User-driven Quality Evaluation of DBpedia[*]

Amrapali Zaveri*
zaveri@informatik.uni-leipzig.de

Dimitris Kontokostas*
kontokostas@informatik.uni-leipzig.de

Mohamed A. Sherif*
sherif@informatik.uni-leipzig.de

Lorenz Bühmann*
buehmann@informatik.uni-leipzig.de

Mohamed Morsey*
morsey@informatik.uni-leipzig.de

Sören Auer+
auer@cs.uni-bonn.de

Jens Lehmann*
lehmann@informatik.uni-leipzig.de

* AKSW/BIS, Universität Leipzig
PO Box 100920, 04009
Leipzig, Germany

+ CS/EIS, Universität Bonn
Römerstra 164, 53117 Bonn
Bonn, Germany

ABSTRACT

Linked Open Data (LOD) comprises of an unprecedented volume of structured datasets on the Web. However, these datasets are of varying quality ranging from extensively curated datasets to crowdsourced and even extracted data of relatively low quality. We present a methodology for assessing the quality of linked data resources, which comprises of a manual and a semi-automatic process. The first phase includes the detection of common quality problems and their representation in a quality problem taxonomy. In the manual process, the second phase comprises of the evaluation of a large number of individual resources, according to the quality problem taxonomy via crowdsourcing. This process is accompanied by a tool wherein a user assesses an individual resource and evaluates each fact for correctness. The semi-automatic process involves the generation and verification of schema axioms. We report the results obtained by applying this methodology to DBpedia. We identified 17 data quality problem types and 58 users assessed a total of 521 resources. Overall, 11.93% of the evaluated DBpedia triples were identified to have some quality issues. Applying the semi-automatic component yielded a total of 222,982 triples that have a high probability to be incorrect. In particular, we found that problems such as object values being incorrectly extracted, irrelevant extraction of information and broken links were the most recurring quality problems. With this study, we not only aim to assess the quality of this sample of DBpedia resources but also adopt an agile methodology to improve the quality in future versions by regularly providing feedback to the DBpedia maintainers.

Keywords

Evaluation, DBpedia, Data Quality, RDF, Extraction

1. INTRODUCTION

The advent of semantic web technologies, as an enabler of Linked Open Data (LOD), has provided the world with an unprecedented volume of structured data currently amounting to 50 billion facts represented as RDF triples. Although publishing large amounts of data on the Web is certainly a step in the right direction, the data is only as usable as its quality. On the Data Web, we have varying quality of information covering various domains. There are a large number of high quality datasets (in particular in the life-sciences domain), which are carefully curated over decades and recently published on the Web. There are, however, also many datasets, which were extracted from unstructured and semi-structured information or are the result of some crowdsourcing process, where large numbers of users contribute small parts. DBpedia [1, 17] is actually an example for both - a dataset extracted from the result of a crowdsourcing process. Hence, quality problems are inherent in DBpedia. This is not a problem per se, since quality usually means fitness for a certain use case [12]. Hence, even datasets with quality problems might be useful for certain applications, as long as the quality is in the required range.

In the case of DBpedia, for example, the data quality is perfectly sufficient for enriching Web search with facts or suggestions about common sense information, such as entertainment topics. In such a scenario, where the DBpedia background knowledge can be, for example, used to show the movies Angelina Jolie was starring in and actors she played with it is rather neglectable if, in relatively few cases, a movie or an actor is missing. For developing a medical application, on the other hand, the quality of DBpedia is probably completely insufficient. Please note, that also on the traditional document-oriented Web we have varying quality of the information and still the Web is perceived to be extremely useful by most people. Consequently, a key challenge is to determine the quality of datasets published on the Web and make this quality information explicit. Other than on the document Web where information quality can be only indirectly, (e.g. via page rank, or vaguely) defined, we can have much more concrete and measurable data quality indicators for structured information, such as correctness of facts, adequacy of semantic representation or degree of coverage.

In this paper, we devise a data quality assessment methodology, which comprises of a manual and a semi-automatic process. We empirically assess, based on this methodology, the data quality of one of the major knowledge hubs on the Data Web – DBpedia. The first phase includes the detection of common quality problems and their representation in a comprehensive taxonomy of potential quality problems. In the manual process, the second phase comprises of the evaluation of a large number of individual resources, according to the quality problem taxonomy, using *crowdsourcing* in order to evaluate the type and extent of data quality problems occurring in DBpedia. Here we would like to clarify the use of crowdsource used in this paper. Crowdsourcing involves the creating if HITs (Human Intelligent Tasks), submitting them to a crowdsourcing platform (e.g. Amazon Mechanical Turk[1]) and providing a (financial) reward for each HIT [11]. However, we use the broader-sense of the word as a large-scale problem-solving approach by which a problem is divided into several smaller tasks (assessing the quality of each triple, in this case) that can be independently solved by a large group of people. Each represented fact is evaluated for correctness by each user and, if found problematic, annotated with one of 17 pre-defined quality criteria. This process is accompanied by a tool wherein a user assesses an individual resource and evaluates each fact for correctness.

