Assessing and Refining Mappings to RDF to Improve Dataset Quality

Anastasia Dimou[1], Dimitris Kontokostas[2], Markus Freudenberg[2], Ruben Verborgh[1], Jens Lehmann[2], Erik Mannens[1], Sebastian Hellmann[2], and Rik Van de Walle[1]

[1] Ghent University - iMinds - Multimedia Lab, Belgium
{firstname.lastname}@ugent.be
[2] Universitat Leipzig, Institut fur Informatik, AKSW, Germany
{lastname}@informatik.uni-leipzig.de

Abstract. RDF dataset quality assessment is currently performed primarily after data is published. However, there is neither a systematic way to incorporate its results into the dataset nor the assessment to the publishing workflow. Adjustments are manually—but rarely—applied. Nevertheless, the root of the violations which often derive from the mappings that specify how the RDF dataset will be generated, is not identified. We suggest an *incremental, iterative and uniform validation workflow* for RDF datasets stemming originally from semi-structured data (e.g., CSV, XML, JSON). In this work, we focus on assessing and improving their mappings. We incorporate i) a *test-driven approach for assessing the mappings* instead of the RDF dataset itself, as mappings reflect how the dataset will be formed when generated; and ii) perform *semi-automatic mapping refinements* based on the results of the quality assessment. The proposed workflow is applied to different cases, e.g., large, crowdsourced datasets as DBpedia, or newly generated, as iLastic. Our evaluation indicates the efficiency of our workflow, as it improves significantly the overall quality of an RDF dataset in the observed cases.

Keywords: Linked Data Mapping, Data Quality, RML, R2RML, RDFUnit

1 Introduction

The Linked Open Data (LOD) cloud[3] consisted of 12 datasets in 2007, grew to almost 300 in 2011[4], and, by the end of 2014, counted up to 1,100[5]. Although more and more data is published as Linked Data (LD), the datasets' quality and consistency varies significantly, ranging from expensively curated to relatively low quality datasets [28]. In previous work [20], we observed that similar violations can occur very frequently. Especially when datasets stem originally from semi-structured formats (csv, XML, etc.) and their RDF representation is obtained

[3] http://lod-cloud.net/
[4] http://lod-cloud.net/state
[5] http://linkeddatacatalog.dws.informatik.uni-mannheim.de/state/

by repetitively applying certain mappings, the violations are often repeated, as well. By *mapping*, we consider the function of semantically annotating data to acquire their enriched representation using the RDF data model. A mapping consists of one or more *mapping definitions* (MDs) that state how RDF terms should be generated, taking into account a data fragment from an original data source, and how these terms are associated to each other and form RDF triples.

The most frequent violations are related to the dataset's schema, namely the vocabularies or ontologies used to annotate the original data [20]. In the case of semi-structured data, the dataset's schema derives from the set of classes and properties specified within the mappings. A mapping might use a single ontology or vocabulary to annotate the data, or a proprietary vocabulary can be generated as the data is annotated. Lately, combinations of different ontologies and vocabularies are often used to annotate data [27], which increases the likelihood of such violations. A violation might derive from (i) *incorrect usage of schemas* in the mapping definitions; and (ii) mistakes in the *original data source*. The second category of violations can be resolved by cleansing the data. In this work, we focus specifically on the first, which is directly related to the mapping process.

Only recently, several research efforts started focusing on formalising LD quality tracking and assessment [28]. Nevertheless, such formalisation approaches remain independent of the LD mapping and publishing process—quality assessment is not even mentioned in the best practices for publishing LD [17]. Existing quality assessment refers to already published data and is, in most cases, performed by third parties rather than data publishers. Thus, incorporating quality assessment results corresponds to incorporating a *Linked Data feedback loop*: existing LD infrastructures still neither intuitively process end-users' input, nor properly propagate the modifications to the MDs and original data. Consequently, the results are rarely and, if so, manually used to adjust the dataset, with the risk of being overwritten when a new version of the original data is published.

In this paper, we therefore propose a methodology that extends LD quality assessment from data *consumption* to also cover data *publication*. We transform the assessment process normally applied to the final dataset so that it applies to the mappings as well. This allows publishers to discover mistakes in mapped RDF data—before they are even generated. Our methodology (i) augments the mapping and publishing workflow of semi-structured source formats with *systematic Linked Data quality assessments* for both the mappings and the resulting dataset; and (ii) *automatically suggests mapping refinements* based on the results of these quality assessments. We consider iterative, uniform, and gradual test-driven LD quality assessments to improve the dataset's overall quality.

The paper is organized as follows: Section 2 details the need of quality assessment during the mapping process, followed by the introduction of a mapping workflow with quality assessment in Section 3. Next, Section 4 explains how quality assessments are applied to mappings, the results of which are used to refine mapping definitions in Section 5. Section 6 highlights different cases where the proposed workflow was used, followed by an evaluation in Section 7. Finally, Section 8 and Section 9 summarize related solutions and conclusions.

2 Incorporating Quality in Mapping and Publishing

Data quality is commonly conceived as "fitness for use" for a certain application or use case [18]. A *data quality assessment metric, measure,* or *indicator* is a procedure for measuring a data quality dimension [3]. A *data quality assessment methodology* is defined as the process of evaluating whether a piece of data meets the information that consumers need in a specific use case [3]. In this respect, our use case is focused on the quality of the generated RDF dataset compared to the ontologies and vocabulary definitions of its schema. The uppermost goal is aiding data publishers to finally acquire a valid and high quality LD by annotating semi-structured data. We focus on the *intrinsic* dimension of data quality [28].

