SemQuire - Assessing the Data Quality of Linked Open Data Sources based on DQV

André Langer[0000−0001−7073−5377], Valentin Siegert[0000−0001−5763−8265], Christoph Göpfert and Martin Gaedke[0000−0002−6729−2912]

Technische Universität Chemnitz, Germany
{andre.langer,valentin.siegert,christoph.goepfert,martin.gaedke}
@informatik.tu-chemnitz.de

Abstract. The World Wide Web represents a tremendous source of knowledge, whose amount constantly increases. Open Data initiatives and the Semantic Web community have emphasized the need to publish data in a structured format based on open standards and ideally linked to other data sources. But that does not necessarily lead to error-free information and data of good quality. It would be of high relevance to have a software component that is capable of measuring the most relevant quality metrics in a generic fashion as well as rating these results.
We therefore present SemQuire, a quality assessment tool for analyzing quality aspects of particular Linked Data sources both in the Open Data context as well as in the Enterprise Data Service context. It is based on open standards such as W3C's RDF, SPARQL and DQV, and implements as a proof-of-concept a basic set of 55 recommended intrinsic, representational, contextual and accessibility quality metrics. We provide a use case for evaluating SemQuire's feasibility and effectiveness.

Keywords: Linked Data, Open Data, Semantic Web, Data Quality, Quality Assessment

1 Introduction

The hurdle-free publication of correct information enables consumers from the public and business sector to solve particular tasks based on available data. However, it is not sufficient to make a bunch of data available through the World Wide Web. Several other requirements have to be fulfilled so that information from a certain knowledge domain becomes valuable and useful for a particular usage scenario. This involves both accessibility aspects in the data retrieval step as well as intrinsic demands on the data itself.

Data Quality (DQ) is a concept describing the appropriateness of a data set based on concrete use case requirements. The examined data set is of excellent quality if it conforms to all needs and is free if defects [10] ("fitness for use" [12]). Otherwise, the quality of a data source is described as poor, if it does not meet the expectations. Quality aspects are usage dependent in general. Information from a data source can be of good quality for one intended use, and totally inappropriate for another purpose (e.g., by lacking required information). This involves requirements both on data instance level, schema level as well as on service level [7].

The analysis of data quality issues is not new and originates already in the 1970s. In Information Science, it involves the formulation of required aspects in terms of quality metrics as indicators and the test of data sets against these quality requirements. Commonly, quantitative measurements with a concrete numeric output are run in (semi-)automated processes, but qualitative analysis steps are possible as well. However, it is still controversial, which quality metrics are of major interest and if a basic set of general-purpose metrics makes sense in general. An excellent overview on this topic was recently provided in publications by Zaveri [13], Hogan [6] or Flemming [3].

Furthermore, the comparison of quality metric measurements and the overall quality assessment among multiple data sources, a series of points in time, or different quality checker tools is not trivial. Several propositions have already been made for exchanging quality measurement results. Mainly, they originate in the Semantic Web community [2] [4], resulting in a recommendation for a Data Quality Vocabulary (D3V)[1] by W3Cs Data Quality Working Group.

We have adopted these previous contributions from other authors and used it in the context of an industrial Linked Enterprise Data Services (LEDS) growth-core project for a proof-of-concept in practise. As a result, we want to present the following contributions:

- The implementation of an up-to-date implementation of a DQ Assessment Component (SemQuire) for the general analysis of structured RDF data sources that returns a machine-readable DQV export of measurement results
- A rating approach that maps each measurement value to a numeric quality assessment score for better interpretability
- The brief discussion of implementation aspects for well-accepted quality metrics

The rest of the paper is structured in the following way. Section 2 contains a more detailed description on DQ metrics and provides an overview on functional requirements for a Data Quality Assessment component. Section 3 presents the prototypical implementation of our SemQuire software component and a list of experiences during the implementation process. Section 4 analyses the correctness of our implementation based on a concrete Use Case with measurement results. In section 5, we mention recent publications of other authors in the quality assessment domain and contrast our work from existent alternative quality checkers from the past. Finally, section 6 sums up our results and contains a plan for future work.

2 Challenges in measuring Quality Metrics

Our driving research question is whether the quality state in online published data sources can be monitored in an automated fashion and compared among different data sources, assessment tools or points in time by the mean of using a set of standard quality metrics and the mapping to a rating score.

[1] https://www.w3.org/TR/vocab-dqv/

The term quality in the context of data source analysis is diffuse and encompasses aspects that go beyond a simple syntactic validation or a correctness check for the absence of contradictions and errors in local data sets. Research in the past has already focused on this challenge and multiple times investigated the different dimensions of quality. Publications like ISO/IEC 25012[2] provide a comprehensive overview and definitions for common and generally accepted metrics and try to classify and cluster the metrics in a more general scheme. We base our research on the data quality dimensions and their categorization identified in a systematic literature review by [13]. They suggest a classification of these metrics and corresponding indicators into four primary groups entitled with Accessibility, Representational, Contextual and Intrinsic Quality aspects. The implementation of such a quality metric should be possible straight-forward according to their unambigious conceptual description in the corresponding literature.