The semi-automatic process involves the generation and verification of schema axioms, which yielded a total of 222,982 triples that have a high probability to be incorrect. We find that while a substantial number of problems exists, the overall quality is with a less than 11.93% error rate relatively high. With this study we not only aim to assess the quality of DBpedia but also to adopt a methodology to improve the quality in future versions by regularly providing feedback to the DBpedia maintainers to fix these problems.

Our main contributions are:

- a crowdsourcing based methodology for data quality assessment (Section 2),
- a comprehensive quality issue taxonomy comprising common knowledge extraction problems (Section 3),
- a crowdsourcing based data quality assessment tool (Section 4),
- an empirical data quality analysis of the DBpedia dataset performed using crowdsourcing (Section 5) and
- a semi-automated evaluation of data quality problems in DBpedia (Section 5).

We survey related work in Section 6 and conclude with an outlook on future work in Section 7.

2. ASSESSMENT METHODOLOGY

In this section, we describe a generalized methodology for the assessment and subsequent data quality improvement of resources belonging to a dataset. The assessment methodology we propose is depicted in Figure 1. This methodology consists of the following four steps: 1. Resource selection, 2. Evaluation mode selection, 3. Resource evaluation and 4. Data quality improvement. In the following, we describe these steps in more detail.

Step I: Resource selection. In this first step, the resources belonging to a particular dataset are selected. This selection can be performed in three different ways:

- *Per Class:* select resources belonging to a particular class
- *Completely random:* a random resource from the dataset
- *Manual:* a resource selected manually from the dataset

Choosing resources per class (e.g. animal, sport, place etc.) gives the user the flexibility to choose resources belonging to only those classes she is familiar with. However, when choosing resources from a class, the selection should be made in proportion to the number of instances of that class. Random selection, on the other hand, ensures an unbiased and uniform coverage of the underlying dataset. In the manual selection option, the user is free to select resources with problems that she has perhaps previously identified.

Step II: Evaluation mode selection. The assignment of the resources to a person or machine, selected in Step I, can be accomplished in the following three ways:

- *Manual:* the selected resources are assigned to a person (or group of individuals) who will then proceed to manually evaluate the resources individually.
- *Semi-automatic:* selected resources are assigned to a semi-automatic tool which performs data quality assessment employing some form of user feedback.
- *Automatic:* the selected resources are given as input to an automatic tool which performs the quality assessment without any user involvement.

For the semi-automatic evaluation, machine learning can be applied as shown in [4] and provided by the *DL-Learner framework* [16, 19], where the workflow can be as follows: (1) based on the instance data, generate OWL axioms which can also be seen as restrictions[2], e.g. learn characteristics (irreflexivity, (inverse) functionality, asymmetry) of properties as well as definitions and disjointness of classes in the knowledge base; (2) ask queries via SPARQL or a reasoner for violations of theses restrictions, e.g. in case of an irreflexive property, triples where subject and object are the same would indeed violate the characteristic of the irreflexivity. In the automatic case, a possible approach is to check for inconsistencies and other modelling problems as, e.g., described in [18].

[1] http://mturk.com

[2] A local Unique Name Assumption is used therefore, i.e. every named individual is assumed to be different from every other, unless stated explicitely otherwise

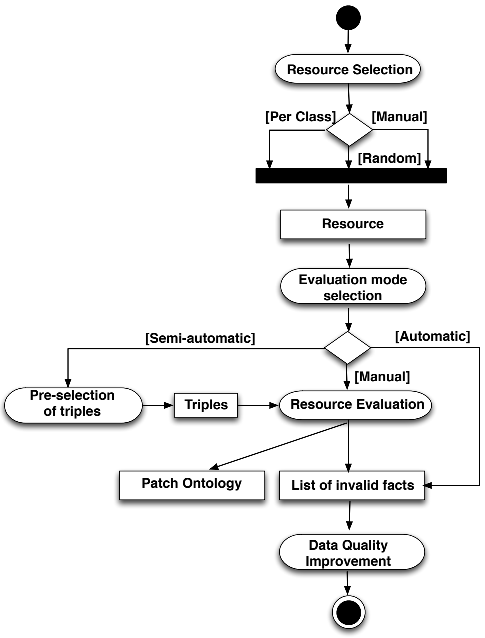

Figure 1: Workflow of the data quality assessment methodology.

Step III: Resource evaluation. In case of manual assignment of resources, the person (or group of individuals) evaluates each resource individually to detect the potential data quality problems. In order to support this step, a quality assessment tool can be used which allows a user to evaluate each individual triple belonging to a particular resource. If, in case of Step II, the selected resources are assigned to a semi-automatic tool, the tool points to triples likely to be wrong. For example, domain or range problems are identified by the tool and then assigned to a person to verify the correctness of the results.

