go through the predicates one by one, or select those suggested by some other error detection method or by an expert. Consider the latter case, where, e.g., the property *dbo:child* from DBpedia ontology was identified as possibly incorrectly used and thus selected in the predicate filter. In the resulting visualization (Fig. 1) we can immediately see classes like *dbo:WrittenWork*, whose instances clearly should not be linked with *dbo:child* property. We manually adjust the visualization to focus on one of them and see that *dbo:WrittenWork* is linked to *dbo:Person* with *dbo:child*. We select the two class nodes and retrieve their example instances. Their labels are shown above the class nodes (Fig. 2). They are in Greek, so perhaps not yet helpful by themselves, but we can click on them and their description is opened in a new browser window. There we can see that both instances are actually persons, but one of them was incorrectly typed as book.

3.2 Showing the Whole Structure To See Missing Links

Smaller summarizations (approx. up to hundred paths) of less complex datasets can be visualized as whole in a single view. This may allow to see another type of error: missing links. Consider the visualization of the RISM Authorities dataset

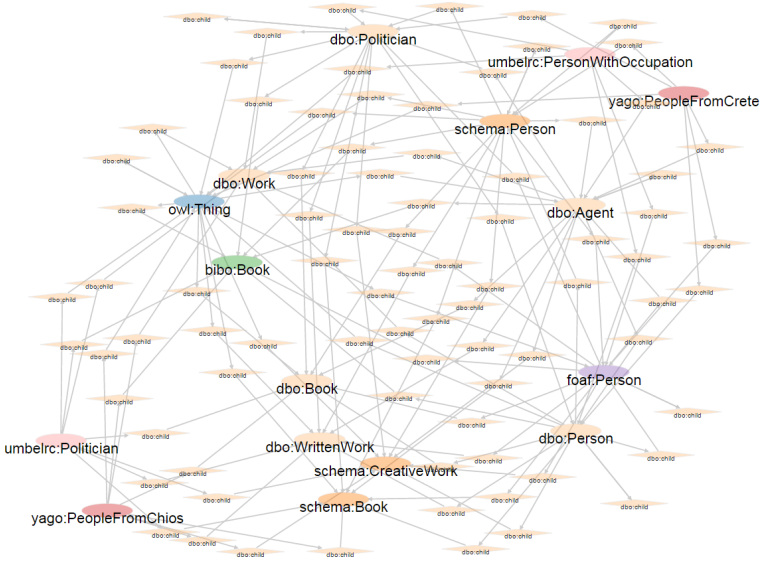

Fig. 1. Usage of dbo:child property in Greek DBpedia visualized in LODSight.

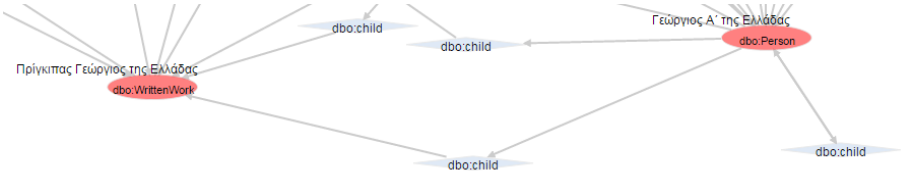

Fig. 2. Example instances of dbo:WrittenWork and dbo:Person linked with dbo:child.

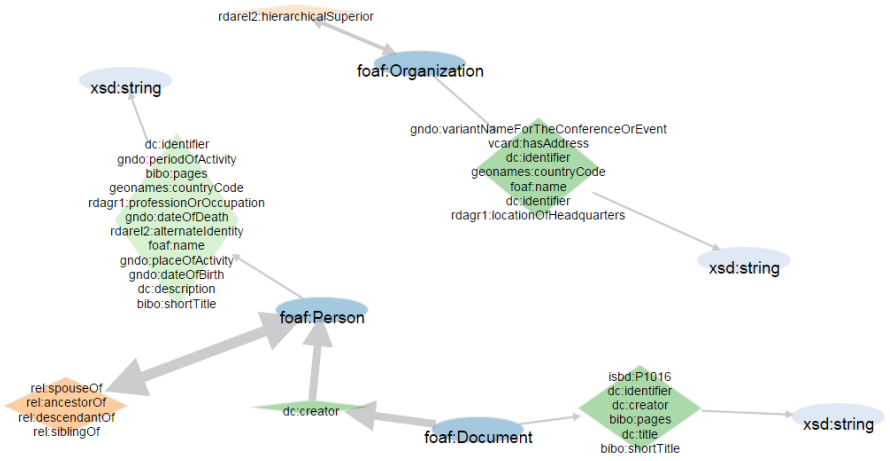

Fig. 3. RISM Authorities dataset summarization in LODSight.

summarization in Fig. 3. It shows that the dataset contains interlinked persons and documents. There are also organizations that are not linked to any person nor document. That indicates that some facts may be missing in the dataset: e.g., that persons are members of organizations or organizations are owners of documents.

