An Intelligent Linked Data Quality Dashboard

Ramneesh Vaidyambath[1], Jeremy Debattista[2][0000-0002-5592-8936], Neha Srivatsa[3], and Rob Brennan[1][0000-0001-8236-362X]

[1] ADAPT Centre, School of Computing, Dublin City University, Dublin 9, Ireland
ramneesh.vaidyambath2@mail.dcu.ie, rob.brennan@dcu.ie
[2] TopQuadrant, Raleigh, NC 27606-5228, USA
[3] Northern Kentucky University, Highland Heights, KY 41099, USA

Abstract. This paper describes a new intelligent, data-driven dashboard for linked data quality assessment. The development goal was to assist data quality engineers to interpret data quality problems found when evaluating a dataset using a metrics-based data quality assessment. This required construction of a graph linking the problematic things identified in the data, the assessment metrics and the source data. This context and supporting user interfaces help the user to understand data quality problems. An analysis widget also helped the user identify the root cause multiple problems. This supported the user in identification and prioritization of the problems that need to be fixed and to improve data quality. The dashboard was shown to be useful for users to clean data. A user evaluation was performed with both expert and novice data quality engineers.

Keywords: Linked Data, Data Quality Analysis, Root Cause Analysis.

1 Introduction

Data quality is often defined as "fitness for use", which characterizes the ability of data to meet users' requirements [1]. Data quality is often determined by evaluating if its features meet the user's requirement. Data quality is often described as a multi-dimensional concept where each dimension is related to a specific user-focused aspect of quality such as accuracy, completeness, consistency, timeliness, and accessibility [2].

Data Quality is a key challenge in Linked Data as the data is often transformed from multiple heterogeneous sources, including semi-structured and unstructured data, which are of varying quality [3] [4]. The term Linked Data refers to a set of best practices for publishing and connecting structured data on the Web [5]. For Linked Data, in addition to generally accepted data quality dimensions, additional dimensions have been defined by Zalveri et al. [3].

Data quality assessment is a process for evaluating if a data meets the user's specific needs [3]. It is usually carried out using a data quality assessment framework. The dimensions are indirectly measured using one or more quality metrics. These metrics report a series of values (typically normalized between 0 and 1) which can then be compared to desired thresholds for pass/fail quality assessment or monitoring quality trends over time. Some assessment frameworks also generate problem reports for the malformed or missing data they detect when calculating the metrics. A single flaw in the dataset can create a cascade of reports as multiple metrics are assessed.

To improve data quality, data corrections are required. It is crucial that the user performing the data correction should have a clear understanding of which quality problems (flaws) are present in the data, how to fix them, where they occur in the dataset, which metrics are impacted by each flaw and how much improvement each fix would bring. Data quality is typically an expensive process and thus a prioritization of fixes is important as there is a trade-off between cost and quality.

In this paper, the research question is to determine "*to what extent will an intelligent dashboard based on knowledge graphs and root cause analysts assist the user understand quality problems in Linked Data and thus enable the user to identify appropriate repairs and prioritize them*".

The following technical approach was followed: i) an existing Data Quality Assessment Framework was selected to assess the data (i.e. generate metrics) and generate data quality problem reports; ii) design and build a new high usability user interface to fetch and view data quality metadata and problem reports; iii) develop supporting services to integrate the data and problem reports, and then identify the related problems for each problem through root cause analysis; iv) display flaws in the dataset in the context of all the related problems.

The remainder of the paper is structured as follows: Section 2 discusses the use cases for the dashboard, section 3 presents background and related work on Linked Data quality assessment and Root Cause Analysis; section 4 describes the design of the dashboard tool; section 5 describes the evaluation of the dashboard tool; section 6 describes the conclusion along with the scope of future work.