Stakeholders with potential interest on quality measurement results can be found both on data publication as well as on data consumption side. A data curator or service provider of a data portal is interested to publish correct data in a useful way. Data consumers on the contrary are interested to find data sources that fit test to their current needs. As a consequence, measurements can be run from all stakeholder groups on all available resources and data service endpoints. These measurement results can then be published as meta data in a machine-readable format for further processing and comparison activities.

In order to do that, analyzed data quality metrics should be stated in an unambiguous and referenceable fashion. The data quality vocabulary (DQV) therefore introduces a set of properties to announce quality measurement results. To identify particular quality aspects, URIs are used as a reference. It is intentionally not the objective of the W3C working group "to define a normative list of dimensions and metrics"[3], thus they only state some basic examples. However, it is also mentioned that "relying on existing classifications and metrics increases interoperability" which symbolizes a valuable intension for Open Data exchange. (A similar approach is followed in the Linked Data community to reference particular existing entities with URIs e.g., in the DBpedia project[4], though it does not contain entries for abstract concepts such as data metrics yet). We therefore put in front in the following a list of potential quality metrics together with a recommended URI in table 1. Be aware, that we currently do not focus on metrics of a limited application domain, metrics with already profound tool support or metrics involving sophisticated data mining or AI methodologies.

We pose the following requirements on a software tool that should be capable of measuring the mentioned quality metrics:

RQ1 It can be applied on data sets containing structured data in an RDF serialization format (unstructured or semi-structured data sources can be processed to some extend using document converters in advance[5])

[2] http://iso25000.com/index.php/en/iso-25000-standards/iso-25012

[3] https://www.w3.org/TR/vocab-dqv/#DimentsionsMetricsHints

[4] https://dbpedia.org

[5] https://www.w3.org/wiki/ConverterToRdf#Frameworks

Accessibility metrics	
(01)	http://dataconcepts.net/metrics/quality/**AuthenticityMetric**
(02)	http://dataconcepts.net/metrics/quality/**DereferencedBacklinksMetric**
(03)	http://dataconcepts.net/metrics/quality/**DereferencedForwardLinksMetric**
(04)	http://dataconcepts.net/metrics/quality/**DigitalSignatureMetric**
(05)	http://dataconcepts.net/metrics/quality/**DumpDownloadAvailableMetric**
(06)	http://dataconcepts.net/metrics/quality/**ExternalLinksMetric**
(07)	http://dataconcepts.net/metrics/quality/**HighThroughputMetric**
(08)	http://dataconcepts.net/metrics/quality/**HumanReadableLicenseMetric**
(09)	http://dataconcepts.net/metrics/quality/**LowLatencyMetric**
(10)	http://dataconcepts.net/metrics/quality/**MachineReadableLicenseMetric**
(11)	http://dataconcepts.net/metrics/quality/**NoMisreportedContentTypeMetric**
(12)	http://dataconcepts.net/metrics/quality/**SPARQLAccessibilityMetric**
(13)	http://dataconcepts.net/metrics/quality/**URIDereferenceabilityMetric**
(14)	http://dataconcepts.net/metrics/quality/**ScalabilityMetric**
(15)	http://dataconcepts.net/metrics/quality/**SlashURIMetric**
Contextual metrics	
(16)	http://dataconcepts.net/metrics/quality/**CommunicationChannelMetric**
(17)	http://dataconcepts.net/metrics/quality/**ContentTrustMetric**
(18)	http://dataconcepts.net/metrics/quality/**CurrencyFreshnessMetric**
(19)	http://dataconcepts.net/metrics/quality/**DatasetFreshnessMetric**
(20)	http://dataconcepts.net/metrics/quality/**ExampleSPARQLQueryMetric**
(21)	http://dataconcepts.net/metrics/quality/**HumanReadableLabelsMetric**
(22)	http://dataconcepts.net/metrics/quality/**ProviderTrustworthinessMetric**
(23)	http://dataconcepts.net/metrics/quality/**ReasoningTrustworthinessMetric**
(24)	http://dataconcepts.net/metrics/quality/**ReputationMetric**
(25)	http://dataconcepts.net/metrics/quality/**ResourceTrustworthinessMetric**
(26)	http://dataconcepts.net/metrics/quality/**StatementDatasetRuleTrustworthinessMetric**
(27)	http://dataconcepts.net/metrics/quality/**StatementTrustworthinessMetric**
(28)	http://dataconcepts.net/metrics/quality/**URIExamplePatternMetric**
(29)	http://dataconcepts.net/metrics/quality/**URIRegExPatternMetric**
(30)	http://dataconcepts.net/metrics/quality/**VocabularyIndicationMetric**
Intrinsic metrics	
(31)	http://dataconcepts.net/metrics/quality/**CorrectDomainRangeDefinitionMetric**
(32)	http://dataconcepts.net/metrics/quality/**DatatypeOrObjectPropertyMisuseMetric**
(33)	http://dataconcepts.net/metrics/quality/**DeprecatedMisuseMetric**
(34)	http://dataconcepts.net/metrics/quality/**EntityAsDisjointClassMembersMetric**
(35)	http://dataconcepts.net/metrics/quality/**HighExtensionalConcisenessMetric**
(36)	http://dataconcepts.net/metrics/quality/**HighIntensionalMetric**
(37)	http://dataconcepts.net/metrics/quality/**InterlinkingCompletenessMetric**
(38)	http://dataconcepts.net/metrics/quality/**InverseFunctionalPropertyUseMetric**
(39)	http://dataconcepts.net/metrics/quality/**MisplacedClassesOrPropertiesMetric**
(40)	http://dataconcepts.net/metrics/quality/**NoMalformedDatatypeLiteralsMetric**
(41)	http://dataconcepts.net/metrics/quality/**NoRDFSyntaxErrorMetric**
(42)	http://dataconcepts.net/metrics/quality/**OntologyHijackingMetric**
(43)	http://dataconcepts.net/metrics/quality/**PopulationCompletenessMetric**
(44)	http://dataconcepts.net/metrics/quality/**PropertyCompletenessMetric**
(45)	http://dataconcepts.net/metrics/quality/**SchemaCompletenessMetric**
(46)	http://dataconcepts.net/metrics/quality/**SyntacticAccurateValuesMetric**
Representational metrics	
(47)	http://dataconcepts.net/metrics/quality/**BlankNodesMetric**
(48)	http://dataconcepts.net/metrics/quality/**DataInterpretabilityMetric**
(49)	http://dataconcepts.net/metrics/quality/**ProlixRDFFeaturesMetric**
(50)	http://dataconcepts.net/metrics/quality/**ReusedVocabularyMetric**
(51)	http://dataconcepts.net/metrics/quality/**SelfDescriptiveFormatMetric**
(52)	http://dataconcepts.net/metrics/quality/**SerializationFormatMetric**
(53)	http://dataconcepts.net/metrics/quality/**ShortURIMetric**
(54)	http://dataconcepts.net/metrics/quality/**UndefinedClassPropertyUsageMetric**
(55)	http://dataconcepts.net/metrics/quality/**VariousLanguageMetric**

