Semi-Automatic Quality Assessment of Linked Data without Requiring Ontology

Saemi Jang, Megawati, Jiyeon Choi, and Mun Yong Yi

Department of Knowledge Service Engineering, KAIST
{sammy1221,megawati,jeeyeon51,munyi}@kaist.ac.kr

Abstract. The development of Semantic Web technology has fuelled the creation of a large amount of Linked Data. As the amount of data increases, various issues have been raised. In particular, the quality of data has become important. A number of studies have been conducted to evaluate the quality of linked data. However, most of the approaches are operational only when a data schema such as ontology exists. In this paper, we present a new approach for conducting linked data quality assessment by evaluating the quality of linked data without involving an ontology. Our approach consists of three activities: (1) pattern analysis, (2) pattern generation, and (3) data quality evaluation. A pattern is a structure used to measure the quality of data. For the validation of the proposed approach, we have conducted two studies - one involving English DBpedia, which has a relatively well-developed ontology, and the other involving Korean DBpedia, which lacks an ontology. Our approach shows comparable performances when compared with RDFUnit for English DBpedia and high accuracy results while assessing the quality of Korean DBpedia, for which RDFUnit cannot be used.

Keywords: Data Quality Assessment, Assessment without Ontology, Linked Data, Pattern Generation, DBpedia

1 Introduction

Linked data is an international endeavor to interconnect structured data on the Web. The development of Semantic Web technology has fuelled the creation of a large amount of Linked Data. There exists more than a thousand number of linked data, covering a wide range of different domains[1]. As the amount of data increases, numerous problems have been discovered regarding the data either syntactically or semantically (e.g. invalid data, data inconsistency, etc). DBpedia also still has such problems even thought it is one of the most well-organized and wildely used linked data resource. In addition, such errors in the extant linked data resources (e.g., DBpedia) may be enlarged in other systems that rely on those resources (e.g., Q&A systems). Thus the quality of the data has become important and a demand for accurate quality assessment methods has increased.

The quality of data is defined as fitness of use [10, 11] and includes various factors such as accuracy, relevancy, representation, and accessibility [10]. There have been many prior data quality assessment approaches [3, 6–8]. Depending on a goal or a target of the assessment methods, different factors are selectively employed and the assessment processes are also different, ranging from semi-automatic approach that requires user involvement to fully automatic approach. The main common ground of them is the data quality assessment is based on ontology that is built from the target linked data. Therefore it is not feasible to use prior data quality assessment approaches for linked data having no ontology. Of course we can build up our own ontology. However it is a difficult and time

[1] State of the LOD Cloud 2014 document published in April 2014 (http://linkeddatacatalog.dws.informatik.uni-mannheim.de/state/)

consuming work since it is done manually or semi-automatically by domain experts [12, 14]. Although automatic ontology generation frameworks have been introduced, it only works for English and limited domains [13, 14].

Main contributions: We propose a novel assessment method that performs the quality assessment of linked data without requiring ontology. In general, a large portion of the data in a knowledge resource is valid data because they usually took several debugging passes not only before but also after being released on the web [4]. We exploit this observation and this is the basic assumption of our approach. We first analyze the data patterns in a knowledge resource and rank the patterns based on the appearance ratio. Then we take top k (e.g., five) patterns for each property and compute the average ratio of them. This average value is a threshold that is the standard for deciding whether a given pattern is valid or not. Finally we take the patterns appearing more frequently than the threshold and they become test case patterns. Based on the generated test patterns, we evaluate the quality of knowledge resource. Since our approach directly utilizes a knowledge resource without requiring ontology, we can apply it to any language and any domain. Also our approach can work with any kind of pattern structures.

We validate our method in two aspects including the accuracy of generated test case patterns and the accuracy of assessment results. To measure the accuracy of generated test case patterns by our method, we use English DBpedia that is one of the most well-maintained knowledge resources and compare patterns generated by our method with the patterns created based on ontology. We also use Korean DBpedia as a localized, non-English DBpedia to measure the accuracy of assessment results, which lacks an ontology. We found that our method shows a high consistency (up to 89%) between generated patterns and existing patterns. Also it reached 79% F1-measure while assessing the quality of the localized DBpedia. These results demonstrate the accuracy and flexibility of our approach.