2 Use Cases

The users of this dashboard will either be data quality engineers or managers in organisations that manage data. The following use cases were developed in the context of the Ordnance Survey Ireland data pipeline to identify dashboard features.
UC1: Assessment of Data. Users should have the means to assess data. In the case of Linked Data, the data can be either in any Linked Data supported serialization formats or published in some database which can be accessed using a Uniform Resource Locator (URL). The dashboard should also allow the stakeholders to configure the data details that need to be assessed along with what needs to be assessed in the data.
UC2: Display Data Quality. The quality of an assessed dataset needs to be displayed to the users to help them decide whether the data can be published or whether the data can be used in their application. Display the overall data quality, dimension level data quality and also the metric level data quality.
UC3: View Data Quality Problems. Users need a means to view the data quality problems that are present in the problem reports of an assessed dataset, to assist the user's in understanding it. Any relevant information's related to each of the data quality problem, such as what exactly does the problem means, what caused this problem to occur,

where else does this problem occur and are there any related problems, can help the stakeholder to get a clear understanding of the problem.
UC4: Enable Root Cause Analysis. Stakeholders should be able to decide on the exact cause of a problem in a problematic thing, so as to decide on how it needs to be fixed. This can be performed by the stakeholder's once all the relevant information's are available and attached to each problematic things. Backend service which prepares this relevant information's to the user and corresponding widgets to display them in a user-friendly manner so as to conclude on the exact root cause of a problem.

3 Background and Related Work

The foundations discussed are Linked Data, Data Quality and Root Cause Analysis.

3.1 Root Cause Analysis

Root Cause Analysis (RCA) is an event analysis technique [9] that helps to identify what, how and why something happened e.g. in network troubleshooting. This is usually an event or outcome that is undesirable. Understanding why it occurred is crucial to develop an effective correctable measure to correct and prevent such undesirable outcomes in the future. The goal of RCA is to identify such underlying causes using a structured approach. The four major steps in the RCA process as explained in [9] are: i) Data Collection, ii) Causal Factor Charting, iii) Root Cause Identification, iv) Recommendation Generation and Implementation. The analytics widget on the dashboard (see later) is designed to perform the data collection and causal factor charting steps automatically and thus enable the user to perform the root cause identification and recommendation generation steps of those problems that need to be fixed.

Three generic standard tools for Root Cause Analysis are: Cause-and-effect diagram (CED), Interrelationship Diagram (ID), Current Reality Tree (CRT). These three tools have been evaluated in [11] to find root causes with varying degrees of accuracy, efficiency, and quality. Based this analysis, CED is utilized in this research to organize the casual relationships between the problematic thing and the problems in the data. In specific, the cause enumeration CED method is used, which simply lists all the possible problems and organizes them in a RDF graph with their relationship with the problematic thing. The advantage of this method is that all the proposed problems are listed and encourage thinking for the solution without any restriction [11].

3.2 Linked Data

In 2007 Berners-Lee outlined Linked Data as a set of 'rules' for publishing data on the Web so all data can be connected as a single global data space [5]. The two fundamental technologies on which Linked Data relies on are Uniform Resource Identifiers (URIs) and Hypertext Transfer Protocol (HTTP). Resources or entities in the Web are identified by URIs and HTTP provide a universal method to retrieve information about these entities.

Linked Data uses the W3C Resource Description Framework (RDF) [6] which is a graph-based data representation model to structure and link the data about each entity or link the entities itself. The RDF model represents data as subject, predicate, and object which together is known as a triple. The subject and object of a triple are usually both URIs or a URI and a literal which identify a resource. The predicate specifies how subject and object are related. The RDF Vocabulary Definition Language (RDFS) [7] and Web Ontology Language (OWL) [8] defines languages (syntax and semantics) for creating Linked Data vocabularies that can be used to describe domains.

The SPARQL Query Language is the widely used query language to retrieve and manipulate RDF/Linked Data. It provides the means to query required and optional graph pattern along with their conjunctions and disjunctions.

3.3 Linked Data Quality Assessment

Data Quality issues in Linked Data have some unique aspects but most are common to the general discipline of Data Quality. The survey paper published by Zaveri et al. [3], provides a systematic review of existing approaches to assessing the quality of Linked Data, with a comparison of 12 data quality assessment tools. Eighteen data quality dimensions and in total 69 metrics are examined with an indication of whether they are measured qualitatively or quantitatively. Qualitative results are based on human observations and quantitative results are counted or measured objectively. All the data quality assessment metrics in this paper are quantitative metrics.

For the purposes of satisfying our use case (section 2), a comparison between the LinkQA, Sieve, RDF Unit, Triple Check-Mate, and Luzzu Linked Data quality assessment frameworks is given here and summarized in Table 2 below. The focus is on the support for interoperability (standard metrics observation reports), generation of problem reports in RDF format, ability to suggest which problems to be corrected and root cause analysis capability.