The earlier dataset quality is assessed, the better: we argue that mapping and publishing data can be considered software engineering tasks, and the cost of fixing a bug rises exponentially when a task progresses [4]. In software development, a common way to validate correct behaviour of a function is to accompany it by a set of unit tests. Similarly, a data mapping function can be accompanied by a set of test cases assigned to the mappings to ensure the correct generation of RDF datasets from input data. In this respect, incorporating quality assessment as part of the mapping and publishing workflow becomes essential, especially taking into account that it prevents the same violations to appear repeatedly within the dataset and over distinct entities. After all, in the mapping phase, structural adjustments can still be applied easily, since it allows us to pinpoint the origin of the violation, reducing the effort required to act upon quality assessment results.

Our approach has two main pillars: (i) *uniform quality assessment* of mapping definitions and the resulting dataset, as their quality is closely related; and (ii) *mapping definition refinements* to automatically improve mappings when problems are detected at the quality assessment.

Uniform quality assessment Instead of assessing an RDF dataset for its schema quality, we apply the quality assessment to the mapping definitions directly, *before* they are used to generate the RDF dataset. Their assessment results are correlated, since MDs specify how the dataset will be formed. For example, violations of the range of a certain property can be assessed by inspecting the corresponding MD, which defines how triples with this property are generated. Even though quality assessment of MDs can cover many violations related to vocabularies and ontologies used to annotate the data, some schema-related violations depend on how the MDs are *instantiated* on the original data. For example, a violation occurs if an object of integer datatype is instantiated with a floating-point value from the original source. Therefore, a *uniform* way of incrementally assessing the quality of the RDF dataset and the mapping definitions should cover both the mappings and the dataset.

Mapping definition refinements If violations are only corrected in the resulting dataset, they will have to be corrected every time a new version of the dataset is generated. Also, when a violation is found, it is not straightforward

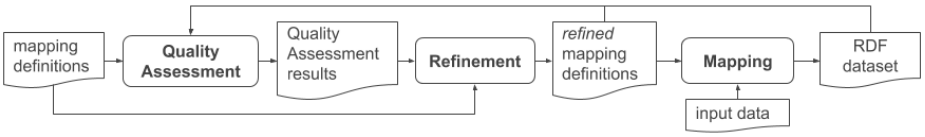

Fig. 1. Quality Assessment enabled Linked Data mapping and publishing workflow

to discover its cause, as the connection with the MDs and the source data is not apparent. A more effective approach is to refine the MDs that generate those triples, so the violation cannot occur in future versions. Furthermore, if the violation is associated with a MD, it can be addressed directly on the place where it occurred, and instead of having to regenerate the entire dataset, only the triples affected by the refinement need to be regenerated to correct the violation.

3 Linked Data and Mappings Assessment and Refinement

Uniform quality assessment requires that, in addition to the generated dataset, the mapping definitions themselves are also RDF triples. This way, the same RDF-based techniques can be applied. Additionally, performing (automated) refinement of MDs requires that machines can process and update them. Such direct processing of MDs is difficult if mappings are tightly coupled to the implementation, as is the case with most existing mapping solutions. In this respect, we focus on RDF-based mapping languages for stating the MDs. Below, we describe such a workflow (Section 3.1) and a solution that materializes it (Section 3.2).

3.1 Linked Data & Mappings Assessment & Refinement Workflow

We propose a *uniform, iterative, incremental assessment and refinement workflow* that produces, at the end, a high-quality RDF dataset. Its steps are explained below and presented in Fig. 1.

1. The schema, as stated in the MDs, is assessed against different quality assessment measures, as it would have been done if it was the actual dataset.
2. The Quality Assessment report lists each violation identified.
3. The Mapping Quality Assessment (MQA) results are used to refine the MDs. The MQA can be repeated until a set of MDs without violations is generated or if the MDs can not be further refined.
4. A refined version of the MDs is generated and used to execute the mapping of data – or a sample of the data.
5. The generated RDF output is assessed, using the same quality assessment framework. The Dataset –and optionally the Mapping– Quality Assessment (DQA) can be repeated until an ameliorated set of MDs is generated.
6. When the MDs are finalized, the actual mapping is performed and the RDF dataset is generated exempt of violations to the greatest possible extent.

```
1 <#Mapping> rml:logicalSource <#InputX> ;
2   rr:subjectMap [ rr:template "http://ex.com/{ID}"; rr:class foaf:Person ];
3   rr:predicateObjectMap [ rr:predicate foaf:knows;
4     rr:objectMap [ rr:parentTriplesMap <#Aquihntance> ].
5 <#Aquihntance> rml:logicalSource <#InputY> ;
6   rr:subjectMap [ rml:reference "Aquintance"; rr:termType rr:IRI; rr:class ex:Person ] ].
```

Listing 1. RML mapping definitions

3.2 Quality Assessment & Refinement with [R2]RML & RDFUnit

We provide a solution that implements the aforementioned workflow. The two main components of our solution are: the RML mapping language (Section 3.2) that uses mapping definitions expressed in RDF, a prerequisite for uniform quality assessment and automated refinements, as we discussed above, and the RDFUnit validation framework (Section 3.2) due to its associated test-case-based architecture [19].

RML. R2RML [5] is the only W3C standardised mapping language for defining mappings of data in relational databases to the RDF data model. Its extension RML [9] broadens its scope and covers also mappings from sources in different semi-structured formats, such as CSV, XML, and JSON. RML documents [9] contain rules defining how the input data will be represented in RDF. The main building blocks of RML documents are Triples Maps (Listing 1: line 1). A Triples Map defines how triples of the form (subject, predicate, object) will be generated.