In the rest of this paper, we first explain our approach for the data quality assessment in Sec. 2 and then we validate the accuracy and usefulness of our approach in two perspectives including the accuracy of generated test case patterns (Sec. 3) and the accuracy of quality assessment results (Sec. 4). Finally we discuss the implications of the findings in relation to prior related research in Sec. 5 and conclude our work in Sec. 6.

2 Quality Assessment Without Requiring Ontology

For the assessment of data quality, we first analyze the data structure of the given knowledge resource. Specially we check whether there exists a data schema or not. If it has a data schema (e.g., ontology), we use a prior assessment method that utilize the data schema like SWIQA [16] and RDFUnit [4]. When there exists no data schema for the knowledge resource, we cannot evaluate the quality based on these prior approaches. Although we can generate a data schema for the target data resource, it is a time consuming work while requiring involvement of domain experts. To address this issue, we propose a novel quality assessment methodology that measures the quality of a given data resource having no data schema. In this section, we first provide the overview of our approach (Sec. 2.1) and explain the details of our test case pattern generation algorithm (Sec. 2.2 and 2.3).

2.1 Overview of Our Approach

Our method is a semi-automatic assessment approach and it mainly consists of three steps: 1) pattern analysis, 2) test case pattern generation, and 3) data quality evaluation (see Fig. 1). To measure the quality of the data resource, a criterion should be defined.

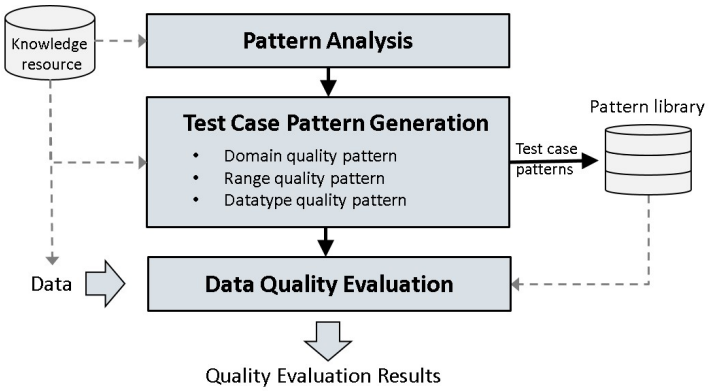

Fig. 1. Workflow of the data quality assessment methodology without ontology.

Data Quality Pattern (DQTP) is one of the widely employed standard for assessing the data quality of linked data and it includes various types of patterns [4]. In the first step, we define pattern structures that will be used for the quality assessment as a form of DQTP. This is a manual work and domain experts determine patterns that suit for the target data resource. According to the pattern structures defined in the first step, we figure out a set of test case patterns that represent valid data from the linked data. This second step is done with our automatic test case pattern generation algorithm, explained below in detail. Finally, we evaluate the quality of the data by applying the generated test case patterns to the data.

2.2 Quality Assessment Criteria

We use the Data Quality Test Pattern (DQTP) to define the quality assessment criteria. DQTP is a tuple (V,S), where V is a set of typed pattern variables and S is a SPARQL[2] query template with placeholders for the variables from V [4]. The quality assessment criteria are defined by domain experts as form of DQTPs. Usually data in a knowledge resource is given in the form of RDF triples, which consists of subject, predicate, and object. Our method is designed to work for a knowledge resource that uses RDF triples. In RDF, a predicate maps a subject into an object. Domain is all possible types which can be contained by the subject. Range is all possible types that can be contained by the object. Literal values ensure a certain data type determined by the property used, e.g. string data type is described as `xs:string` in English DBpedia. RDFSDOMAIN, RDFSRANGE and RDFSRANGED in Zaveri et al. [4] are well-known examples of DQTPs for RDF. In our method, the role of DQTP is the same with the them, but it is different from them as it directly works on the knowledge resource and not on the ontology.

2.3 Test Case Pattern Generation Algorithm

For a given pattern structure (i.e. DQTP), we generate test case patterns automatically. The goal of our test case pattern generation algorithm is to find data patterns that have correct information. In general, most information in a knowledge resource is valid since they are built with domain experts while taking several bug fixing processes. Base on this observation, we assume that more frequently appeared data patterns are more credible patterns.