LinkQA [12] is a framework for assessing Linked Data quality using network metrics. It is extensible, which makes it easy to incorporate additional metrics. It generates an HTML based report to display the results of quality assessment. It doesn't support any correction techniques, nor does it list the problems found in the data. The also tool lacks a user-friendly user interface and so is low usability.

Sieve [13] is a Command Line Interface which performs a quality assessment based on the provenance metadata graph which is generated from a data source. The provenance metadata provides the history of the origin of data. The tool is limited to domains providing provenance metadata. Also, it doesn't generate any problem reports.

RDF Unit [14] is a Command Line Interface for test-driven quality assessment of Linked Data. It assists in defining quality test patterns using SPARQL query templates. Thus it helps to assess the data set based on custom SPARQL queries. But such SPARQL queries cannot assess complex metrics. It does generate the problem report which is in RDF supported format.

Triple Check-Mate [15] is a data quality assessment tool which identifies and presents the problem present for each resource to the user. It does generate a problem report to be viewed in its HTML front-end for user evaluation and correction. But the problem report generated is stored in MySQL database in Non-RDF supported format. It enables the user to understand the problems associated with each triple of a resource and thereby assist the user to correct the problems.

Luzzu [16] is a web service-based quality assessment tool which supports interoperability. The metrics in Luzzu is customizable and highly scalable. It also generates a detailed problem report, in RDF format, which can assist in understanding the data quality problem. But currently, Luzzu doesn't employ any analysis of this generated problem report. The problem report is dumped as an RDF serialized output file, which doesn't support SPARQL queries to fetch the data effectively. Also, for data that is assessed, there is no identifiable link between the quality metadata and the problem report. The original Luzzu UI only displays the quality score and does not display the identified problems to the user nor assist the user to understand the identified problems.

Table 1. Comparison Between Existing Linked Data Quality Assessment Frameworks/Tools.

Feature \ Tool	LinkQA	Sieve	RDFUnit	Triple Check Mate	Luzzu
User Interface	×	×	×	✓	✓
Interoperability	×	×	×	×	✓
RDF format supported Problem Report	×	×	✓	×	✓
Suggest Problems to Correct	×	×	×	✓	×
Support Root Cause Analysis	×	×	×	×	×

Of these tools, the Luzzu framework's feature to generate the detailed problem reports in RDF makes it highly suitable for our technical approach. Its web services make it highly interoperable and with a few additional features (described later), this makes Luzzu the best tool to assess the quality of Linked Data. Table 2 provides a comparison between the old versions of the Luzzu GUI with the proposed one (fig. 3).

Table 2. Comparison Between Existing Luzzu Dashboard Features.

Feature \ Tool	Initial Version	OSI Version	Proposed Version
Interdependence between Framework and user Interface	High	High	Low
Overall Data Quality Summary Statistics	✓	✓	✓
Dimension Summary Statistics	×	×	✓
Metric Summary Statistics	✓	✓	✓
Detailed Problem Report	×	×	✓
Support Root Cause Analysis and Fix Prioritization	×	×	✓

4 Design

This section will give you an overview of the new system design which includes the Luzzu Framework for quality assessment, the Intelligent Dashboard for data analysis and the supporting tools and services which makes it possible.

4.1 Architecture

The system with the new analytics dashboard consist of 4 major components (as shown in Fig 1): Luzzu Framework, Triplestore, Analytics Dashboard, and Service Wrapper.

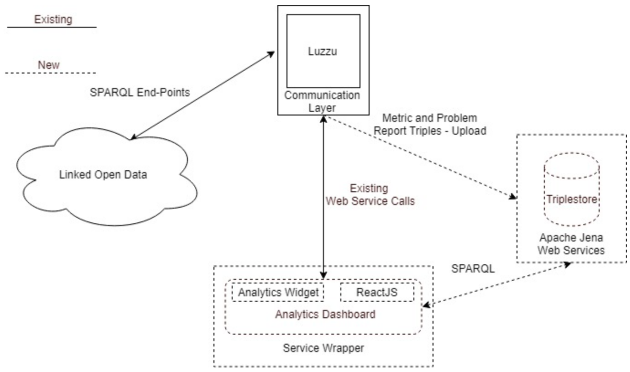

Fig. 1. Intelligent Dashboard System Architecture

Luzzu Framework:

The Luzzu Framework is extended to include direct connection to a triplestore for storage of the metadata and problem report generated during each assessment of a Linked Data dataset. This makes the system more flexible as the Dashboard can directly query the outputs using SPARQL in the triplestore. The metrics and its corresponding dimensions listed in table 3 are assessed by Luzzu for the experiments listed later.