A Triples Map consists of three main parts: the Logical Source, the Subject Map and zero or more Predicate-Object Maps. The Subject Map (line 2, 6) defines how unique identifiers (URIs) are generated for the resources mapped and is used as the subject of all RDF triples generated from this Triples Map. A Predicate-Object Map (line 3) consists of Predicate Maps, which define the rule that generates the triple's predicate (line 3) and Object Maps or Referencing Object Maps (line 4), which define how the triple's object is generated. The Subject Map, the Predicate Map and the Object Map are Term Maps, namely rules that generate an RDF term (an IRI, a blank node or a literal). A Term Map can be a *constant-valued term map* (line 3) that always generates the same RDF term, or a *reference-valued term map* (line 6) that is the data value of a referenced data fragment in a given Logical Source, or a *template-valued term map* (line 2) that is a valid string template that can contain referenced data fragments of a given Logical Source.

RDFUnit [20] is an RDF validation framework inspired by test-driven software development. In software, every function should be accompanied by a set of unit tests to ensure the correct behaviour of that function through time. Similarly, in RDFUnit, every vocabulary, ontology, dataset or application can be associated by a set of data quality test cases. Assigning test cases in ontologies results in tests that can be reused by datasets sharing the same ontology. This fits well to the mapping and publishing pipeline as we focus on the [R2]RML ontologies.

The test case definition language of RDFUnit is SPARQL, convenient to directly query for identifying violations. For rapid test case instantiation, a pattern-based SPARQL-Template engine is supported where the user can easily bind variables into patterns. An initial library of 17 common patterns was developed in [20, Table 1] which is now further extended.[6] RDFUnit has a *Test Auto Generator* (TAG) component. TAG searches for schema information and automatically instantiates new test cases. Schema can be in the form of RDFS or OWL axioms that RDFUnit translates into SPARQL under Closed World Assumption (CWA) and Unique Name Assumption (UNA). These TCs cover validation against: domain, range, class and property disjointness, (qualified) cardinality, (inverse) functionality, (a)symmetricity, irreflexivity and deprecation. Other schema languages such as *IBM Resource Shapes*[7] or *Description Set Profiles*[8] are also supported. RDFUnit includes support for automatic schema enrichment via DL-Learner [22] machine learning algorithms. RDFUnit can check an RDF dataset against multiple schemas but when this occurs, RDFUnit does not perform any reasoning/action to detect inconsistencies between the different schemas.

4 [R2]RML Mapping Definitions Quality Assessment

It is straightforward to process [R2]RML mapping definitions as datasets, because they have a native RDF representation and are written from the viewpoint of the generated triples. Our assessment process targets both (i) consistency validation of the mapping definitions against the R2RML and RML schema and, mainly, (ii) consistency validation and quality assessment of the dataset to be generated against the schema defined in the mapping definitions. This first point is handled directly by RDFUnit; the second point is handled by emulating the resulting RDF dataset to assess its schema conformance.

Consistency validation of the mapping definitions The validation of mapping definitions against the [R2]RML schema is directly handled by RDFUnit extending the supported OWL axioms. New RDFUnit TAGs were defined to support all OWL axioms in [R2]RML ontology, e.g., each Triples Map should have exactly one Subject Map, producing a total of 78 automatically generated test cases.

Consistency validation and quality assessment of the dataset as projected by its mapping definitions In order to assess a dataset based only on the mapping definitions that state how it is generated, we considered the same set of schema validation patterns normally applied on the RDF dataset (cf. Table 1). Nevertheless, instead of validating the predicate against the subject and object, we extract the predicate from the Predicate Map and validate it against the Term Maps that define how the subject and object will be formed. For instance, the

[6] https://github.com/AKSW/RDFUnit/blob/master/configuration/patterns.ttl
[7] http://www.w3.org/Submission/2014/SUBM-shapes-20140211/
[8] http://dublincore.org/documents/dc-dsp/

extracted predicate expects a Literal as object, but the Term Map that generates the object can be a Referencing Object Map that generates resources instead.

To achieve this, the properties and classes in the MDs are identified and their namespaces are used to retrieve the schemas and generate the test cases as if they were the actual dataset. We extended the corresponding RDFUnit test cases to apply to the MDs, adjusting the assessment queries.[9] For instance, the `WHERE` clause of the SPARQL test case that assesses a missing language is:

```
1 ?resource ?P1 ?c .
2 FILTER (lang(?c) = '')
```

In order to detect the same violation directly from a mapping definition, the `WHERE` clause of the assessment query is adjusted as follows:

```
1 ?poMap rr:predicate ?P1 ;
2        rr:objectMap ?resource .
3 ?P1 rdfs:range rdf:langString .
4 FILTER NOT EXISTS {?resource rr:language ?lang}
```