[2] http://www.w3.org/TR/rdf-sparql-query/

RDFUnit	Approach	Definition
RDFSDOMAIN	DQP	The attribution of a resource's property (with a certain value) is only valid if the resource is of a certain type.
RDFSRANGE	RQP	The attribution of a resource's property is only valid if the value is of a certain type
RDFSRANGED	TQP	The attribution of a resource's property is only if the literal value has a certain datatype

Table 1. Definition of Patterns. (Definition refer to RDFUnit article)

To figure out valid test case patterns from the whole dataset in the knowledge resource, we use a two-step algorithm. In the first step, we check the pattern of all data depending on the given DQTP and compute the appearance ratio of each pattern. Then, for each predicate, we select the top k patterns and compute the ratio of the number of RDF triples that represent the selected patterns over the whole number of RDF triples in the knowledge resource. When we get the ratios for all predicates, we compute the average value of them. This average value becomes the threshold for selecting test case patterns. In the second step, we build the set of test case pattern. We check all patterns in the knowledge resource and add the patterns whose appearance ratio is higher than the threshold into the test case pattern set. If for a predicate there is no pattern having higher appearance ratio, we take the pattern having highest ratio for the property.

2.4 Data Quality Evaluation

Data quality evaluation entails the measurement of quality dimensions that can be considered as the characteristics of the resource [2]. In this paper, information accuracy and logical consistency are the feature of quality dimensions. While the existing method [4] uses only one type according to the definition of ontology, our approach uses one or more types determined by threshold (Sec. 2.3). As the circumstances require, an upper-class type is used in our approach. For quality assessement, our approach evaluates whether the data conform to one of the identified types or not.

3 Validation: Test Case Pattern Generation

The best way to exactly measure the accuracy of our quality assessment approach is comparing the evaluation results with the ground truth. However making ground truth is not feasible because it requires manual examinations of all the data in the knowledge resource. Instead, we compare our method with one of the previous work that relies on ontology. We use English DBpedia[3] and RDFUnit [4] as a benchmark considering that labels are found in almost all classes and properties in the ontology of English DBpedia and RDFUnit is the most recent pattern-based quality assessment method.

3.1 Test Case Generation without Ontology for English DBpedia

We defined three types of patterns including Domain Quality Pattern (DQP), Range Quality Pattern (RQP), and dataType Quality Pattern(TQP)[4]. The three patterns have same criteria with RDFSDOMAIN, RDFSRANGE, and RDFSRANGED, respectively Table 1 shows the definition of each pattern. The only difference is that our method works on the linked data itself, different from RDFUnit that uses ontology. With the three

[3] http://wiki.dbpedia.org/
[4] Test case patterns are available from `https://github.com/KAIST-KIRC/SAQA`

	Predicate	DQP	RQP	TQP
English DBpedia	2,750	1,368	601	739
Korean DBpedia	1,070	955	317	166

Table 2. The number of unique predicates and unique patterns in each DBpedia.

	DQP	RQP	TQP
The total number of patterns for RDFUnit	4,844	1,614	944
The number of unique predicates in RDFUnit	2,421	807	944
The total number of patterns with triples in resource ($\|P_{total}\|$)	2,328	978	501
The total number of generated patterns with our method ($\|P_{gen}\|$)	2,310	956	496
The Pattern generation rate ($\|P_{gen}\|/\|P_{total}\|$)	99.2%	97.8%	99.0%
The number of consistent patterns($\|P_{consist}\|$)	2,064	768	336
The pattern generation accuracy of our method ($\|P_{consist}\|/\|P_{gen}\|$)	89.4%	80.3%	67.7%

Table 3. Overview of the evaluation results for English DBpedia

patterns, we first show the actual test case generation process of our method on English DBpedia. Then, we validate the accuracy of our method by comparing the generated test case patterns with the set of patterns generated from ontology by RDFUnit.