Table 3. List of Quality Metrics and Corresponding Dimensions.

Metric	Dimension
Undefined Classes and Properties	Interpretability
Human Readable Labelling and Description	Understandability
Extensional Conciseness (Estimated)	Conciseness
Machine-readable License	Licensing
Incorrect Domain or Range Datatypes	Consistency
Correct use of Entities as Members of Disjoint Classes	Consistency
Misplaced Classes or Properties	Consistency
Misused Owl Datatype Or Object Properties	Consistency
Compatible Datatypes	Syntactic Validity

Apache Jena Triplestore: Apache Jena Fuseki is used as the SPARQL triplestore. It consists of Trivial Database (TDB) component for RDF storage and query. File Upload service exposed by Fuseki Server is utilized to load the Metadata and Problem Report generated in Luzzu. The metadata is stored as a separate graph in the triplestore where the metadata graph name can be found in the default graph. Each problem report generated is stored as an independent graph in the triplestore with a unique link to the metadata graph. SPARQL queries are used for remote access of data from the Triplestore via HTTP.

Service Wrapper: This component consists of several API services to assist the Analytics Dashboard in performing back-end knowledge base operations as well as state management of the dataset being assessed. The state management services maintain the configuration state of each dataset that is added in the dashboard for assessment in persistent file storage. The configuration details include the dataset name, dataset SPARQL end-point or dataset dump filename, unique identifier and the metrics that need to be assessed along with the required acceptance threshold per metric. Any changes made to the dataset configuration details in the dashboard is also updated in the persistent file storage with the help of these services. The knowledge base operation services generate the unified knowledge base which consists of all the triples with a problematic thing along with all the related exceptions to support root cause analysis. Exceptions are the issues identified for a resource, triple or part of the triple in Luzzu. The knowledge base generation pipeline is shown in Figure 2.

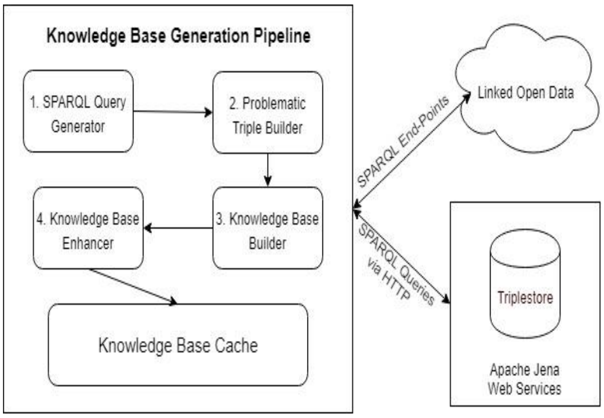

Fig. 2. Knowledge Base Pipeline

Table 4 lists how knowledge base generation service identifies the correct part of the triple which is a problematic thing and hence map it to identify the related problems based on to which part of the triple does the problematic things belongs to.

Table 4. List of Quality Metrics and Corresponding Dimensions.

Metric	Exception(s)	Problematic Thing Part
Undefined Classes and Properties	Undefined Class	Object
	Undefined Property	Predicate
Human Readable Labelling and Description	Human Readable Label	Subject
Extensional Conciseness (Estimated)	Resource Replica	Subject
Incorrect Domain or Range Datatypes	Incorrect Range	Predicate, Object
	Incorrect Domain	Subject, Predicate
	Unknown Type	Subject, Object
Entities as Members of Disjoint Classes	Multi Typed Resource with Disjointed Classes	Subject
Misplaced Classes or Properties	Misplaced Class	Predicate
	Misplaced Property	Object
Misused Owl Datatype or Object Properties	Misused Object Property	Predicate, Object
	Misused Datatype Property	Predicate, Object
Compatible Datatypes	Dt-unknown-dt	Object