The validation is *Predicate-Map-driven* in principle. The expected value (line 3), as derived from the Predicate Map, is compared to the defined one (line 4), as derived from the corresponding Object Map. The next example is an RDFUnit SPARQL test case for assessing if the `rdf:type` of a triple's ObjectMap conforms to the `rdfs:range` definition of an object property. Applying this test case to the aforementioned MD (cf. Listing 1), a violation is registered, as `foaf:knows` has `foaf:Person` and not `ex:Person` as range – assuming the ontology does not define `ex:Person` as equivalent or subclass of `foaf:Person`.

```
1  SELECT DISTINCT ?resource WHERE {
2    ?mappingTo rr:subjectMap ?resource .
3    { ?resource rr:class ?T1 . } UNION {
4      ?mapping rr:predicateObjectMap ?classPoMap .
5      ?classPoMap rr:predicate rdf:type ;
6                  rr:objectMap/rr:constant ?T1 . }
7    ?mappingFrom rr:predicateObjectMap ?poMap .
8    ?poMap rr:predicate/rdfs:range ?T2  ;
9           rr:objectMap ?objM .
10   ?objM rr:parentTriplesMap ?mappingTo .
11   FILTER NOT EXISTS {
12    ?T2 (rdfs:subClassOf|(owl:equivalentClass|^owl:equivalentClass))* ?T1.}}
```

In order our assessment to be complete, the defined test cases cover all possible alternative ways of defining equivalent MDs that generate the same triples. For instance, the default way to generate the type for a resource is through the rr:class property in the Subject Map (e.g., line 2 of Listing 1). However, one may also define the type via a Predicate Object Map having rdf:type in its Predicate Map.

RDFUnit can annotate test cases by requesting additional variables and binding them to specific result properties. Using the example of Listing 4 we map for instance, variable `?T1` as `spin:violationValue` and variable `?T2` as the expected class. When a violation is identified, the annotations are applied and a result like the following is registered:

[9] https://github.com/AKSW/RDFUnit/blob/master/data/tests/Manual/www.w3.org/ns/r2rml/rr.tests.Manual.ttl

```
<5b7a80b8> a rut:ExtendedTestCaseResult;
  rut:testCase rutt:rr-produces-range-errors ;
  # (...) Further result annotations
  spin:violationRoot ex:objectMapX ;
  spin:violationPath rr:class ;
  spin:violationValue ex:Person ;
  rut:missingValue foaf:Person ;
  ex:erroneousPredicate  foaf:knows ;
```

However, some of the test cases normally applied to a dataset rely on the final values or refer to the complete dataset and thus, can only be validated after the mapping is performed –detected at *data-level* assessment (DQA). Such examples are (qualified) cardinality, (inverse) functionality, (a)symmetricity and irreflexivity. For example, we cannot validate an inverse functional property such as foaf:homepage without the actual values. Invalid mappings can occur as the mapping definitions are instantiated based on the input source, even though the mapping definitions appear to be valid. For instance, if the input data returns, a value like "American", instead of "http://dbpedia.org/resource/United_States", it would result in generating the URI `<American>`, which is invalid.

5 [R2]RML Refinements based on Quality Assessment

The results of Mapping Quality Assessment (MQA) can be used to suggest modifications or even automatically refine mapping definitions. The RDFUnit ontology provides multiple result representations in different formats [19],including RDF-based serialisations (rut:ExtendedTestCaseResult result type). Therefore, its results are easily processable by an agent that can automatically add and delete triples or suggest actions to the data publisher. In Table 1, we outline all examined violation patterns and indicate which Term Map should be refined and how. The suggested refinements are the minimum required actions to be taken to refine the mapping definitions, e.g., turn an Object Map to generated resources instead of literals, and serve as indicative *proof-of-concept* of the automation's feasibility.

Mapping refinements. Dealing with *range-level* violations requires different actions, depending on the value of the Object Map or Referencing Object Map. The Predicate Map is used to retrieve the property and identify its range, which is then compared to the corresponding Object Map or Referencing Object Map.

If the Predicate Map contains an *object property*, for instance, but the object is generated by a Referencing Object Map, which generates resources with type different than the predicate's range –as defined by the corresponding vocabulary or ontology, the predicate's range is added as class to the Referencing Object Map. Such a violation was reported at the example mentioned in the previous section (Section 5). Besides manual adjustments like defining `ex:Person` as equivalent or a subclass of `foaf:Person`, the statement that the Referencing Object Map type should be a `ex:Person`, can be replaced by a `foaf:Person`:

```
DEL: ex:objectMapX rr:class ex:Person .
ADD: ex:objectMapX rr:class foaf:Person.
MOD: adjust the definition of ex:Person
```

OWL axiom – Violation type	Level	Expect	Define	Automatic refinement
class disjointness	E	SbjMap	SbjMap	–
property disjointness	E	PreMap	PreMap	–
`rdfs:range` – class type	E	PreMap	(Ref)ObjMap	DEL: ObjMap ADD: PreMap domain to RefObjMap
`rdfs:range` – IRI instead of literal	E	PreMap	(Ref)ObjMap	DEL: (Ref)ObjMap ADD: ObjMap with literal termType
`rdfs:range` – literal instead of IRI	E	PreMap	ObjMap	DEL: ObjMap ADD: (Ref)ObjMap or ADD: ObjMap with IRI termType
`rdfs:range` – missing datatype	E	PreMap	(Ref)ObjMap	DEL: ObjMap ADD: ObjMap with PreMap datatype
`rdfs:range` – incorrect datatype	E	PreMap	(Ref)ObjMap	DEL: (Ref)ObjMap ADD: ObjMap with PreMap datatype
missing language	E	ObjMap	ObjMap	–
`rdfs:domain`	E	PreMap	SbjMap	ADD: PreMap domain to SbjMap
missing `rdf:type`	W	SbjMap	SbjMap	ADD: PreMap domain to SbjMap
deprecation	W	PreMap	PreMap	–
`owl:complementOf`	W	PreMap	SbjMap	–

Table 1. Violations detected by assessing the mapping definitions. The first column describes the type of violation, the second its level (Warning or Error). The third specifies the expected RDF term according to the ontology or schema, while the fourth the term map defining how the RDF term is generated. The last specifies the refinement.