Table 2 shows the statics of DBpedia we used. To generate test cases, we examined all types for a subject and an object connected by a predicate to define possible domain, range, and/or data types using SPARQL query. Then we calculated the ratio for each pattern following our test case generation algorithm (Sec. 2.3). We took top five patterns (i.e. $k = 5$) and the average ratio (i.e. threshold) was 22% for DQP. We decided 17% as the threshold for RQP following the same processes used for DQP. Based on the thresholds, we generated the test case patterns for DQP and RQP. For TQP, most of the triples has a single data pattern. Consequently, we used the top one pattern for each predicate.

3.2 Analysis

We compared the test case pattern generation results with those generated by RDFUnit. For RDFUnit, we used RDFSDOMAIN, RDFSRANGE, and RDFSRANGED that are matched with DQP, RQP, and TQP respectively. We do not consider RDFUnit patterns that do not have any associated triples in the resource. Our approach generates patterns by triples in the resource, but the RDFUnit is able to generate patterns using ontology even if the knowledge resource does not have the triple.

Table 3 shows the overview of the comparison between our approach and RDFUnit. It shows more than 97% (up to 99%) of pattern generation rates when triples exist. We also measured the consistency, which means the ratio of matched patterns between a set of patterns generated by our approach and the patterns generated by RDFUnit. Our approach achieves 89.35%, 80.33%, and 67.7% consistency for DQP, RQP, and TQP respectively. We noticed a relatively lower consistency rate for TQP, which is due to the fact that the generated patterns have equivalent meanings with those generated by RDFUnit, but they come from different resources. For instance, DBpedia ontology defined object data type of `dbo:alias`[5] as `rdf:langString` but the generated pattern has `xsd:String` as TQP of `dbo:alias`. This problem can be solved by adding a mechanism that maps same data types to representative data types. In our framework, this mechanism is not

[5] We used `http://prefix.cc` to express all name spaces as prefix. In the case of Korean DBpedia, it does not exist in `prefix`. So we expressed `http://ko.dbpedia.org/property/` as `prop-ko` and `http://ko.dbpedia.org/resource/` as `db-ko`.

owl:class (685)		
label	cnt	%
en	685	100%
el	577	84.23%
de	553	80.73%
nl	486	70.95%
fr	423	61.75%
ja	252	36.79%
it	208	30.36%
ko	91	13.28%
pt	87	12.70%
es	76	11.09%
pl	28	4.09%
sl	17	2.48%
zh	11	1.61%
tr	7	1.02%
ga	5	0.73%
eu	4	0.58%
ca	3	0.44%
ar	3	0.44%
id	3	0.44%
bn	3	0.44%
ru	2	0.29%
be	1	0.15%

owl:ObjectProperty (1079)		
label	cnt	%
en	1079	100%
de	562	52.09%
el	292	27.06%
nl	260	24.10%
fr	117	10.84%
sr	96	8.90%
pt	66	6.12%
ja	50	4.63%
es	36	3.34%
pl	25	2.32%
it	22	2.04%
tr	15	1.39%
ru	6	0.56%
ca	2	0.19%
ar	1	0.09%
ga	1	0.09%
id	1	0.09%
eu	1	0.09%
cs	1	0.09%

owl:DatatyPeproperty (1716)		
label	cnt	%
en	1715	99.94%
de	716	41.72%
el	340	19.81%
nl	297	17.31%
sr	161	9.38%
fr	122	7.11%
pt	69	4.02%
ja	60	3.50%
es	29	1.69%
it	12	0.70%
pl	12	0.70%
sl	5	0.29%
tr	5	0.29%
ru	5	0.29%
ga	3	0.17%
id	2	0.12%
bn	1	0.06%

Fig. 2. The statistics of the label for the class and property in DBpedia Ontology

implemented yet and we leave it for future work. Nonetheless, our approach generally achieves high pattern generation rates and the generated patterns show high consistency with the patterns generated from ontology by RDFUnit. Such high generation rates and consistency rates are in support of the reliable performances of the proposed approach in the environment where ontology is readily available.

4 Validation: Quality Assessment Accuracy

There are localized versions of DBpedia in 125 languages and most of them do not have their ontologies (Fig. 2). Our assessment method mainly aims to handle such a localized DBpedia and to evaluate the quality of the knowledge resource. To show the generality and usefulness of our approach, we apply our approach to one of the localized DBpedia, Korean DBpedia. In this section we first examine the test case generation process for the localized DBpedia. Then, we analyze the quality assessment results produced by our approach and validate its accuracy.