Step IV: Data quality improvement. After the evaluation of resources and identification of potential quality problems, the next step is to improve the data quality. There are at least two ways to perform an improvement:

- Direct: editing the triple, identified to contain the problem, with the correct value
- Indirect: using the Patch Request Ontology[3] [13] which allows gathering user feedbacks about erroneous triples.

3. QUALITY PROBLEM TAXONOMY

A systematic review done in [23] identified a number of different data quality dimensions (criteria) applicable to Linked Data. After carrying out an initial data quality assessment on DBpedia (as part of the first phase of the manual assessment methodology cf. Section 5.1), the problems identified were mapped to this list of identified dimensions. In particular, *Accuracy, Relevancy, Representational-consistency and Interlinking* were identified to be problems affecting a large number of DBpedia resources. Additionally, these dimensions were further divided into categories and sub-categories. Table 1 gives an overview of these data quality dimensions along with their categories and sub-categories. We indicate whether the problems are automatically detectable (column D) and fixable (column F). The ones marked with a ✔in column D refer to those categories that can be automatically identified such as invalid datatypes (`"1981-01-01T00:00:00+02:00"^^xsd:gYear`), irrelevant properties (`dbpprop:imageCaption`) or dead links. The column F refers to those categories that can be automatically amended, like fixing an invalid datatype (`"1981"^^xsd:gYear`) or removing triples with irrelevant properties and dead links. If the problem is fixable, we determined whether the problem can be fixed by amending the (i) extraction framework (E), (ii) the mappings wiki (M) or (iii) Wikipedia itself (W). Moreover, the table specifies whether the problems are specific to DBpedia (marked with a ✔) or could potentially occur in any RDF dataset. For example, the sub-category *Special template not properly recognized* is a problem that occurs only in DBpedia due to the presence of specific keywords in Wikipedia articles that do not cite any references or resources (e.g. {{Unreferenced stub|auto=yes}}). On the other hand, the problems that are not DBpedia specific can occur in any other datasets. In the following, we provide the quality problem taxonomy and discuss each of the dimensions along with its categories and sub-categories in detail by providing examples.

Accuracy.

Accuracy is defined as *the extent to which data is correct, that is, the degree to which it correctly represents the real world facts and is also free of error* [23]. We further classify this dimension into the categories (i) object incorrectly extracted, (ii) datatype problems and (iii) implicit relationship between attributes.

Object incorrectly extracted. This category refers to those problems which arise when the object value of a triple is flawed. This may occur when the value is either (i) incorrectly extracted, (ii) incompletely extracted or (iii) the special template in Wikipedia is not recognized:

- *Object value is incorrectly extracted*, e.g.: `dbpedia:Oregon_Route_238 dbpprop:map "238.0"^^http://dbpedia.org/datatype/second.` This resource about state highway Oregon Route 238 has the incorrect property 'map' with value 238. In Wikipedia the attribute 'map' refers to the image name as the value: map=Oregon Route 238.svg. The DBpedia property only extracted the value 238 from his attribute value and gave it the datatype 'second' assuming it is a time value, which is incorrect.
- *Object value is incompletely extracted*, e.g.: `dbpedia:Dave_Dobbyn dbpprop:dateOfBirth "3"^^xsd:integer.` In this example, only the day of birth of a person is extracted and mapped to the 'dateofBirth' property when it should have been the entire date i.e. day, month and year. Thus, the object value is not completely extracted.
- *Special template not properly recognized*, e.g.: `dbpedia:328_Gudrun dbpprop:auto "yes"@en.` Certain article classifications in Wikipedia (such as "This article does not cite any references or sources.") are performed via special templates (e.g. {{Unreferenced stub|auto=yes}}). Such templates should be listed on a black-list and omitted by the DBpedia extraction in order to prevent non-meaningful triples.

[3] `http://141.89.225.43/patchr/ontologies/patchr.ttl#`

Datatype problems.

This category refers to those triples which are extracted with an incorrect datatype for a typed literal.

- *Datatype incorrectly extracted*, e.g.: `dbpedia:Stephen_Fry dbpedia-owl:activeYearsStartYear "1981-01-01T00:00:00+02:00"^^xsd:gYear`. In this case, the DBpedia ontology datatype property `activeYearsStartYear` has `xsd:gYear` as range. Although the datatype declaration is correct, it is formatted as `xsd:dateTime`. The expected value is `"1981"^^xsd:gYear`.

Implicit relationship between attributes.

This category of problems may arise due to (i) representation of one fact in several attributes, (ii) several facts encoded in one attribute or (iii) an attribute value computed from another attribute value in Wikipedia.