Based on the metric assessed, all the problems identified for that metric is fetched from the triplestore. The triplestore internally fetches the complete triple from the dataset to form the Subject, Predicate and the Object. This builds the complete problematic triple. Each problematic triple constructed is then added to the knowledge base along with the failed metric and the corresponding exception. Once all the problematic triple is added to the knowledge base, it is then enhanced by identifying all the related exceptions raised for the problematic part of the triple. The identified related exceptions are considered as the causal factors of the problem in root cause analysis. This knowledge base in the added to the cache which is maintained by the service and cleared after a certain configured time period.

Analytics Dashboard: The new high usability Analytics Dashboard (Fig. 3) provides a GUI where the user can configure the details of the dataset that need to be assessed along with the metrics. It also consists of a new Problem Report Widget and Analytics Widget. The Problem Report Widget lets the user view all the problems related to the assessed dataset on a per metric basis. The Analytics Widget provides a summary of each problem in the assessed dataset along with the casual factors identified via RCA for each independent problem. This helps the user to identify the root cause of the problem. The summarized view also helps the user to pick out those issues that need to prioritize in fixing. The dashboard intelligently sorts the problem list based on the problems which have the most impact on the dataset. The Resource View and Problematic Thing View let the user view the problems in two additional ways. The Resource View helps the user understand all the problems identified per resource. It also identifies any resource replicas. The Problematic Thing View helps the user understand each problem type detected. It provides information such as impacted resources, related exceptions raised for the same problematic thing and linked problems. It is intended to provide inferential tests to assist in this process in further work.

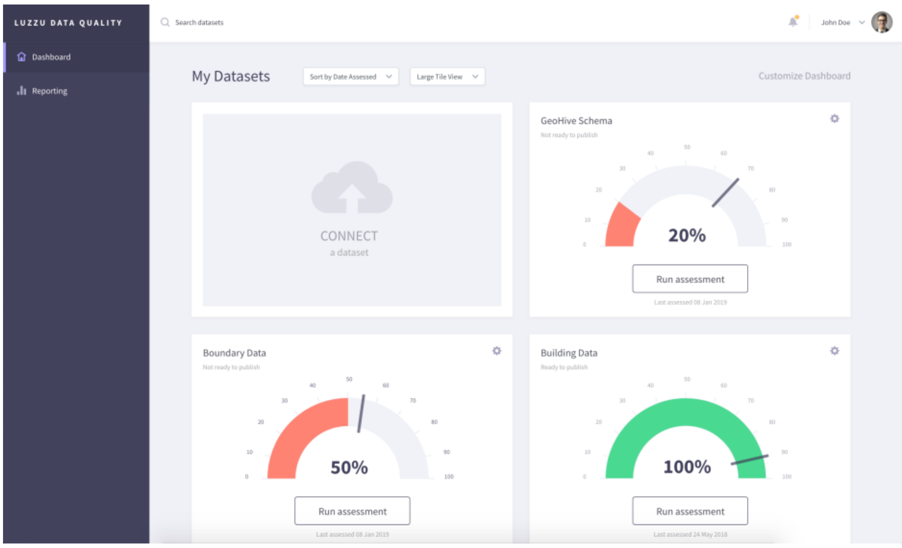

Fig. 3. New Dashboard Design

5 Evaluation

This section describes the user experiment performed to evaluate the new dashboard.

Overview: The purpose of this experiment was to evaluate the extent to which the new dashboard with data-driven and analytics widgets was helpful for the users in *understanding specific data quality problems* discovered by the Luzzu data quality assessment tool in Linked Data. It also evaluates the extent to which the widgets were able to assist the user in *identifying the root cause* of a problematic thing (data issue) and thus identify the priority issues to resolve in order to improve the data quality score.

Hypotheses: The experimental hypotheses are as follows:

H1: The Problem Report Widget and Analytics Widget provided both expert and novice users with a better understanding of the data quality problems identified by Luzzu.

H2: The Analytics Widget enhanced the user's ability to identify and prioritize which problems to be fixed than the Problem Report Widget.