Automatically refining *domain-level* violations requires comparing recursively the type(s) assigned to the Subject Map with each predicate's domain, as specified at the different Predicate Maps. If not explicitly defined or inferred via a subclass, the predicate's domain is additionally assigned. This also requires a follow-up check for disjoint classes, which is of crucial importance especially when composition of different vocabularies and ontologies occurs.

Mapping Refinements based on Dataset Quality Assessment. Violations identified when the MDs are instantiated with values from the input source, can lead to a new round of refinements, if violations can be associated with a certain MD.

Mapping Refinements impact on Dataset Quality. The number of automated resolutions for violations detected at the mapping level depends on (i) the number of iterations over the data chunks of the input source (e.g., number of rows), (ii) the number of references to the input source (e.g., number of referred columns) and (iii) the number of returned values from the input source for each reference. To be more precise, if $\mathcal{I}$ is the number of iterations, $\mathcal{R}$ is the set of references the input source, and $V(r)$ values are returned for $r \in R$, then the total number of errors per violation is equal to the number of triples generated from this mapping definition: $I \cdot \prod_{r \in R} V(r)$. This means that the number of errors per violation identified (and resolved) at mapping level grows linearly in function of the number of iterations, and increases drastically if multiple references and returned values exist. For instance, assuming a mapping definition with 2 references to the input, where up to 3 values can be returned for each reference, contains a violation. Applied to an XML file with 1,000 elements, this could cause up to 6,000 error-prone triples in the worst case.

6 Use Cases and Adoption

Our Mapping Assessment and Refinement workflow with RML and RDFUnit is already being used in multiple different contexts. The DBpedia community adapted our mapping assessment solution to improve its mappings. Other popular, medium-sized datasets also benefit of our solution to ameliorate their mappings, such as DBLP. Moreover, various projects fully relied on our solution for their dataset generation, such as CDFLG and iLastic. Last, the proposed workflow was used to refine a challenge submission. Every dataset is unique in the way mappings are applied and different types of errors arise in each case. We indicatively describe a number of representative use cases below.

DBpedia *[23]* provides a *collaborative mapping approach* of Wikipedia infoboxes to the DBpedia ontology[10] through the db*pedia mappings wiki*[11]. DBpedia uses a wiki markup syntax for the mapping definitions and the output is adjusted in conformance to the DBpedia ontology. Although DBpedia uses the same wikitext syntax as Wikipedia –its original source– to define the MDs, the quality of wikitext-based MDs cannot be assessed directly, and thus certainly not in the same way as their resulting dataset. Thus, we automated the conversion of all DBpedia mappings to RML in order to make them processable from our tool stack. We introduced *wikitext serialisation* as a new Reference Formulation, since RML can be extended to express MDs for any type of input source. In total, we generated 674 distinct mapping documents for English, 463 for Dutch and a total of 4,468 for all languages. We used the DBpedia 2014 release and focused on a complete evaluation on the English and Dutch language editions as well as a mapping-only evaluation of all languages supported in the DBpedia mappings wiki. DBpedia originates from crowdsourced and semi-structured content and can thus be considered a noisy dataset. The MQA report was provided to the DBpedia community[12], who took advantage of it to manually refine DBpedia MDs. Automated refinements were not applicable in this case—as DBpedia framework still functions with the original MDs in wiki markup—but suggestions were provided.

Faceted DBLP The Computer Science bibliography (DBLP) collects open bibliographic information from major computer science journals and proceedings. Faceted DBLP builds upon the DBLP++ dataset, an enhancement of DBLP, originally stored in a mysql database. DBLP MDs are originally defined using D2RQ and were converted to RML using D2RQ-to-R2RML[13] to be processable by our workflow. DBLP, is a *medium-sized* dataset of very good quality according to our evaluation. Nonetheless, the workflow resulted in improvements.

[10] http://wiki.dbpedia.org/Ontology
[11] http://mappings.dbpedia.org
[12] http://goo.gl/KcSu3E
[13] https://github.com/RMLio/D2RQ_to_R2RML.git

Contact Details of Flemish Local Governments Dataset *(*cdflg*)* [14] In the scope of the EWI[15] project, the CDFLG dataset was generated using our workflow [6]. This is a real case of contact details for local governments in Flanders. CDFLG is annotated using the OSLO ontology [16], defined by the Open Standards for Linking Governments Working Group (V-ICT-OR, OSLO) under the OSLO (Open Standards for Local Administrations) Programme. Two subsequent versions of its RML mapping definitions were used to generate this dataset were assessed for their quality. The decrease of mapping violations over the mapping evolution indicates that our methodology can correctly identify errors.

iLastic [17] Our methodology was used in a use case for iMinds[18], a research institute founded by the Flemish Government, which published its own data regarding researchers, publications, projects, external partners etc., using the proposed workflow. The mapping definitions that specify how the data is mapped to the RDF model were stated using RML. After the primary mapping definitions were stated, they were fed to our proposed implementation and were refined twice, once based on the MQA results and once based on the DQA results, leading to their final version which is free of violations.

CEUR-WS The ESWC2015 Semantic Publishing Challenge (SPC)[19] is focused on refining and enriching CEUR-WS[20] linked dataset originally generated at the ESWC2014 edition[21]. It contains Linked Data about workshops, their publications and their authors. The workflow was well aligned with the requirements of this year's challenge and was used to evaluate last year's submission based on RML [8] and refine it to produce the base for this year's submission [15].