- *One fact is encoded in several attributes*, e.g.: `dbpedia:Barlinek dbpprop:postalCodeType "Postal code"@en`. In this example, the value of the postal code of the town of Barlinek is encoded in two attributes 'postal code type = Postal code' and 'postalcode = 74-320'. DBpedia extracts both these attributes separately instead of combining them together to produce one triple, such as: `dbpedia:Barlinek dbpprop:postalCode "74-320"@en`.
- *Several facts are encoded in one attribute*, e.g.: `dbpedia:Picathartes dbpedia-owl:synonym "Galgulus Wagler, 1827 (non Brisson, 1760: preoccupied)"@en`. In this example, even though the triple is not incorrect, it contains two pieces of information. Only the first word is the synonym, the rest of the value is a reference to that synonym. In Wikipedia, this fact is represented as ""synonyms = "Galgulus" ⟨*small*⟩ Wagler, 1827 ("non" [[Mathurin Jacques Brisson|Brisson]], 1760: [[Coracias|preoccupied]])/⟨/*small*⟩"". The DBpedia framework should ideally recognize this and separate these facts into several triples.
- *Attribute value computed from another attribute value*, e.g.: `dbpedia:Barlinek dbpprop:populationDensityKm "auto"@en`. In Wikipedia, this attribute is represented as "population density km2 = auto". The word "auto" is an indication in Wikipedia that the value associated to that attribute should be computed "automatically". In this case, the population density is computed automatically by dividing the population by area.

Relevancy.

Relevancy refers to *the provision of information which is in accordance with the task at hand and important to the users' query* [23]. The only category *Irrelevant information extracted* of this dimension can be further sub-divided into the following sub-categories: (i) extraction of attributes containing layout information, (ii) image related information, (iii) redundant attribute values and (iv) other irrelevant information.

- *Extraction of attributes containing layout information*, e.g.: `dbpedia:Lalsyri dbpprop:pushpinLabelPosition "bottom"@en`. Information related to layout of a page in Wikipedia, such as the position of the label on a pushpin map relative to the pushpin coordinate marker, in this example specified as "bottom", is irrelevant when extracted in DBpedia.
- *Image related information*, e.g.: `dbpedia:Three-banded_Plover dbpprop:imageCaption "At Masai Mara National Reserve, Kenya"@en`. Extraction of an image caption or name of the image is irrelevant in DBpedia as the image is not displayed for any DBpedia resource.
- *Redundant attributes value*, e.g.: The resource `dbpedia:Niedersimmental_ District` contains the redundant properties `dbpedia-owl:thumbnail, foaf:depiction, dbpprop:imageMap` with the same value "Karte Bezirk Niedersimmental 2007.png" as the object.
- *Other irrelevant information*, e.g.: `dbpedia:IBM_Personal_Computer dbpedia:Template:Infobox_information_appliance "type"@en`. Information regarding a templates infobox information, in this case, with an object value as "type" is completely irrelevant.

Representational-consistency.

Representational-consistency is defined as *the degree to which the format and structure of information conforms to previously returned information and other datasets.* [23] and has the following category:

- *Representation of number values*, e.g.: `dbpedia:Drei_Flsse_Stadion dbpprop:seating Capacity "20"^^xsd:integer`. In Wikipedia, the seating capacity for this stadium has the value "20.000", but in DBpedia the value displayed is only 20. This is because the value is inconsistently represented with a dot after the first two decimal places instead of a comma.

Interlinking.

Interlinking is defined as *the degree to which entities that represent the same concept are linked to each other* [23]. This type of problem is recorded when links to external websites or external data sources are either incorrect, do not show any information or are expired. We further classify this dimension into the following categories:

- *External websites:* Wikipedia usually contains links to external web pages such as, for example, the home page of a company or a music band. It may happen that these links are either incorrect, do not work or are unavailable.
- *Interlinks with other datasets:* Linked Data mandates interlinks between datasets. These links can either be incorrectly mapped or may not contain useful information. These problems are recorded in the following sub-categories: 1. links to Wikimedia, 2. links to Freebase, 3. links to Geospecies, 4. links generated via Flickr wrapper.

4. A CROWDSOURCING QUALITY ASSESSMENT TOOL

In order to assist several users in assessing the quality of a resource, we developed the *TripleCheckMate* tool[4] aligned

[4] available at `http://github.com/AKSW/TripleCheckMate`

Dimension	Category	Sub-category	D	F	DBpedia specific
Accuracy	Triple incorrectly extracted	Object value is incompletely extracted	–	E	–
		Object value is incompletely extracted	–	E	–
		Special template not properly recognised	✔	E	✔
	Datatype problems	Datatype incorrectly extracted	✔	E	–
	Implicit relationship between attributes	One fact encoded in several attributes	–	M	✔
		Several facts encoded in one attribute	–	E	–
		Attribute value computed from another attribute value	–	E + M	✔
Relevancy	Irrelevant information extracted	Extraction of attributes containing layout information	✔	E	✔
		Redundant attribute values	✔	–	–
		Image related information	✔	E	✔
		Other irrelevant information	✔	E	–
Represensational-Consistency	Representation of number values	Inconsistency in representation of number values	✔	W	–
Interlinking	External links	External websites	✔	W	–
	Interlinks with other datasets	Links to Wikimedia	✔	E	–
		Links to Freebase	✔	E	–
		Links to Geospecies	✔	E	–
		Links generated via Flickr wrapper	✔	E	–