Table 1: In SemQuire implemented DQ metrics with referenceable LD URI

RQ2 Input data can be specified in a push (direct input,upload) and/or pull (fetch from url, fetch from SPARQL endpoint) manner

RQ3 Relevant metrics that should be measured can be selected in advance from a list of available implemented metrics

RQ4 If metrics depend or relate to each other, any dependencies should be resolved during calculation without remeasuring duplicate aspects

RQ5 The measurement assignment as well as the metrics should be referenceable by using a persistent URI

RQ6 A measurement report should be generated after finishing all measurements containing concrete measurement values

RQ7 The measurement report should be exportable in a machine-readable format, preferably using DQV

RQ8 Optionally, an overall quality assessment score should be calculated with ratings for each measurement result

RQ9 Optionally, the current measurement should be comparable with other quality measurements

RQ10 The software tool should provide a Web UI for human interaction and presentation as well as a service backend for automation purposes and bulk processing

A conceptual program flow for fulfilling these requirements is briefly depicted in fig. 1.

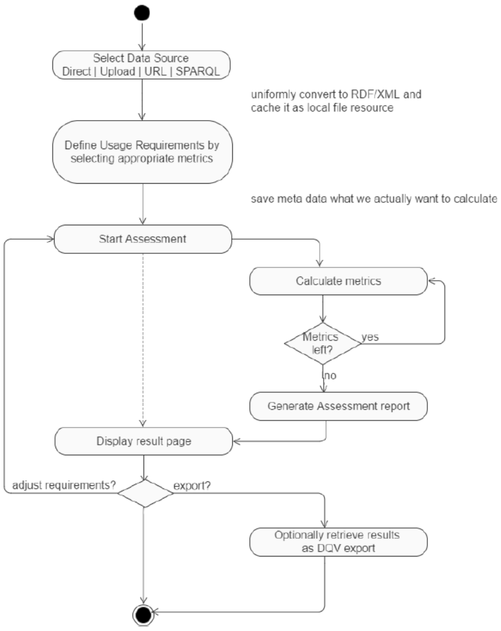

Select Data Source
Direct | Upload | URL | SPARQL
uniformly convert to RDF/XML and
cache it as local file resource
Define Usage Requirements by
selecting appropriate metrics
save meta data what we actually want to calculate
Start Assessment
Calculate metrics
Metrics
left?
yes
no
Generate Assessment report
Display result page
adjust requirements?
export?
Optionally retrieve results
as DQV export

Fig. 1: Activity diagram for a data quality assessment tool

3 The SemQuire approach

In the following, we present SemQuire, a practical engineering approach for the data quality assessment of structured data sources. SemQuire is a result of the German Linked Enterprise Data Services (LEDS) growth-core project. The primary objective of the LEDS project is to build a novel, future-proof technology platform that is capable of combining, extending and enriching corporate data stores with external, open-available data. One of the most critical aspects in this concept is the (automated) assurance of certain quality requirements in the process of knowledge combination. Open Data Services often provide hereby an inhomogeneous variety of data structures ranging from very detailed, conscientiously curated data collections with a very high number of corresponding properties down to data providers with only little information value.