4.1 Data Quality Assessment for Korean DBpedia

Localized version of Korean DBpedia consists of 32 million triples with 18,617 different properties. Korean DBpedia itself has 9,424 properties while the rests are properties from English DBpedia and external properties. Among those properties, we only used properties that are carried by more than 100 triples. There exist only 1,070 properties for the condition. Korean DBpedia does not have an ontology, the fact that contains description about domain and range for each subject and object connected by its properties. For Korean DBpedia, we also used the three types of patterns (DQP, RQP and TQP) and took top five patterns, as they were the same in the case of English DBpedia.

Resource	Pattern	Property	Certain type
English DBpedia	DQP	dbo:deathPlace	dbo:Agent, dbo:Person
	RQP		dbo:Place, dbo:Wikidata:Q532, dbo:PopulatedPlace
Korean DBpedia	DQP	prop-ko:죽은곳	dbo:Agent, dbo:Person
	RQP		dbo:Place, dbo:Wikidata:Q532, dbo:PopulatedPlace

Table 4. Example test case of DQP and RQP. `prop-ko:죽은곳` is equivalent to `dbo:deathPlace`.

Total		Domain			Range			Datatype		
Triples	TC	TC	Pass	Error	TC	Pass	Error	TC	Pass	Error
1,492,331	2,452,023	1,470,389	1,075,953	394,436	613,535	176,423	437,112	368,099	309,286	58,813

Table 5. Overview of the quality assessment of the Korean DBpedia

Test Case Pattern Generation: For DQP and RQP we computed the threshold in the same way with the case of English DBpedia. In the case of TQP, we consider not only the data type but also the language tag since the Korean DBpedia use language tag instead of defining a language value as a data type of object. For instance, the value of `prop-ko:이름`, which means "name" in English, can be in the form of a string data type or having its language tag (e.g. `@ko`). The threshold ratios are about 18% and 16% for DQP and RQP respectively. We generated a set of test case patterns based on the threshold. Similar with the English DBpedia, most of the triples have a single data pattern for TQP and we identified the top one for each predicate. Table 4 shows examples of the generated test case patterns.

Data Quality Assessment: Our methodology generated 1,438 test case patterns by 1,070 properties in Korean DBpedia. It was tested against more than 1.4 million triples from Korean DBpedia. Table 5 provides an overview of the data quality assessment from the resource. Totally 2.4 million pattern matching tests were performed and about 1.4 million, 613 thousands, and 360 thousands tests were done for DQP, RQP, and TQP, respectively. Among them about 64%, 73%, and 29% of tests were passed for DQP, RQP, and TQP, respectively. This analysis has more details, which are explained in greater depth below.

4.2 Accuracy Analysis

To evaluate the accuracy of our assessment method on a localized DBpedia having no ontology, we built a data set consisting of randomly selected 1,000 triples, and employed two human evaluators to check the validity of each triple based on Wikipedia[6] data: If they found the information in a Wikipedia page, the triple will be labelled as true, otherwise false. The triples human evaluators marked as valid are considered as *actually valid* triples. Based on Krejcie et al. [22], 1,000 samples is a sufficient size to construct 95% confidence level with a margin of 3.5% of error.

We measured the inter-rater agreement value based on the Cohen's kappa measure [23] and the value between two evaluators was 0.7207. Table 6 shows the accuracy of the patterns in terms of precision, recall, and F1-measure. Precision is the ratio of actually valid triples to the set of triples determined as valid by our assessment method. On the other hand, recall is the ratio of triples assessed as valid to the actually valid triples. Also F-1 measure is defined by $\frac{2\times(precision\times recall)}{(precision+recall)}$ and it means the harmonic mean of the precision and recall performances. The average F1-measure weighed by the number of triples is about 0.7 (up to 0.79). This high accuracy scores achieved by the proposed approach well demonstate its usefulness and generality.