Table 1: Data quality dimensions, categories and sub-categories identified in the DBpedia resources. Detectable (column D) means problem detection can be automised. Fixable (column F) means the issue is solvable by amending either the extraction framework (E), the mappings wiki (M) or Wikipedia (W). The last column marks the dataset specific subcategories.

with the methodology described in Section 2, in particular with Steps 1 – 3. To use the tool, the user is required to authenticate herself, which not only prevents spam but also helps in keeping track of her evaluations. After authenticating herself, she proceeds with the selection of a resource (Step 1). She is provided with three options: (i)*per class*, (ii)*completely random* and (iii)*manual* (as described in Step I of the assessment methodology).

After selecting a resource, the user is presented with a table showing each triple belonging to that resource on a single row. Step 2 involves the user evaluating each triple and checking whether it contains a data quality problem. The link to the original Wikipedia page for the chosen resource is provided on top of the page which facilitates the user to check against the original values. If the triple contains a problem, she checks the box is wrong. Moreover, she is provided with a taxonomy of pre-defined data quality problems where she assigns each incorrect triple to a problem. If the detected problem does not match any of the existing types, she has the option to provide a new type and extend the taxonomy. After evaluating one resource, the user saves the evaluation and proceeds to choosing another random resource and follow the same procedure.

Another important feature of the tool is to allow measuring of inter-rater agreements. That is, when a user selects a random method (*Any* or *Class*) to choose a resource, there is a 50% probability that she is presented with a resource that was already evaluated by another user. This probability as well as the number of evaluations per resource is configurable. Allowing many users evaluating a single resource not only helps to determine whether incorrect triples are recognized correctly but also to determine incorrect evaluations (e.g. incorrect classification of problem type or marking correct triples as incorrect), especially when crowdsourcing the quality assessment of resources. One important feature of the tool is that although it was built for DBpedia, it is parametrizable to accept any endpoint and, with very few adjustments in the database back-end (i.e. ontology classes and problem types) one could use it for any Linked Data dataset (open or closed).

5. EVALUATION OF DBPEDIA DATA QUALITY

5.1 Evaluation Methodology

Manual Methodology

We performed the assessment of the quality of DBpedia in two phases: *Phase I: Problem detection and creation of taxonomy* and *Phase II: Evaluation via crowdsourcing.*

Phase I: Creation of quality problem taxonomy. In the first phase, two researchers independently assessed the quality of 20 DBpedia resources each. During this phase an initial list of data quality problems, that occurred in each resource, was identified. These identified problems were mapped to the different quality dimensions from [23]. After analyzing the root cause of these problems, a refinement of the quality dimensions was done to obtain a finer classification of the dimensions. This classification of the dimensions into sub-categories resulted in a total of 17 types of data quality problems (cf. Table 1) as described in Section 3.

Phase II: Crowdsourcing quality assessment. In the second phase, we crowdsourced the quality evaluation wherein we invited researchers who are familiar with RDF to use the TripleCheckMate tool (described in Section 4). First, each user after authenticating oneself, chooses a resource by one of three options mentioned in Section 2. Thereafter, the extracted facts about that resource are shown to the user. The user then looks at each individual fact and records whether it contains a data quality problem and maps it to the type of quality problem.

Semi-automatic Methodology

We applied the semi-automatic method (cf. Section 2), which consists of two steps: (1) the generation of a particular set of schema axioms for all properties in DBpedia and (2) the manual verification of the axioms.

Step I: Automatic creation of an extended schema. In this step, the enrichment functionality of DL-Learner [4] for SPARQL endpoints was applied. Thereby for all properties in DBpedia, axioms expressing the (inverse) functional, irreflexive and asymmetric characteristic were generated, with a minimum confidence value of 0.95. For example, for the property `dbpedia-owl:firstWin`, which is a relation between Formula One racers and grand prix, axioms for all four mentioned types were generated: Each Formula One racer has only one first win in his career (functional), each grand prix can only be won by one Formula One racer (inverse functional). It is not possible to use the property `dbpedia-owl:firstWin` in both directions (asymmetric), and the property is also irreflexive.

Step II: Manual evaluation of the generated axioms. In the second step, we used at most 100 random axioms per axiom type and manually verified whether this axiom is appropriate. To focus on possible data quality problems, we restricted the evaluation data to axioms where at least one violation can be found in the knowledge base. Furthermore, we tried to facilitate the evaluation by taking also the target context into account, i.e. if it exists we consider the definition, domain and range as well as one random sample for a violation. When evaluating the inverse functionality for the property `dbpedia-owl:firstWin`, we can therefore make use of the following additional information:

```
Domain: dbpedia-owl:FormulaOneRacer Range: dbpedia-
    owl:GrandPrix
Sample Violation:
 dbpedia:Fernando_Alonso dbpedia-owl:firstWin dbpedia
     :2003_Hungarian_Grand_Prix.
 dbpedia:WikiProject_Formula_One dbpedia-owl:firstWin
     dbpedia:2003_Hungarian_Grand_Prix.
```

5.2 Evaluation Results

Manual Methodology.