7 Evaluation and Discussion

The datasets mentioned in Section 6 were used for our evaluation. In this section, the results are described and certain observations are discussed in more details for each dataset and overall. The results are aggregated in Table 2 and are available at `http://rml.io/data/ISWC15`. For our evaluation, we used an 8-core Intel i7 machine with 8GB RAM and 256 SSD HD.

Overall, it is clear that the *computational complexity and time are significantly reduced* when assessing the mapping definitions compared to the complete RDF dataset (cf. Table 2). It takes 11 seconds to assess the approximately 700 mappings of English DBpedia, compared to assessing the whole DBpedia dataset

[14] `http://ewi.mmlab.be/cd/all`
[15] `http://ewi.mmlab.be`
[16] `http://purl.org/oslo/ns/localgov#`
[17] `http://explore.ilastic.be/`
[18] `http://iminds.be`
[19] `https://github.com/ceurws/lod/wiki/SemPub2015`
[20] `ceur-ws.org`
[21] `http://challenges.2014.eswc-conferences.org/index.php/SemPub`

	Dataset Assessment					*Mapping Assessment*					**Affect.**
Dataset	**Size**	**TC**	**Time**	**Fail.**	**Viol.**	**Size**	**Time**	**Fail.**	**Viol.**	**Ref.**	**triples**
DBpEn	62M	9,458	16.h	1,128	3.2M	115K	11s	1	160	–	255K
DBpNL	21M	10,491	1.5h	683	815K	53K	6s	1	124	–	106K
DBpAll	–	–	–	–	–	511K	32s	1	1,316	–	–
DBLP	12M	462	12h	7	8.1M	368	12s	2	8	6	8M
iLastic	150K	690	12s	23	37K	825	15s	3	26	23	37K
CDFLG	0.6K	2068	7s	15	678	558K	13s	4	16	13	631
CEUR-WS	2.4K	414	6s	7	783	702	5s	3	12	7	783

Table 2. Evaluation results summary. In the *Dataset Assessment* part, we provide the Size (number of triples), number of test cases, evaluation Time, Failed test cases and total individual Violations. In the *Mapping Assessment* part, we provide the mapping document Size (number of triples), evaluation Time, Failed test cases and Violation instances. Finally, we provide the number of dataset violations that can be addressed refining the mappings and estimated corresponding dataset violations that are resolved.

that takes of the order of several hours. In the latter case, the assessment requires examining each triple separately to identify, for instance, that 12M triples violated the range of foaf:primaryTopic, whereas with our proposed approach, only 1 triple needs to be examined. It is indisputable the workflow's *effectiveness*, as, in all cases that the dataset generation fully relies on its mapping definitions, the majority of violations is addressed. Moreover, if a set of RML mapping definitions is assessed for its quality, for every other new data source also mapped using these mapping definitions, the quality assessment does not need to be repeated for that part. Next, we discuss the results for each dataset in details:

DBpedia Most violations in DBpedia have *range-level* origin. When RDF is generated from the wikitext, the object type is not known and may result in wrong statements, as the DBpedia extraction framework automatically adjusts the predicate/object extraction according to the DBpedia ontology definitions. *Domain-level* violations occur as well, because users manually provide the class a Wikipedia infobox is mapped to and the ontology properties each infobox property will use. Our framework can, in this case, identify mismatches between the user-defined class and the rdfs:domain of each provided property. We observe that 8% of the errors in DBpedia in English and 13% of DBpedia in Dutch can be fixed directly at *mapping-level*. Not only the errors as such are directly pinpointed, but it also takes negligible time to have the refinements of the violations accomplished. The evaluation of all mappings for all 27 supported language editions resulted in a total of 1316 *domain-level* violations.

DBLP dataset has 7 individual violations, leading to 8.1M violated triples. The `swrc:editor` predicate defined in a Predicate Map expects a resource of `swrc:Person` type for its `domain` instead of `foaf:Agent` as defined in the corresponding Subject Map causing 21K errors. Similarly, approximately 3M errors occurred because a Predicate Map exists with `dcterms:bibliographicCitation` as its value whose `rdfs:domain` is `bibo:BibliographicResource`. However, the corresponding Subject Map(s) generate resources of type `dcmitype:Text`, `foaf:Document` or `swrc:Book` but definitely not the expected one, thus data publishers should remain warned for potential contradictions. Moreover, the missing

range of `foaf:page` and `foaf:homepage` can be fixed by refining the mapping definitions but, for links to external resources, it is common practice not to define their type. Except for 12K inverse functional violations for `foaf:homepage` that can not be addressed directly from the mapping definitions, all rest violations (98%) could be refined.

CDFLG In the first version of the CDFLG dataset, we found four violations: One caused by Predicate Object Maps that all have predicates that expect oslo:Address as their domain. However, the Subject Map is defined to be of type oslo:BasicAddress. In the same context, an incorrect range violation was identified for oslo:availableAt property. In general, violations related to Referencing Object Maps are among the most frequently encountered. Last, the object property schema:nationality was mapped as literal. The second version of CDFLG is a result of manually refining the mapping definitions according to the first mapping assessment's results. Besides the *domain level* violation, only few of the range violations remained (7%).

iLastic is particularly interesting because the workflow was used from the primary version of the mapping definitions, till they became free of violations. The first version was assessed and even contained R2RML schema violations, e.g., `rr:constant` had a string-valued object instead of a resource. If these mapping definitions were used, almost one fourth (25%) of its triples would be prone to errors. Although every violation was fixed after a couple of iterations assessing and refining the mapping definitions. For example, `cerif:isClassifiedBy` expects a `cerif:Classification` and not a `skos:Concept`, while `bibo:uri` expects a literal and not a resource as range. Similarly, `dcterms:issued` expects a literal and not `xsd:gYear`. A violation that occurred repeatedly was associated with the `cerif:internalidentifier` that requires a string-valued object, whereas it was associated with an Object Map that generated `xsd:positiveInteger` objects.