Dataset: The Ordnance Survey Ireland's buildings data[1] is used for the experimental evaluation. This consists of geospatial data about various buildings in Ireland as Linked Data which is published on the Web. This is a general data which adheres to the Linked

[1] http://data.geohive.ie

Data best standards in publishing and usage. The active collaboration on ADAPT Center with OSi, the interest of OSi data engineers to this dashboard and prior understanding of data quality problems related to this data are the main factor for using this data for the experiment. Access to this data can be requested by researchers by requesting to OSi. The quality of this data is assessed by Luzzu for those metrics that are listed in table 3.

Users: The users consisted of two groups: i) 4 novice users and ii) 4 expert users. The novice users consisted of computer science master's students in DCU and the expert users were Linked Data researchers from the ADAPT center. The novice users consist of those who have basic knowledge of web technologies and user interfaces. The expert users consist of those who have basic knowledge of web technologies, user interfaces and also Linked Data quality concepts.

Procedure: The experiment consisted of four tasks which the users had to perform either using the problem report widget or the analysis widget. The users were given the following set of tasks:

i) List all the problems identified for a particular resource.
ii) Identify which part of a triple is a problematic thing.
iii) List the problem identified for problematic thing for a particular metric.
iv) Identify the root cause problem for each problematic thing and prioritize the problematic things that need to be fixed.

The tasks i, ii and iii are created to determine to what extent the user has understood about the data quality problem, identified by Luzzu, with the help of Problem Report Widget and Analytics Widget in the new dashboard. Task iv is created to determine which widget was better at assisting the user to identify and prioritize the fix required to improve the data quality. These tasks were designed to ensure that the user uses multiple features of the dashboard to conclude on the answers which help the user understand a data quality problem. The users were asked to perform these tasks independently using the dashboard, which was hosted online and then selects appropriate answers from the questionnaire. The qualitative measures for each of these tasks are recorded by the user in the questionnaire along with other metrics such as effort, confidence, and usefulness.

Data Collected: Table 5 contains the raw data collected for each task.

Analysis Method: The values collected during the experiment is used to gather evidence that the hypothesis H1 and H2 are true. The response score was calculated from the questionnaire response for each of the tasks. This score is based on the correct and incorrect responses logged by the user for each task. The response score for task i, ii and iii is used to gather evidence for hypothesis H1 and the response score for task iv is used to gather evidence for hypothesis H2. The values for the effort, confidence, and usefulness metrics are used to compare the usability and effectiveness of both the widgets. The effort score determines how organized and concise the widget was to help the user to point to the required information. The lesser the effort, the better. The confidence score shows how effective the widget was in helping the user to find the problem or required information regarding a problem. The higher the confidence score, the better the widget performed. The usefulness score indicates how much useful the user felt the information’s provided by the widget was in helping them to understand the problem and then answer the relevant queries.

Table 5. User Data Collected.

	User		***Response Score by Task***				***Effort by Task***				***Confidence by Task***				***Usefulness by Task***			
	Type	***No.***	***1***	***2***	***3***	***4***	***1***	***2***	***3***	***4***	***1***	***2***	***3***	***4***	***1***	***2***	***3***	***4***
Problem Report	N	U1	10	10	10	0	4	4	3	7	8	6	8	8	8	8	8	8
		U3	10	10	10	0	4	4	4	7	7	7	7	6	5	6	6	6
	E	U2	10	10	10	10	2	2	2	4	9	7	9	7	8	8	9	8
		U7	10	10	10	10	3	3	4	5	8	7	8	7	8	8	8	8
Analytics	N	U5	10	10	0	3.3	5	4	5	7	9	10	5	5	8	8	5	8
		U6	10	10	0	5	6	3	5	9	9	9	7	3	3	5	5	4
	E	U4	10	10	10	10	1	1	1	9	1	10	9	10	9	10	6	7
		U8	10	10	10	10	2	3	2	7	9	9	9	9	8	8	8	8

Analysis of Data: The average scores for tasks i, ii and iii (problem understanding) for novice users and expert users was 25 and 30 respectively out of a maximum of 30. This is the average sum of the score for all the tasks performed by the user. This shows that the expert users were able to understand the problem better than novice users. This shows that the hypothesis H1 is not true.