An overview of the evaluation results is shown in Table 2[5]. Overall, only 16.5% of all resources were not affected by any problems. On average, there were 5.69 problems per resource and 2.24 problems excluding errors in the *dbprop namespace*[6] [17]. While the vast majority of resources have problems, it should also be remarked that each resource has 47.19 triples on average, which is higher than in most other LOD datasets. The tool was configured to allow two evaluations per resource and this resulted to a total of 268 inter-evaluations. We computed the inter-rater agreement for those resources, which were evaluated by two persons by adjusting the observed agreement with agreement by chance as done in Cohen's kappa[7]. The inter-rater agreement results – 0.34 for resource agreement and 0.38 for triple agreement – indicate that the same resource should be evaluated more than twice in future evaluations. To assess the accuracy of the crowdsourcing evaluation, we took a random sample of 700 assessed triples (out of the total 2928) and evaluated them for correctness based on the formula in [15] intended to be a representative of all the assessed triples. Additionally, we assumed a margin of 3.5% of error, which is a bound that we can place on the difference between the estimated correctness of the triples and the true value, and a 95% confidence level, which is the measure of how confident we are in that margin of error[8]. From these 700 triples, 133 were evaluated incorrectly resulting in about 81% of triples correctly evaluated.

Total no. of users	58
Total no. of distinct resources evaluated	521
Total no. of resources evaluated	792
Total no. of distinct resources without problems	86
Total no. of distinct resources with problems	435
Total no. of distinct incorrect triples	2928
Total no. of distinct incorrect triples in the *dbprop* namespace	1745
Total no. of inter-evaluations	268
No. of resources with evaluators having different opinions	89
Resource-based inter-rater agreement (Cohen's Kappa)	0.34
Triple-based inter-rater agreement (Cohen's Kappa)	0.38
No. of triples evaluated for correctness	700
No. of triples evaluated to be correct	567
No. of triples evaluated incorrectly	133
% of triples correctly evaluated	81
Average no. of problems per resource	5.69
Average no. of problems per resource in the *dbprop* namespace	3.45
Average no. of triples per resource	47.19
% of triples affected	11.93
% of triples affected in the *dbprop* namespace	7.11

Table 2: Overview of the manual quality evaluation.

Table 3 shows the total number of problems, the distinct resources and the percentage of affected triples for each problem type. Overall, the most prevalent problems, such as broken external links are outside the control of the DBpedia extraction framework. After that, several extraction and mapping problems that occur frequently mainly affecting accuracy, can be improved by manually adding mappings or possibly by improving the extraction framework.

When looking at the detectable and fixable problems from Table 1, in light of their prevalence, we expect that approximately one third of the problems can be automatically detected and two thirds are fixable by improving the DBpedia extraction framework. In particular, implicitly related attributes can be properly extracted with a new extractor, which can be configured using the DBpedia Mappings Wiki. As a result, we expect that the improvement potential is that the problem rate in DBpedia can be reduced from 11.93% to 5.81% (calculated by subtracting 7.11% from 11.93% reported in Table 2). After revising the DBpedia extraction framework, we will perform subsequent quality assessments using the same methodology in order to realize and demonstrate these improvements.

Semi-automatic Methodology.

The evaluation results in Table 4 show that for the irreflexive case all 24 properties that would lead to at least one violation should indeed be declared as irreflexive. Applying the irreflexive characteristic would therefore help to find overall 236 critical triples, for e.g. `dbpedia:2012_Coppa_Italia_Final dbpedia-owl:followingEvent`

[5] Also available at: `http://aksw.org/Projects/DBpediaDQ`

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

[7] `http://en.wikipedia.org/wiki/Cohen%27s_kappa`

[8] `http://research-advisors.com/tools/SampleSize.htm`

Criteria	IT	DR	AT %
Accuracy			
Object incorrectly extracted	32	14	2.69
Object value is incorrectly extracted	259	121	23.22
Object value is incompletely extracted	229	109	20.92
Special template not recognized	14	12	2.30
Datatype problems	7	6	1.15
Datatype incorrectly extracted	356	131	25.14
Implicit relationship between attributes	8	4	0.77
One fact is encoded in several attributes	670	134	25.72
Several facts encoded in one attribute	87	54	10.36
Value computed from another value	14	14	2.69
Accuracy unassigned	31	11	2.11
Relevancy			
Irrelevant information extracted	204	29	5.57
Extraction of layout information	165	97	18.62
Redundant attributes value	198	64	12.28
Image related information	121	60	11.52
Other irrelevant information	110	44	8.45
Relevancy unassigned	1	1	0.19
Representational-consistency			
Representation of number values	29	8	1.54
Representational-consistency unassigned	5	2	0.38
Interlinking			
External websites (URLs)	222	100	19.19
Interlinks with other datasets (URIs)	2	2	0.38
Links to Wikimedia	138	71	13.63
Links to Freebase	99	99	19.00
Links to Geospecies	0	0	0.00
Links generated via Flickr wrapper	135	135	25.91
Interlinking unassigned	3	3	0.58

Table 3: Detected number of problem for each of the defined quality problems. IT = Incorrect triples, DR = Distinct resources, AT = Affected triples.