[6] `http://ko.wikipedia.org/`

	Triples	Precision	Recall	F1-measure
DQP	981	0.7100	0.8022	0.7533
RQP	424	0.9308	0.3438	0.5021
TQP	263	0.7395	0.8503	0.7910

Table 6. Accuracy of each pattern as precision, recall, and f1-measure.

4.3 Error Analysis on Korean DBpedia

The error occurrence rate of the total triple is 36.31%. DQP was produced for most of the triples and has an error rate of 26.83%. The highest occurrence with the most error cases in Korean DBpedia is found with the `rdf:range` violation. It is seen by error rate of RQP, which reaches over 71%, which seems high relative to other studies [3, 4, 18].

DQP and RQP are the most defined patterns in Linked Data. Therefore, quality problems related to the domain and range are very common to be found in any dataset. In previous studies [3, 4], similar problems were observed. Paticularly in our case, there are many cases where there is no definition for `rdf:type` in the Korean DBpedia data. This feature caused the data to encounter many problems while checking correct domain, range, and/or datatype for a property. For example, the `rdf:type` of `db-ko:캐나다`, which is the same as `db:Canada`, should be defined as `dbo:Country`. The `db:Canada` has `dbo:Country`, but `db-ko:캐나다` does not have any types. Moreover, in terms of range, we can not define range only by looking at object type. DBpedia triples are extracted from Wikipedia data stream as URIs or literal. At this time, the object range validation cannot be performed [4]. There are many cases in which value of the object are extracted as string or literal, not as URI. Although it is represented in a different form, the value itself has an equal meaning, but still it does not meet `rdf:range` in quality evaluation. For the reasons, which have been mentioned, the recall of RQP is much lower than the other two.

Other problems related to domain and range occurred when types were not labelled as `rdf:type` but used as string instead, particularly for range. For such cases, we classified them as a datatype problem. We found other cases of quality problem regarding datatype, i.e. incorrect datatype setting and incorrect object value. An example for the first case is when the data concerning the date must be set as `xs:date`, but it is set to `xs:integer` instead. For the second case, let's take `prop-ko:활동기간`, which means "active period" in English, as an example. The object value is a period of time but, instead of duration, only the beginning point of the duration is directly extracted from the Wikipedia page.

In the case of datatype quality, we found that quality problems occurred in two cases. First, datatype does not match the object. For instance, the object of `prop-ko:태어난곳`, which means "birth place" in English, must be within `dbo:Place` or string datatype in Korean DBpedia. However, objects of `prop-ko:태어난곳` are represented as `xs:integer`. Second, property ambiguity is a common problem. One property could have more than one meaning, which then affects its object type. This happens when the property is not represented by `rdfs:label` or `dbo:abstract`. For instance, for property `prop-ko:종목`, which means "event" (e.g. Olympic event), can have 2 totally different types of objects, i.e. the name of the event itself or the number of events. It raised another problem because we had to choose which datatype should be taken.

5 Related work

Linked data quality assessment There are a number of data quality assessment approaches for linked data. Zaveri et al. [2], classified the data quality dimensions into the accessibility, intrinsic, contextual, and representational from analyzing several approaches and tools. Quality assessment tools are typically used for semi-automatic or automatic