The SemQuire application consists out of four main components:

- A WebGUI for enabling human users to manually check particular data sets for quality issues, relying on Googles MDL front-end template library
- A RESTful web service API for machine-to-machine interaction, currently implemented in NodeJS with TypeScript Transpiling
- A set of implemented metrics that is easily extensible, mainly based on rdflib and other Python libraries
- A graph database, currently using Stardog, accessed via an industrial data middleware (eccenca DataPlatform)

The entire system architecture is depicted in fig. 2 and deployed in a Docker container. In contrast to other previously existing quality checker tools, SemQuire is to the best of our knowledge the first that allows the machine-readable export of all measurement results in DQV, follows a rating concept for all quality measurements and calculates a comparable overall assessment score. The SemQuire component can be publicly accessed via `https://goo.gl/nYv9sX` for demonstration purposes. Figure 3 depicts screenshots of the SemQuire prototype.

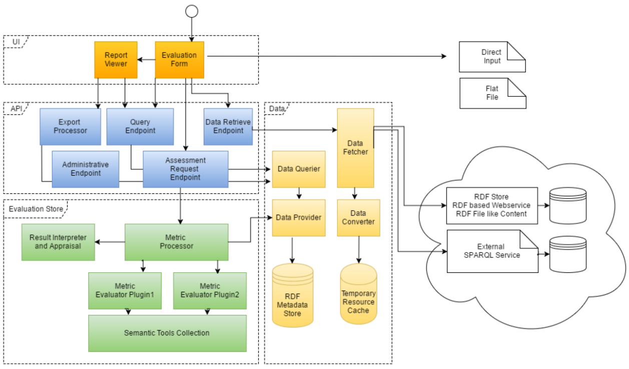
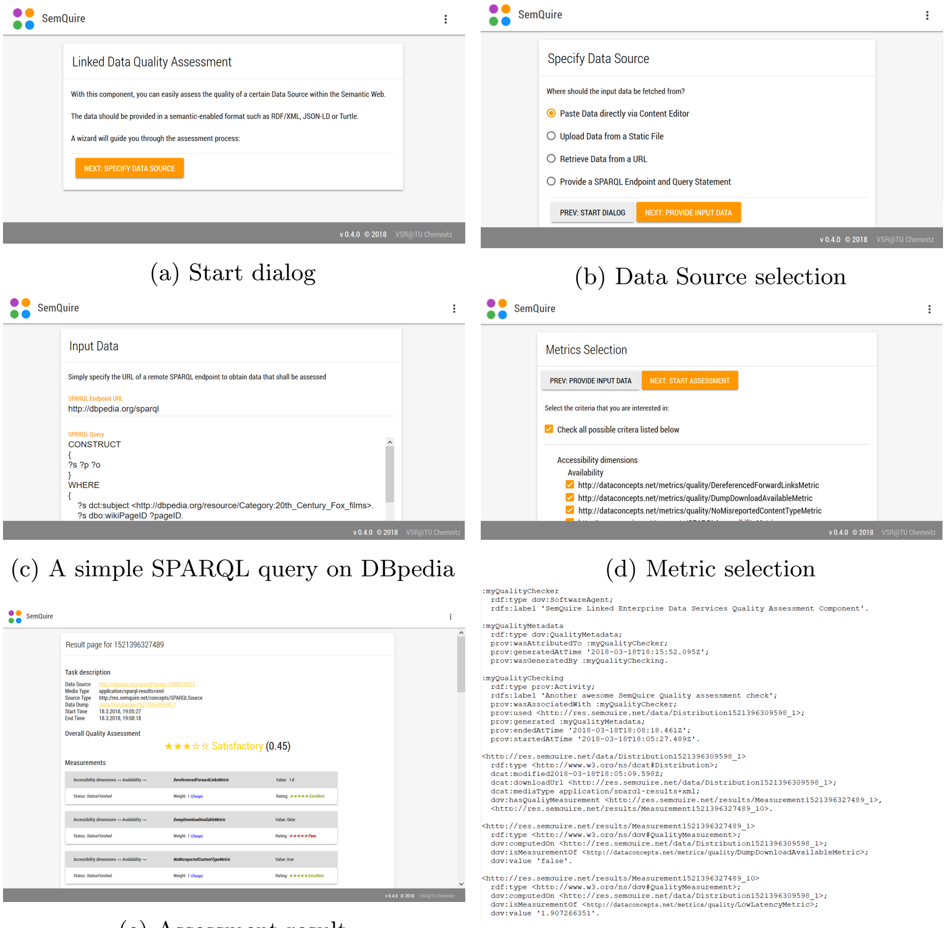


Fig. 2: Components of the SemQuire quality assessment tool



SemQuire

Linked Data Quality Assessment

With this component, you can easily assess the quality of a certain Data Source within the Semantic Web.