CEUR-WS 12 violations were identified in the dataset generated using RML for ESWC 2014 challenge and 10 out of them could already be detected at the mapping definitions. Most of them (7) were *domain-level* violations, annotating, for instance, resources of type `bibo:Volume` with properties for `bibo:Document` or for `<http://www.loc.gov/mads/rdf/v1#Address>`, e.g., for specifying the city, implying unwittingly that resources are both *Documents* and *Addresses*. The rest of the detected violations were related to contradicted datatypes, for instance, incorrectly specifying the datatype as `xsd:gYear`, while it is expected to be string. The mapping definitions for ESWC 2015 submission were produced using our workflow, were assessed and do not contain violations any more.

8 Related Work

We summarize the state of the art of the relevant fields: data mappings to the RDF data model and LD quality assessment.

Mapping Languages Several solutions exist to perform mappings from different data formats and serialisations to the RDF data model. In the case of data in XML format, existing XML solutions were used to define the mappings, such as XSLT, e.g., AstroGrid-D[22], or XPath, e.g., Tripliser[23], while the only mapping language defined specifically for XML to RDF mappings is X3ML[24]. In the same context, existing querying languages were also considered to describe the mappings, e.g., XSPARQL [2] which is a language that combines XQuery and SPARQL or Tarql[25]. Due to the lack of query languages or other ways to refer to data in CSV format or spreadsheets, different mapping languages were occasionally defined, e.g., the XLWrap's mapping language [21] that converts data in spreadsheets to RDF, or the declarative OWL-centric mapping language Mapping Master's M^2 [25] that converts data from spreadsheets into the Web Ontology Language (OWL). For relational databases, different mapping languages were defined [14], but the W3C-standardized R2RML prevailed.

Quality Assessment Different approaches have been developed that try to tackle various aspects of LD quality. These approaches can be broadly classified into (i) manual (e.g. [1, 3, 24, 28]); (ii) semi-automated (e.g. [10, 16]); or (iii) automated (e.g. [7, 13]) methodologies. These approaches introduce systematic methodologies to assess the quality of a dataset. Depending on the approach, we notice inability to produce easily interpretable results, a considerable amount of user involvement, application on specific datasets only or inability to evaluate a complete dataset during the assessment. SPIN[26] is a W3C submission aiming at representing rules and constraints on Semantic Web models using SPARQL. The approach described in [12] advocates the use of SPARQL and SPIN for RDF data quality assessment. In a similar way, Fürber et al. [11] define a set of generic SPARQL queries to identify missing or illegal literal values and datatypes and functional dependency violations. Another related approach is the *Pellet Integrity Constraint Validator*[27], which translates OWL integrity constraints into SPARQL queries. A more light-weight, although less expressive, RDF constraint syntax that is decoupled from SPARQL is offered from *Shape Expressions* (ShEx) [26] and *IBM Resource Shapes*[28].

9 Conclusions and Future Work

In this paper, we propose a methodology for assessing Linked Data quality for data originally stemming from semi-structured formats. We propose a workflow that relies on assessing the mapping definitions, rather than the RDF dataset they generate. The assessment report points exactly to the root causes of the

[22] http://www.gac-grid.de/project-products/Software/XML2RDF.html
[23] http://daverog.github.io/tripliser/
[24] https://github.com/delving/x3ml/blob/master/docs/x3ml-language.md
[25] https://github.com/cygri/tarql
[26] http://www.w3.org/Submission/spin-overview/
[27] http://clarkparsia.com/pellet/icv/
[28] http://www.w3.org/Submission/2014/SUBM-shapes-20140211/

violations and can be actively used to refine the mapping definitions. The automation of refinements or suggestions is facilitated based on a comprehensive analysis of different cases, and encountered violations are addressed at the origin. This essentially allows publishers to catch and correct violations before they even occur; moreover, fixing violations early avoids propagation where one flawed mapping rule leads to many faulty triples. The evaluation shows that our methodology is applicable to i) datasets without native [R2]RML mapping definitions, such as DBLP, ii) large datasets, such as DBpedia, as well as iii) datasets in the whole process of defining their mappings, such as iLastic. It was proven that assessing the quality of mapping definitions is more efficient in terms of computational complexity, and requires significantly less time to be executed compared to assessing the entire dataset. As our evaluation indicates, it takes only a few seconds to assess the mapping definitions, while it can be time-consuming and performance-intensive when this happens at dataset level. Especially with large datasets, this can take up to several hours. Our methodology was adopted by both the community of significant public datasets, such as DBpedia, and several projects, resulting in published Linked Data of higher quality. In the future, we plan to automate and improve the application of mapping definition refinements and integrate this step into the workflow of an interactive user interface.

Acknowledgements. This paper's research activities were funded by Ghent University, iMinds, the Institute for the Promotion of Innovation by Science and Technology in Flanders, the Fund for Scientific Research-Flanders and by grants from the EU's 7th & H2020 Programmes for projects ALIGNED (GA 644055), GeoKnow (GA 318159) and LIDER (GA 610782).