measurement. LINK-QA [6] is an extensible tool that allows for the evaluation of linked data mapping using network metrics. It is an automatic approach that can perform the quality assessment of the links.
On the other hand, most of the quality assessment approaches are semi-automatic. DaCura [8] is able to collect and curate evolving linked data that maintain quality over time. It requires a lot of human efforts for modifying schema involving domain experts, data harvesters, and consumers. Another framework called SWIQA [16] automatically identifies data quality by SPARQL queries which represent the quality rule. The rule is defined by analyzing the Ontology, programing knowledge is not required in this time. Also other studies proposed linked data quality assessment methods [5, 16, 17]. Even though these studies introduced useful ways to assess data quality, they all require an ontology or data schema. In our study, we semi-automatically generated patterns that are able to evaluate the quality from linked data without requiring an ontology. We used Korean DBpedia, which is one of the localized versions of DBpedia. In the following, we identified the quality problems in DBpedia resources, also automatic ontology generation methods, related to our approach.
Data Quality Assessment of DBpedia DBpedia is a central hub of LOD cloud. The quality problems of DBpedia were studied through manual, crowdsourcing [18] and semi-automated approaches [19]. In [3], a framework for the DBpedia quality assessment is presented. It involves manual and semi-automatic processes. In the semi-automatic phase, the framework requires the axiom, which is created by ontology learning [20] or manual verification. Another study classified more details about quality problems [4]. In this study, 12 data quality test patterns were created from DBpedia user community feedback, Wikipedia maintenance system and ontology analysis. Most of local DBpedia do not have ontology, but have similar data formation. Therefore we have devised an approach for the quality assessment of data by paying attention to this research.
Automatic Ontology generation Traditionally, ontology is generated by domain experts. However, building an ontology for a huge amount of data is a difficult and time consuming task. Consequently, there are several studies on the automatic ontology generation. Text2Onto [14] is an ontology learning framework from textual data by representing the learned knowledge at a meta-level in the form of Probabilistic Ontology Model. The framework calculates a confidence score for each learned object and it also allows a user to trace the evolution of the ontology. The framework extracts ontologies from language texts by employing natural language processing. As such, the framework is limited by languages - it only supports English, Spanish, and German texts.
Sie and Yeh's study [13] combines the results of specific knowledge network and automatic ontology generation from metadata. This approach builds digital libraries that have metadata documents and schema information. Another study generated OWL ontology automatically from XML [21]. Those approaches generated ontologies from data schemas, which are not available in our localized DBpedia. Recently, Pilehvar and Navigli's work [24] addressed the alignment of an arbitrary pair of lexical resources independently of their specific schema. They proposed to induce a network structure for dictionary resources, however textual similarity remains an important component of their approach.

6 Conclusion and Future work

In this paper, we proposed an approach for evaluating the quality of linked data without requiring the use of ontology. The approach semi-automatically generates patterns from a knowledge resource without using any data schema or ontology. Pattern is a structure that is derived from data. Patterns are then instantiated into test cases to measure the quality of data in terms of domain, range, and datatype of a property. We evaluated our approach going through two phases. First, we compared the patterns generated without using

ontology with the existing benchmark patterns that were generated by using ontology. We used dataset from English DBpedia. The consistency between the generated patterns and the existing patterns are high (89.35% for DQP, 80.33% for RQP, and 67.74% for DTQP). Second, we applied our approach to evaluating data quality using a localized DBpedia, which does not have ontology. We used Korean DBpedia as example of localized DBpedia. Our approach generated 1,438 test case patterns from Korean DBpedia. We evaluated the quality of over 1.4 million triples in the resource by using patterns generated by our approach. Through the evaluation results, we found several problems that are caused by the lack of schema, as well as the problems of data itself.
The current approaches for assessing the quality of linked data are only possible with the presence of data schema or ontology. This work is the first step of developing an approach for evaluating data quality without requiring such data schema when automatic generation of ontology is difficult. Further research is needed in order to conduct full-scale evaluation of the potential of the proposed approach. Further, we plan to evaluate data quality problems, caused by a lack of schema, by utilizing external resources (e.g. WordNet, Thesaurus). We are also looking for more varied patterns that can be applied to quality assessment. Finally, we plan to not only improve the quality assessment, but also to create a complete validation system for determining trustworthiness of triples. Notwithstanding these limitations, however, the current findings clearly show that the proposed approach opens a new possibility of conducting quality assessment when the knowledge resource that lacks a well developed ontology has to be used.

Acknowledgments

This work was supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIP) (No. R0101-15-0054, WiseKB: Big data based self-evolving knowledge base and reasoning platform)