3.3 Detecting Errors in Ontology Usage Across Several Datasets

Filtering the visualization to entities of a selected ontology might be used in conjunction with merging summarizations of several datasets into one graph. This way an expert on the given ontology might check its usage in several datasets at once, instead of looking at each dataset separately. Consider an expert on FOAF who wants to check if the ontology is used correctly in the RISM Authorities and ESWC2015 datasets.[2] The expert selects both datasets and sets the ontology filter to FOAF. The result is in Fig. 4. The expert might spot, e.g., the possibly

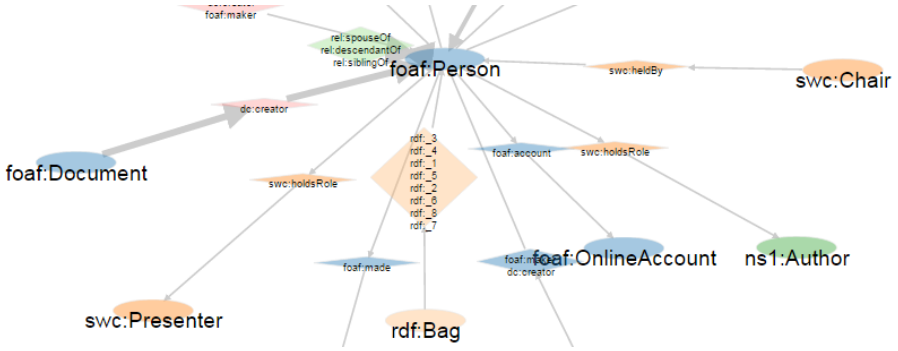

Fig. 4. A part of the visualization of FOAF usage merged from two datasets.

suboptimal usage of *dc:creator* to link *foaf:Document* to *foaf:Person* (*foaf:maker* is recommended by FOAF documentation to link document and person instances instead of *dc:creator*). The expert could find out which datasets contain such triples and inform their maintainers about the proper usage. The functionality for showing which part of the graph comes from which dataset in such merged visualization is however yet to be implemented.

4 Conclusions and Future Work

We proposed that dataset structure visualization might be helpful for detecting errors in a dataset. We enriched existing dataset summary visualization tool, LODSight, with several new features and showed their usage in several scenarios

[2] Randomly chosen out of the datasets we summarized with LODSight so far.

of error detection. Preliminary results suggest that in case of large datasets[3] the capabilities of the visualization are somewhat limited – the same results can be achieved using some existing automated error detection method more easily. For smaller datasets, whose whole summarizations can be viewed on one screen, we so far identified two possible use cases when the visualization might be useful: finding missing links and checking ontology usage across several datasets. Although the former might be done automatically without the visualization, the visualization may allow an expert user to more easily decide whether disconnected subgraphs in the summarization are a result of an error or just a coincidence. The latter cannot be easily replaced by automated tests, since it would be hard to prepare tests for every possible combination of properties and classes from different ontologies; in contrast, an expert can spot the incorrect usage immediately in the visualization. Future work will include investigating other error detection scenarios, thorough evaluation, and reliability enhancement of the tool.

The research is supported by UEP IGA F4/90/2015 and by long-term institutional support of research activities by Faculty of Informatics and Statistics, Univ. of Economics, Prague.

References

1. Atencia, M., David, J., Scharffe, F.: Keys and pseudo-keys detection for web datasets cleansing and interlinking. In: Knowledge Engineering and Knowledge Management, pp. 144–153. Springer (2012)
2. Dudáš, M., Svátek, V., Mynarz, J.: Dataset summary visualization with LODSight. In: The 12th Extented Semantic Web Conference (ESWC2015). `http://lod2-dev.vse.cz/lodsight/lodsight-eswc2015-demopaper.pdf`
3. Khatchadourian, S., Consens, M.: Explod: Summary-based exploration of interlinking and RDF usage in the linked open data cloud. The Semantic Web: Research and Applications pp. 272–287 (2010)
4. Kinsella, S., Bojārs, U., Harth, A., Breslin, J.G., Decker, S.: An interactive map of semantic web ontology usage. In: Information Visualisation, 2008. IV'08. 12th International Conference. pp. 179–184. IEEE (2008)
5. Kontokostas, D., Westphal, P., Auer, S., Hellmann, S., Lehmann, J., Cornelissen, R., Zaveri, A.: Test-driven evaluation of linked data quality. In: Proceedings of the 23rd international conference on World Wide Web. pp. 747–758. ACM (2014)
6. Paulheim, H.: Identifying wrong links between datasets by multi-dimensional outlier detection. In: 3rd International Workshop on Debugging Ontologies and Ontology Mappings, WoDOOM (2014)
7. Paulheim, H.: Automatic knowledge graph refinement: A survey of approaches and evaluation methods (2015), `http://www.semantic-web-journal.net/system/files/swj1083.pdf`
8. Péron, Y., Raimbault, F., Ménier, G., Marteau, P.F.: On the detection of inconsistencies in RDF data sets and their correction at ontological level (2011)
9. Zaveri, A., Kontokostas, D., Sherif, M.A., Bühmann, L., Morsey, M., Auer, S., Lehmann, J.: User-driven quality evaluation of DBpedia. In: Proceedings of the 9th International Conference on Semantic Systems. pp. 97–104. ACM (2013)

[3] In terms of the number of combinations of classes and properties used in the dataset.