The data should be provided in a semantic-enabled format such as RDF/XML, JSON-LD or Turtle.

A wizard will guide you through the assessment process:

NEXT: SPECIFY DATA SOURCE

v 0.4.0 © 2018 VSR@TU Chemnitz

(a) Start dialog

SemQuire

Specify Data Source

Where should the input data be fetched from?

● Paste Data directly via Content Editor

○ Upload Data from a Static File

○ Retrieve Data from a URL

○ Provide a SPARQL Endpoint and Query Statement

PREV: START DIALOG | NEXT: PROVIDE INPUT DATA

v 0.4.0 © 2018 VSR@TU Chemnitz

(b) Data Source selection

SemQuire

Input Data

Simply specify the URL of a remote SPARQL endpoint to obtain data that shall be assessed

SPARQL Endpoint URL
http://dbpedia.org/sparql

SPARQL Query

```
CONSTRUCT
{
?s ?p ?o
}
WHERE
{
   ?s dct:subject <http://dbpedia.org/resource/Category:20th_Century_Fox_films>.
   ?s dbo:wikiPageID ?pageID.
```

v 0.4.0 © 2018 VSR@TU Chemnitz

(c) A simple SPARQL query on DBpedia

SemQuire

Metrics Selection

PREV: PROVIDE INPUT DATA | NEXT: START ASSESSMENT

Select the criteria that you are interested in:

☑ Check all possible critera listed below

Accessibility dimensions
Availability
☑ http://dataconcepts.net/metrics/quality/DereferencedForwardLinksMetric
☑ http://dataconcepts.net/metrics/quality/DumpDownloadAvailableMetric
☑ http://dataconcepts.net/metrics/quality/NoMisreportedContentTypeMetric

v 0.4.0 © 2018 VSR@TU Chemnitz

(d) Metric selection

SemQuire

Result page for 1521396327489

Task description

Data Source	http://dbpedia.org/sparql?query=CONSTRUCT...
Media Type	application/sparql-results+xml
Source Type	http://res.semquire.net/concepts/SPARQLSource
Data Dump	/data/Distribution1521396309598_1
Start Time	18.3.2018, 19:05:27
End Time	18.3.2018, 19:08:18

Overall Quality Assessment

★★★☆☆ Satisfactory (0.45)

Measurements

Accessibility dimensions → Availability →	*DereferencedForwardLinksMetric*	Value: *1.0*
Status: *StatusFinished*	Weight: *1* (Change)	Rating: ★★★★★ *Excellent*
Accessibility dimensions → Availability →	*DumpDownloadAvailableMetric*	Value: *false*
Status: *StatusFinished*	Weight: *1* (Change)	Rating: ☆☆☆☆☆ *Poor*
Accessibility dimensions → Availability →	*NoMisreportedContentTypeMetric*	Value: *true*
Status: *StatusFinished*	Weight: *1* (Change)	Rating: ★★★★★ *Excellent*

v 0.4.0 © 2018 VSR@TU Chemnitz

(e) Assessment result

```
:myQualityChecker
  rdf:type dqv:SoftwareAgent;
  rdfs:label 'SemQuire Linked Enterprise Data Services Quality Assessment Component'.

:myQualityMetadata
  rdf:type dqv:QualityMetadata;
  prov:wasAttributedTo :myQualityChecker;
  prov:generatedAtTime '2018-03-18T18:15:52.095Z';
  prov:wasGeneratedBy :myQualityChecking.

:myQualityChecking
  rdf:type prov:Activity;
  rdfs:label 'Another awesome SemQuire Quality assessment check';
  prov:wasAssociatedWith :myQualityChecker;
  prov:used <http://res.semquire.net/data/Distribution1521396309598_1>;
  prov:generated :myQualityMetadata;
  prov:endedAtTime '2018-03-18T18:08:18.461Z';
  prov:startedAtTime '2018-03-18T18:05:27.489Z'.

<http://res.semquire.net/data/Distribution1521396309598_1>
  rdf:type <http://www.w3.org/ns/dcat#Distribution>;
  dcat:modified2018-03-18T18:05:09.598Z;
  dcat:downloadUrl <http://res.semquire.net/data/Distribution1521396309598_1>;
  dcat:mediaType application/sparql-results+xml;
  dqv:hasQualiyMeasurement <http://res.semquire.net/results/Measurement1521396327489_1>,
  <http://res.semquire.net/results/Measurement1521396327489_10>.

<http://res.semquire.net/results/Measurement1521396327489_1>
  rdf:type <http://www.w3.org/ns/dqv#QualityMeasurement>;
  dqv:computedOn <http://res.semquire.net/data/Distribution1521396309598_1>;
  dqv:isMeasurementOf <http://dataconcepts.net/metrics/quality/DumpDownloadAvailableMetric>;
  dqv:value 'false'.

<http://res.semquire.net/results/Measurement1521396327489_10>
  rdf:type <http://www.w3.org/ns/dqv#QualityMeasurement>;
  dqv:computedOn <http://res.semquire.net/data/Distribution1521396309598_1>;
  dqv:isMeasurementOf <http://dataconcepts.net/metrics/quality/LowLatencyMetric>;
  dqv:value '1.907266351'.
```