References

1. Batini, C., Cappiello, C., Francalanci, C., Maurino, A. Methodologies for data quality assessment and improvement. ACM Computing Surveys (CSUR), 41(3), 16 (2009)
2. Zaveri, A., Rula, A., Maurino, A., Pietrobon, R., Lehmann, J., Auer, S., Hitzler, P. Quality assessment methodologies for linked open data. Submitted to Semantic Web Journal (2013)
3. Zaveri, A., Kontokostas, D., Sherif, M. A., Bühmann, L., Morsey, M., Auer, S., Lehmann, J. User-driven quality evaluation of dbpedia. In Proceedings of the 9th International Conference on Semantic Systems (pp. 97-104). ACM (2013)
4. Kontokostas, D., Westphal, P., Auer, S., Hellmann, S., Lehmann, J., Cornelissen, R., Zaveri, A. Test-driven evaluation of linked data quality. In Proceedings of the 23rd international conference on World Wide Web (pp. 747-758). ACM (2014)
5. Hogan, A., Harth, A., Passant, A., Decker, S., Polleres, A. Weaving the pedantic web (2010)
6. Guéret, Christophe and Groth, Paul and Stadler, Claus and Lehmann, Jens. Assessing linked data mappings using network measures. In The Semantic Web: Research and Applications (pp. 87-102). Springer Berlin Heidelberg (2012)
7. Bizer, Christian and Cyganiak, Richard. Quality-driven information filtering using the WIQA policy framework. Web Semantics: Science, Services and Agents on the World Wide Web, 7(1), 1-10 (2009)
8. Feeney, Kevin Chekov and O'Sullivan, Declan and Tai, Wei and Brennan, Rob. Improving curated web-data quality with structured harvesting and assessment. International Journal on Semantic Web and Information Systems (IJSWIS), 10(2), 35-62 (2014)
9. Bedini, I., Nguyen, B. Automatic ontology generation: State of the art. PRiSM Laboratory Technical Report. University of Versailles (2007)
10. Wang, Richard Y and Strong, Diane M. Beyond accuracy: What data quality means to data consumers. Journal of management information systems, 5-33 (1996)

11. Juran, Joseph and Godfrey, A Blanton. Quality handbook. Republished McGraw-Hill (1999)
12. Wächter, Thomas and Fabian, Götz and Schroeder, Michael. DOG4DAG: semi-automated ontology generation in obo-edit and protégé. In Proceedings of the 4th International Workshop on Semantic Web Applications and Tools for the Life Sciences (pp. 119-120). ACM (2011)
13. Sie, Shun-hong and Yeh, Jian-hua. Automatic ontology generation using schema information. In Web Intelligence, 2006. WI 2006. IEEE/WIC/ACM International Conference on (pp. 526-531). IEEE (2006)
14. Cimiano, Philipp and Völker, Johanna. Text2Onto. In Natural language processing and information systems (pp. 227-238). Springer Berlin Heidelberg (2005)
15. Wienand, Dominik and Paulheim, Heiko. Detecting incorrect numerical data in dbpedia. In The Semantic Web: Trends and Challenges (pp. 504-518). Springer International Publishing (2014)
16. Fürber, Christian and Hepp, Martin. Swiqa-a semantic web information quality assessment framework. In ECIS (Vol. 15, p. 19) (2011)
17. Fürber, Christian and Hepp, Martin. Using semantic web resources for data quality management. In Knowledge Engineering and Management by the Masses (pp. 211-225). Springer Berlin Heidelberg (2010)
18. Acosta, Maribel and Zaveri, Amrapali and Simperl, Elena and Kontokostas, Dimitris and Auer, Sören and Lehmann, Jens. Crowdsourcing linked data quality assessment. In The Semantic Web–ISWC 2013 (pp. 260-276). Springer Berlin Heidelberg (2013)
19. Wienand, Dominik and Paulheim, Heiko. Detecting incorrect numerical data in dbpedia. In The Semantic Web: Trends and Challenges (pp. 504-518). Springer International Publishing (2014)
20. Lehmann, J. DL-Learner: learning concepts in description logics. The Journal of Machine Learning Research, 10, 2639-2642 (2009)
21. Yahia, Nora and Mokhtar, Sahar A and Ahmed, AbdelWahab. Automatic generation of OWL ontology from XML data source. arXiv preprint arXiv:1206.0570 (2012)
22. Krejcie, Robert V and Morgan, Daryle W. Determining sample size for research activities. Educ Psychol Meas (1970)
23. Cohen, Jacob. A coefficient of agreement for nominal scales. Educational and psychological measurement 20.1 : 37-46 (1960)
24. Pilehvar, Mohammad Taher, and Roberto Navigli. A robust approach to aligning heterogeneous lexical resources. A← A 1 (2014): c2.