`dbpedia:2012_Coppa_Italia_Final`, which is not meaningful as no event is the following event of itself. For asymmetry, we got 81 approved properties, for example, containing `dbpedia-owl:starring` with domain `Work` and range `Actor`. Compared with this, there are also some properties where asymmetry is not always appropriate, e.g. `dbpedia-owl:influenced`.

Functionality, i.e. having at most one value of a property, can be applied to 76 properties. During the evaluation, we observed invalid facts such as, for example, two different values `2600.0` and `1630.0` for the density of the moon `Himalia`. We spotted overall 199,480 errors of this type in the knowledge base. As the result of the inverse functionality evaluation, we obtained 13 properties where the object in the triple should only be related to one unique subject, e.g. there should only be one Formula One racer which won a particular grand prix, which is implicit when using the property `dbpedia-owl:lastWin`.

6. RELATED WORK

Web data quality assessment frameworks. There are a number of data quality assessment dimensions that have already been identified relevant to Linked Data, namely, accuracy, timeliness, completeness, relevancy, conciseness, consistency, to name a few [2]. Additional quality criteria such as uniformity, versatility, comprehensibility, amount of data, validity, licensing, accessibility and performance were also introduced to be additional means of assessing the quality of LOD [7]. Additionally, there are several efforts in developing data quality assessment frameworks in order to assess the data quality of LOD. These efforts are either semi-automated [7], automated [8] or manual [3, 20].

Even though these frameworks introduce useful methodologies to assess the quality of a dataset, either the results are difficult to interpret, do not allow a user to choose the input dataset or require a considerable amount of user involvement. In our experiment, we used crowdsourcing to perform the evaluation because (1) none of the frameworks provided the granularity of quality criteria that we identified to be quality problems in DBpedia resources and (2) we were interested in whether it was possible to use crowdsourcing to assess and thus improve the quality of a dataset.

Concrete Web Data quality assessments. An effort to assess the quality of web data was undertaken in 2008 [5], where 14.1 billion HTML tables from Google's general-purpose web crawl were analyzed in order to retrieve those tables that have high-quality relations. Additionally, there have been studies focused on assessing the quality of RDF data [9] to report the errors occurring while publishing RDF data and the effects and means to improve the quality of structured data on the web. As part of an empirical study [10] 4 million RDF/XML documents were analyzed, which provided insights into the level of conformance in these documents with respect to the Linked Data guidelines. Even though these studies accessed a vast amount of web or RDF/XML data, most of the analysis was performed automatically and therefore the problems arising due to contextual discrepancies were overlooked. Another study aimed to develop a framework for the DBpedia quality assessment [14]. In this study, particular problems of the DBpedia extraction framework were taken into account and integrated in the framework. However, only a small sample (75 resources) were assessed in this case and an older DBpedia version (2010) was analyzed.

Crowdsourcing-based tasks. There are already a number of efforts which use crowdsourcing focused on a specific type of task. For example, crowdsourcing is used for entity linking or resolution [6], quality assurance and resource mangement [22] or for enhancement of ontology alignments [21] especially in Linked Data. However, in our case, we did not submit tasks to the popular internet marketplaces such as Amazon Mechanical Turk or CrowdFlower[9]. Instead, we used the intelligence of a large number of researchers who were particularly conversant with RDF to help assess the quality of one of the important and most linked dataset, DBpedia.

7. CONCLUSION AND OUTLOOK

To the best of our knowledge, this study is the first comprehensive empirical quality analysis for more than 500 resources of a large Linked Data dataset extracted from crowdsourced content. We found that a substantial number of problems exist and the overall quality, with a 11.93% error rate, is moderate. Moreover, the semi-automatic analysis revealed more than 200,000 violations of property characteristics. In addition to the quality analysis of DBpedia, we devised a generic methodology for Linked Data quality analysis, derived a comprehensive taxonomy of extraction quality problems and developed a tool which can assist in

[9] `http://crowdflower.com/`

Characteristic	#Properties		Correct	#Violations			
	Total	Violated		Min.	Max.	Avg.	Total
Irreflexivity	142	24	24	1	133	9.8	236
Asymmetry	500	144	81	1	628	16.7	1358
Functionality	739	671	76	1	91581	2624.7	199480
Inverse Functionality	52	49	13	8	18236	1685.2	21908

Table 4: Results of the semi-automatic evaluation. The table shows the total number of properties that have been suggested to have the given characteristic by Step I of the semi-automatic methodology, the number of properties that would lead to at least one violation when applying the characteristic, the number of properties where the characteristic is meaningful (manually evaluated) and some metrics for the number of violations.

the evaluation. All these contributions can be reused for analyzing any other extracted dataset (by domain experts). The detailed analysis of data quality problems allows us to devise and implement corresponding mitigation strategies. Many of the problems found can be firstly automatically detected and secondly avoided by (1) improving existing extractors, (2) developing new ones (e.g. for implicitly related attributes) or (3) improving and extending mappings and extraction hints on the DBpedia Mappings Wiki.