(f) DQV export

Fig. 3: SemQuire WebUI screenshots

We implemented a set of common quality metrics from multiple quality groups (see table 1) dealing with different views on a data source.

Metrics from the Accessibility group deal with technical data access aspects. Some of them are not applicable to data sets that are provided in a push manner to the system by the user (e.g., a file upload of a data dump or directly by pasting the data content), and refer to remote URLs or SPARQL endpoint concerns such as *Latency*, *Scalability*, *Throughput* or *SPARQLAccessibility*. Others evaluate meta data contained in the document itself or in retrievable well-known access paths such as *License information*, the *Availability of a Dump download*, *Digital Signiture* or *appropriate ContentType information*. Another dimension checks contained external URIs in the retrieved data set for dereferenceability. Especially the execution of the ladder metrics can become time-consuming for large documents with an increased number of URIs.

A second group dealt with representational aspects of the provided data. We implemented metrics, that check if the same data can be retrieved in different RDF serialization formats, if well-known vocabularies are reused, and if the usage of constructs like BlankNodes or other prolix RDF features is avoided. The usage of ShortURIs might also be seen as an intrinsic aspect and are subject for discussion regarding the char length of a concept representation. From our experience during implementation, this can be use case and domain dependent. As other publications did not state a recommended explicit maximum length for a short URI, we used 80 chars as a general threshold.

In the following, we were interested in analyzing general intrinsic quality aspects of open accessible structured data. After checking the general validity with a respective validator, SemQuire converts them internally uniformly into RDF/XML. Next, either traditional RDF validators can be applied or more sophisticated third-party tools such as RDFAlerts [5]. In order to check other intrinsic dimensions such as consistency, completeness and conciseness metrics, it is first of all necessary to retrieve schema information on the used ontologies in the document. Dereferencing all used namespaces within one document is one possible, flexible automated approach. However, still not all ontology description sites offer a machine-readable version of the vocabulary. Completeness checks provide another challenge for a quality checker by requiring additional background knowledge (”gold standard”). Obviously, this is hard to achieve for certain application domains under an Open World Assumption for distributed data. Additionally, a comparison based on literal values is not practical useful for different languages or spellings. Instead, a completeness check based on entity URIs is more valuable. However, it also has to consider owl:sameAs relationships for similar concepts identified under different URI domain names. SemQuire checks all intrinsic metrics based on available document from the current and linked documents.

In contrast, contextual metrics require an additional usage context for the concrete application scenario by the user. For some contextual dimensions such as timeliness or understandability, simple parameter inputs can be requested by the system or even meaningful standard values can be applied statically. Checking relevancy needs a complex contextual input to satisfy the metric on a high level. Assessing trust either needs kinds of black- or whitelists, an authority or also a complex contextual input. Provenance data can hereby also be an input regarding some trust metrics. To circumvent a complex input, the PageRank approach can be used to return a initialization regarding the relevance and the more detailed trust metric about content trust. Such an initialization will still not behalf as a high-end trust network or description of relevance, but gives the contextual metrics a kick-off in the right direction. Solving a contextual metric with crowd-sourcing seems not to fit for us, as each human brings in his own bias.

For all metrics of interest, each measurement result value is then mapped to a rating score, representing the fulfillment of the investigated aspect. It is a numeric value between 0.0 (not fulfilled at all) and 1.0 (perfect). All individual ratings are then linearly combined to an overall quality assessment score. Details can be found in [8].

4 Evaluation

To show the effectiveness of SemQuire, we conducted a case study and used a small example of real-world open data resources to solve a common task for evaluation purposes. In our example case, a user is interested in getting information on all existing movies in the film series of *James Bond*. We chose three different linked open data source candidates, which we queried with SemQuire, and compared later on the results. Namely, the three selected providers were *DBpedia*[6], *Wikidata*[7] and *LinkedMDB*[8].

Therefore, we designed three different SPARQL queries to obtain with SemQuire all information about movies of the *James Bond* film series. The queries differ mainly in the used vocabularies for each data provider, but the semantic is always the same as we search for all *James Bond* films and their outgoing relations or properties.