References

1. M. Acosta, A. Zaveri, E. Simperl, D. Kontokostas, S. Auer, and J. Lehmann. Crowdsourcing linked data quality assessment. In *12th International Semantic Web Conference, 21-25 October 2013, Sydney, Australia*, pages 260–276, 2013.
2. S. Bischof, S. Decker, T. Krennwallner, N. Lopes, and A. Polleres. Mapping between rdf and xml with xsparql. *Journal on Data Semantics*, 1(3):147–185, 2012.
3. C. Bizer and R. Cyganiak. Quality-driven information filtering using the wiqa policy framework. *Web Semant.*, 7(1):1–10, 2009.
4. B. W. Boehm. *Software Engineering Economics*. Prentice Hall PTR, 1981.
5. S. Das, S. Sundara, and R. Cyganiak. R2RML: RDB to RDF Mapping Language. Working group recommendation, W3C, Sept. 2012.
6. L. De Vocht, M. Van Compernolle, A. Dimou, P. Colpaert, R. Verborgh, E. Mannens, P. Mechant, and R. Van de Walle. Converging on semantics to ensure local government data reuse. *Proceedings of the 5th workshop on Semantics for Smarter Cities (SSC14), 13th International Semantic Web Conference (ISWC)*, 2014.
7. J. Debattista, C. Lange, and S. Auer. Representing dataset quality metadata using multi-dimensional views. In *Proceedings of the 10th International Conference on Semantic Systems*, pages 92–99, 2014.

8. A. Dimou, M. Vander Sande, P. Colpaert, L. De Vocht, R. Verborgh, E. Mannens, and R. Van de Walle. Extraction & Semantic Annotation of Workshop Proceedings in HTML using RML. In *Sem. Publ. Challenge of 11th ESWC*, 2014.
9. A. Dimou, M. Vander Sande, P. Colpaert, R. Verborgh, E. Mannens, and R. Van de Walle. RML: A Generic Language for Integrated RDF Mappings of Heterogeneous Data. In *Workshop on Linked Data on the Web*, 2014.
10. A. Flemming. Quality characteristics of linked data publishing datasources. Master's thesis, Humboldt-Universität of Berlin, 2010.
11. C. Fürber and M. Hepp. Using semantic web resources for data quality management. In P. Cimiano and H. Pinto, editors, *Knowledge Engineering and Management by the Masses*, volume 6317 of *LNCS*, pages 211–225. Springer, 2010.
12. C. Fürber and M. Hepp. Using SPARQL and SPIN for data quality management on the semantic web. In *BIS*, volume 47 of *LNBIP*, pages 35–46, 2010.
13. C. Guéret, P. T. Groth, C. Stadler, and J. Lehmann. Assessing Linked Data Mappings Using Network Measures. In *Proceedings of the 9th Extended Semantic Web Conference*, Lecture Notes in Computer Science, pages 87–102, 2012.
14. M. Hert, G. Reif, and H. C. Gall. A comparison of RDB-to-RDF mapping languages. I-Semantics '11, pages 25–32. ACM, 2011.
15. P. Heyvaert, A. Dimou, R. Verborgh, E. Mannens, and R. Van de Walle. Extraction and Semantic Annotation of Workshop Proceedings in HTML using RML. In *Semantic Publishing Challenge of the 12th ESWC*, 2015.
16. A. Hogan, A. Harth, A. Passant, S. Decker, and A. Polleres. Weaving the Pedantic Web. In *Linked Data On the Web*, 2010.
17. B. Hyland, G. Atemezing, and B. Villazón-Terrazas. Best Practices for Publishing Linked Data. Working group note, W3C, Jan. 2004.
18. J. Juran and F. Gryna. *Juran's Quality Control Handbook*. Industrial engineering series. McGraw-Hill, 1988.
19. D. Kontokostas, M. Brümmer, S. Hellmann, J. Lehmann, and L. Ioannidis. NLP data cleansing based on Linguistic Ontology constraints. In *Proc. of the Extended Semantic Web Conference 2014*, 2014.
20. 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*, pages 747–758, 2014.
21. A. Langegger and W. Wöß. XLWrap – Querying and Integrating Arbitrary Spreadsheets with SPARQL. In *Proceedings of 8th ISWC*, pages 359–374. Springer, 2009.
22. J. Lehmann. DL-learner: Learning concepts in description logics. *Journal of Machine Learning Research*, 10:2639–2642, 2009.
23. J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P. N. Mendes, S. Hellmann, M. Morsey, P. Kleef, S. Auer, and C. Bizer. DBpedia - a Large-scale, Multilingual Knowledge Base Extracted from Wikipedia. *Sem. Web Journal*, 2014.
24. P. N. Mendes, H. Mühleisen, and C. Bizer. Sieve: linked data quality assessment and fusion. In *EDBT/ICDT Workshops*, pages 116–123. ACM, 2012.
25. M. J. O'Connor, C. Halaschek-Wiener, and M. A. Musen. Mapping Master: a flexible approach for mapping spreadsheets to OWL. ISWC'10, 2010.
26. E. Prud'hommeaux, J. E. Labra Gayo, and H. Solbrig. Shape expressions: An rdf validation and transformation language. In *Proceedings of the 10th International Conference on Semantic Systems*, pages 32–40. ACM, 2014.
27. M. Schmachtenberg, C. Bizer, and H. Paulheim. Adoption of the linked data best practices in different topical domains. volume 8796 of *LNCS*. Springer, 2014.
28. A. Zaveri, A. Rula, A. Maurino, R. Pietrobon, J. Lehmann, and S. Auer. Quality assessment for linked data: A survey. *Semantic Web Journal*, 2015.