With this study, we not only aim to assess the quality of this sample of DBpedia resources but also adopt an agile methodology to improve the quality in future versions by regularly providing feedback to the DBpedia maintainers to fix these problems. We plan to improve the DBpedia extraction framework along these detected problems and periodically revisit the quality analysis (in regular intervals) in order to demonstrate possible improvements.

8. REFERENCES

[1] S. Auer and J. Lehmann. What have Innsbruck and Leipzig in common? extracting semantics from wiki content. In *ESWC*, volume 4519 of *LNCS*, pages 503–517. Springer, 2007.

[2] C. Bizer. *Quality-Driven Information Filtering in the Context of Web-Based Information Systems*. PhD thesis, Freie Universität, March 2007.

[3] C. Bizer and R. Cyganiak. Quality-driven information filtering using the wiqa policy framework. *Web Semantics*, 7(1):1 – 10, Jan 2009.

[4] L. Bühmann and J. Lehmann. Universal owl axiom enrichment for large knowledge bases. In *EKAW*, 2012.

[5] M. J. Cafarella, A. Y. Halevy, D. Z. Wang, E. Wu, and Y. Zhang. Webtables: exploring the power of tables on the web. *PVLDB*, 1(1):538–549, 2008.

[6] G. Demartini, D. Difallah, and P. Cudré-Mauroux. Zencrowd: Leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking. In *21st International Conference on World Wide Web WWW 2012*, pages 469 – 478, 2012.

[7] A. Flemming. Quality characteristics of linked data publishing datasources. Master's thesis, Humboldt-Universität of Berlin, 2010.

[8] C. Guéret, P. T. Groth, C. Stadler, and J. Lehmann. Assessing linked data mappings using network measures. In *ESWC*, volume 7295 of *LNCS*, pages 87–102. Springer, 2012.

[9] A. Hogan, A. Harth, A. Passant, S. Decker, and A. Polleres. Weaving the pedantic web. In *LDOW*, 2010.

[10] A. Hogan, J. Umbrich, A. Harth, R. Cyganiak, A. Polleres, and S. Decker. An empirical survey of linked data conformance. *Journal of Web Semantics*, 14:14–44, July 2012.

[11] J. Howe. The rise of crowdsourcing. *Wired Magazine*, 14(6), 06 2006.

[12] J. Juran. *The Quality Control Handbook*. McGraw-Hill, New York, 1974.

[13] M. Knuth, J. Hercher, and H. Sack. Collaboratively patching linked data. *CoRR*, 2012.

[14] P. Kreis. Design of a quality assessment framework for the dbpedia knowledge base. Master's thesis, Freie Universität Berlin, 2011.

[15] Krejcie and Morgan. Determining sample size for research activities. *Educational and Psycholoigcal Measurement*, 30:607–610, 1970.

[16] J. Lehmann. DL-Learner: learning concepts in description logics. *Journal of Machine Learning Research (JMLR)*, 10:2639–2642, 2009.

[17] J. Lehmann, C. Bizer, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann. DBpedia - a crystallization point for the web of data. *Journal of Web Semantics*, 7(3):154–165, 2009.

[18] J. Lehmann and L. Bühmann. Ore - a tool for repairing and enriching knowledge bases. In *ISWC2010*, Lecture Notes in Computer Science, Berlin / Heidelberg, 2010. Springer.

[19] J. Lehmann and P. Hitzler. Concept learning in description logics using refinement operators. *Machine Learning journal*, 78(1-2):203–250, 2010.

[20] B. C. Mendes P.N., Mühleisen H. Sieve: Linked data quality assessment and fusion. In *LWDM*, 2012.

[21] C. Sarasua, E. Simperl, and N. Noy. Crowdmap: Crowdsourcing ontology alignment with microtasks. In *The Semantic Web – ISWC 2012*, Lecture Notes in Computer Science, pages 525–541. Springer Berlin Heidelberg, 2012.

[22] J. Wang, T. Kraska, M. J. Franklin, and J. Feng. Crowder: crowdsourcing entity resolution. *Proc. VLDB Endow.*, 5:1483–1494, July 2012.

[23] A. Zaveri, A. Rula, A. Maurino, R. Pietrobon, J. Lehmann, and S. Auer. Quality assessment methodologies for linked open data. Under review, available at http://www.semantic-web-journal.net/content/quality-assessment-methodologies-linked-open-data.