Not all offered metrics by SemQuire are relevant for the test case, so we carefully selected only metrics that help in the assessment process of finding the most appropriate data source for solving the task. The metrics were chosen by either importance for the test case or based on interesting differences in the results and ratings of SemQuire. Hence, we will show in the following differences between the data provider candidates according to the scenario with respect to six metrics and the underlying data. The corresponding measurement' ratings are shown in table 2. Additionally, we provide the numbers of returned triples (T#) as a statistical meta info for better understanding. Two metrics' results are further shown in fig. 4 for contrasting purposes of results and ratings in SemQuire.

Population Completeness (PopulComp) Regarding the test case of gathering all *James Bond* films, it is important if the endpoints really return all relevant movies, thus have a population completeness of *100%*. Surprisingly, metric (43) shows that *LinkedMDB* is not referencing all *James Bond* films, but only *48%*. It could be the case that *LinkedMDB* is not referencing all *James Bond* films to the category about *James Bond* films, which would explain this low percentage.

Serialization Format (SeriForm) As the test case does not explicitly specify how the data will be used, it can be very interesting for further processing to have the possibility of retrieving different serialization formats. Metric (52) measures in how many formats the data can be provided. SemQuire indicates that only *DBpedia* is able to provide more than one, so more than the standard RDF/XML format with content negotiation.

Various Languages (VarLang) Beyond the processing of the data, the data might also be shown to humans and thus it can be important that various languages are included in the data set. As our queries are not filtering on any language, metric (55) is able to check if there are various languages or not in the underlying data. *LinkedMDB* is again beyond the two others, as it is only providing the information in one language.

[6] `http://dbpedia.org/`

[7] `http://wikidata.org/`

[8] `http://linkedmdb.org/`

URI Dereferenceability (URIDeref) The metric (13) about dereferenceability of the URIs is relevant for the evaluation, as the importance of SemQuire's mapping approach from absolute values to normalized ratings can be seen. The results of this metric depict the count of all dereferenceable URIs within the data. All endpoints provide a different amount of triples, and thus there are also differences in the results. On the contrary, the ratings of this metric show that the difference between the three endpoints is not even relevant, as they are for all pretty good and close. The rating is hereby created with respect to the overall triple numbers of the data, and is thus more significant than the results.

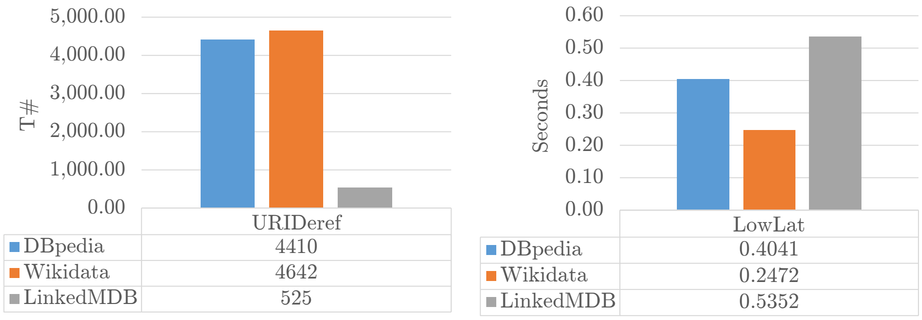
External Links (ExternL) With regard to an open world model, one endpoint is often not able to provide all information within its domain. The metric (06) is checking whether the provided data includes a link to external data outside the data endpoint domain. Interestingly, only DBpedia provides external links to other domains.

Low Latency (LowLat) The advantages of a low latency for one request to the endpoint can be important at tasks with a time factor that often live-update their data. The test case is not necessarily referring to a need of low latency, but the metric (09) is still interesting for a general QoS rating of the endpoint and a possible extension of the test scenario. The results are quite different, but the rating again gives an idea on how good these results are.

Based on the results and ratings of the six metrics, a decision upon which endpoint should be used, is required (which involved human interaction in the past). We use SemQuire's possibility to combine the discussed measurements to an overall quality assessment value (Score). The resulting order is depicted in table 2, the recommended endpoint to choose is consequently in our test case *DBpedia*.

Endpoints	Score	T#	URIDeref	ExternL	LowLat	PopulComp	SeriForm	VarLang
DBpedia	0.97	10001	0.9802	1	0.8653	1	1	1
Wikidata	0.65	9626	0.9996	0	0.9176	1	1	0
LinkedMDB	0.38	724	0.9795	0	0.8216	0.48	0	0

Table 2: SemQuire's ratings, score and T# for each endpoint



(a) URI Dereferenceability Results

(b) Low Latency Results

Fig. 4: URIDeeref & LowLat Results

5 Related Work

Examples for vocabularies to describe data and service quality from the Semantic Web community are the daQ [2], DQM vocabulary [4] or the current W3C draft for a data quality vocabulary (DQV)[9]. Furthermore, several data quality checker implementations already existed in the past. They differ on various characteristics such as functionality, processable data format, implementation language, user interface or result output manner. Examples are Diachron [13], KBMetrics [11], LDSrcAss [3], Luzzu [2], RDFAlerts [5], Roomba OpenData Checker [1], Sieve [9] or SWIQA [4] . Some of them only focused on a limited use case or are not publicly available any longer. Moreover, assessment results were often provided in different output formats and not comparable to each other. For instance, the OpenData Checker calculated metrics from data quality indicators specifically for CKAN data stores and simply outputed them in percent. KBMetrics used a scoring system to make different data sources comparable. SWIQA calculated a quality score based on the percentage how many instances violate given data quality rules. Emphasis has therefore been placed on the requirement to make quality measurements comparable by using semantic means. Table 3 contrasts all mentioned software tools based on the original usage requirements we posed in section 2. Currently, SemQuire is the only tool that satisfies all defined requirements.