The average score of task iv (fix identification and prioritization) for the Problem Report Widget and the Analysis Widget were 5 and 7.1 respectively. The average score shows that users found the Analytics Widget to be better than the Problem Report Widget in identifying and prioritizing the problems to be fixed. This suggests that the hypothesis H2 is true.

The average SUS usability and effectiveness scores for the Problem Report Widget and the Analysis Widget were 72.25 and 78 respectively. The average score for novice users was 74 and 79.25 for expert users. This shows that the users found Analytics Widget to be more effective and usable than the Problem Report Widget in general. Also, the expert users found the dashboard to be useful than novice users.

6 Conclusion

We have studied the extent to which an intelligent data-driven dashboard can assist the user to understand the data quality problems in Linked data and thus enable them to identify the fixes was investigated. User experiments were performed to investigate i) if the users were able to understand the identified problems better using the widgets in the new dashboard, ii) which widget was better in assisting the user to identify and prioritize the problems that need to be fixed. The overall result of the experiment was to find that the dashboard proved to be useful to the users but that expert users were better supported that non-experts.

Future work will improve the dashboard with additional reporting and machine learning algorithms to further automate the root cause analysis of the problems.

Acknowledgement. This research has received funding from the ADAPT Centre for Digital Content Technology, funded under the SFI Research Centres Programme (Grant 13/RC/2106) and co-funded by the European Regional Development Fund.

References

1. G. K. Tayi and D. P. Ballou, "Examining data quality," Communications of the ACM, vol. 41, pp. 54-57, 1998.
2. C. Cappiello, C. Francalanci and B. Pernici, "Data quality assessment from the user's perspective," in Proceedings of the 2004 international workshop on Information quality in information systems, 2004.
3. A. Zaveri, A. Rula, A. Maurino, R. Pietrobon, J. Lehmann and S. Auer, "Quality assessment for linked data: A survey," Semantic Web, vol. 7, pp. 63-93, 2016.
4. J. Debattista, C. Lange and S. Auer, "Representing Dataset Quality Metadata using Multi-Dimensional Views.," CoRR, vol. abs/1408.2468, 2014.
5. C. Bizer, T. Heath and T. Berners-Lee, "Linked data - the story so far," Int. J. Semantic Web Inf. Syst., vol. 5, p. 1–22, 2009.
6. E. Miller, "An introduction to the resource description framework," Bulletin of the American Society for Information Science and Technology, vol. 25, pp. 15-19, 1998.
7. B. McBride, "The resource description framework (RDF) and its vocabulary description language RDFS," in Handbook on ontologies, Springer, 2004, pp. 51-65.
8. D. L. McGuinness, F. Van Harmelen and others, "OWL web ontology language overview," W3C recommendation, vol. 10, p. 2004, 2004.
9. J. J. Rooney and L. N. V. Heuvel, "Root cause analysis for beginners," Quality progress, vol. 37, pp. 45-56, 2004.
10. A. M. Doggett, "A statistical comparison of three root cause analysis tools," Journal of Industrial Technology, vol. 20, pp. 2-9, 2004.
11. A. M. Doggett, "Root cause analysis: a framework for tool selection," Quality Management Journal, vol. 12, pp. 34-45, 2005.
12. C. Guéret, P. Groth, C. Stadler and J. Lehmann, "Assessing linked data mappings using network measures," in Extended semantic web conference, 2012.
13. P. N. Mendes, H. Mühleisen and C. Bizer, "Sieve: linked data quality assessment and fusion," in Proceedings of the 2012 Joint EDBT/ICDT Workshops, 2012.
14. D. Kontokostas, P. Westphal, S. Auer, S. Hellmann, J. Lehmann, R. Cornelissen and A. Zaveri, "Test-driven evaluation of linked data quality," in Proceedings of the 23rd international conference on World Wide Web, 2014.
15. D. Kontokostas, A. Zaveri, S. Auer and J. Lehmann, "Triplecheckmate: A tool for crowdsourcing the quality assessment of linked data," in International Conference on Knowledge Engineering and the Semantic Web, 2013.
16. J. Debattista, S. Auer and C. Lange, "Luzzu - A Methodology and Framework for Linked Data Quality Assessment.," J. Data and Information Quality, vol. 8, pp. 4:1-4:32, 2016.