Criterion	Diachron	KBMetrics	LDSrcAss	Luzzu	RDFAlerts	Roomba	SemQuire	LDIF/Sieve	SWIQA
Available	y	n	n	y	y	y	y	y	n
Language	Java	?	?	Java	Java	NodeJS	NodeJS	Java	?
RQ1	y	y	y	y	y	(y)	y	y	y
RQ2	n	n	n	n	n	n	y	y	n
RQ3	y	y	(y)	y	n	(y)	y	y	y
RQ4	n	n	n	y	?	n	y	n	n
RQ5	y	n	n	(y)	(y)	n	y	n	n
RQ6	y	y	y	y	y	y	y	y	y
RQ7	n	n	n	y	n	(y)	y	n	?
RQ8	n	y	y	y	n	n	y	n	y
RQ9	n	y	n	y	n	n	y	n	n
RQ10	y	y	y	y	y	n	y	n	?

Table 3: Comparison of quality assessment tools wrt. requirements from section 2

6 Conclusion

In this paper, we presented SemQuire a practical implementation of a quality assessment component that can be used as a toolkit to measure and assure the quality of open or enterprise data sources that expose information in a common RDF serialization format. SemQuire relies on the theoretical findings of previously published surveys dealing with most relevant quality metrics. It implements 55 of the most common quality indicators. In advance, we conducted a brief market overview and compared other existing tools with our component with the result, that there is currently, to the best of our knowledge, no other software component available that fulfills all requirements of interest.

[9] `https://www.w3.org/TR/vocab-dqv/`

Acknowledgment. This work was supported by the grant from the German Federal Ministry of Education and Research (BMBF) for the LEDS Project under grant agreement No 03WKCG11D.

References

1. Ahmad, A.A., Troncy, R., Senart, A.: Roomba: An extensible framework to validate and build dataset profiles. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). vol. 9341, pp. 325–339 (2015)
2. Debattista, J., Lange, C., Auer, S.: Daq, an ontology for dataset quality information. In: CEUR Workshop Proceedings. vol. 1184 (2014)
3. Flemming, A.: Qualitätsmerkmale von Linked Data-veröffentlichenden Datenquellen pp. 1–174 (2011), http://www.dbis.informatik.hu-berlin.de/fileadmin/research/papers/diploma_seminar_thesis/Diplomarbeit_Annika_Flemming.pdf
4. Fürber, C., Hepp, M.: Towards a vocabulary for data quality management in semantic web architectures. Proceedings of the 1st International Workshop on Linked Web Data Management - LWDM '11 p. 1 (2011)
5. Hogan, A., Harth, A., Passant, A., Decker, S., Polleres, A.: Weaving the pedantic Web. In: CEUR Workshop Proceedings. vol. 628 (2010)
6. Hogan, A., Umbrich, J., Harth, A., et al.: An empirical survey of linked data conformance. Web Semant. 14, 14–44 (Jul 2012)
7. Langer, A., Gaedke, M.: Fame.q -a formal approach to master quality in enterprise linked data. In: Proceedings of the 15th International Conference WWW/Internet (ICWI2016). pp. 51–58. IADIS (October 2016)
8. Langer, A., Gaedke, M.: Daqar - an ontology for the uniform exchange of comparable linked data quality assessment requirements (June 2018), to appear in: Proceedings of the 18th International Conference on Web Engineering (ICWE2018)
9. Mendes, P.N., Mühleisen, H., Bizer, C.: Sieve: Linked data quality assessment and fusion. In: Proceedings of the 2012 Joint EDBT/ICDT Workshops. pp. 116–123. EDBT-ICDT '12, ACM, New York, NY, USA (2012)
10. Redman, T.C.: Data Quality: The Field Guide. Digital Press, Newton, MA, USA (2001)
11. Ruan, T., Dong, X., Li, Y., Wang, H.: KBMetrics A Multi-purpose Tool for Measuring the Quality of Linked Open Data Sets (2015)
12. Wang, R. Y., and Strong, D.M.: Beyond accuracy: What data quality means to data consumers. Journal of Management Information Systems. Journal of Management Information Systems 12(4), 5–33 (1996)
13. Zaveri, A., Rula, A., Maurino, A., et al.: Quality Assessment for Linked Open Data: A Survey. Semantic Web Journal (by IOS Press) 1, 1–31 (2014)