Linked Data Quality

Subhi Issa

Conservatoire National des Arts et Métiers, CEDRIC,
subhi.issa@cnam.fr

Abstract. The wides pread of semantic web technologies such as RDF, SPARQL and OWL enables individuals to build their databases on the web, write vocabularies, and define rules to arrange and explain the relationships between data according to the Linked Data principles. As a consequence, a large amount of structured and interlinked data is being generated daily. A close examination of the quality of this data could be very critical, especially if important researches and professional decisions depend on it. Several linked data quality metrics have been proposed, and they cover numerous dimensions of linked data quality such as completeness, consistency, conciseness and interlinking. In this work, we are interested in linked data quality dimensions, especially the completeness and conciseness of linked datasets. A set of experiments were conducted on a real-world dataset (DBpedia) to evaluate our proposed approaches.

Keywords: LOD, Linked Data Quality, Completeness, Conciseness

1 Problem Statement

Because a large amount of information is being generated daily, and information needs to be of high quality to be useful, the need for quality assessment of this data on the internet is more urgent ever before. On the other hand, Linked Open Data [1] (LOD) has appeared as a result of the development of semantic web technologies, such as RDF, SPARQL and OWL. A research of the quality of information has been successfully applied on the traditional information system, with the rational databases having a positive impact on the used organizational processes [3]. This raises the question of the applicability of this approach in the context of web of data. Zaveri et al. [17] surveyed 18 different linked data quality dimensions that can be applied to assess the quality of Linked Data. The goal of this work is to propose approaches that focus on figuring whether this information completely represents the real world and that is logically consistent in itself. Our objective is not measuring an absolute completeness and conciseness but rather measuring their aspects.

1.1 Completeness

Completeness is a data quality measure that refers to the degree to which all required information is present in a particular dataset [17]. We illustrate in

[1] http://5stardata.info/en/

this section the main idea behind our approach through an example that shows the issues and the difficulties encountered in the calculation of a dataset completeness. Let us consider the set of scientists described in the well-known open linked dataset, DBpedia. We would like to calculate the completeness of a scientist description (e.g. *Albert_Einstein*), which will be the proportion of properties used in the description of this scientist to the total number of properties in *Scientist_Schema*. In DBpedia, the *Scientist*[2] class has a list of 4 properties (e.g. *doctoralAdvisor*), but these properties are not the only ones used in the description of a scientist (e.g. the *birthdate* property is not present in this list). Indeed, the *Scientist* class has a super class called *Person*. So, the description of a scientist may also take into account properties of the *Scientist* class and all its ancestors.

$$\begin{aligned}&Scientist\_Schema = \{Properties\ on\ Scientist\} \cup\\ &\{Properties\ on\ Person\} \cup \{Properties\ on\ Agent\} \cup\\ &\{Properties\ on\ Thing\}\end{aligned}$$

$$\text{such that: } Scientist \sqsubseteq Person \sqsubseteq Agent \sqsubseteq Thing$$

However, we can obtain the size of *Scientist_Schema*, which is equal to 664 (A-Box properties) in the case of DBpedia with a simple SPARQL query[3]. Thus, the completeness of the description of *Albert_Einstein* could be calculated as follows:

$$\begin{aligned}Comp(Albert\_Einstein) &= \frac{|Properties\ on\ Albert\_Einstein|}{|Scientist\_Schema|}\\ &= \frac{21}{664} = 3,16\%\end{aligned}$$

Although, the property *weapon* is in *Scientist_Schema*, but it is not relevant for the *Albert_Einstein* instance. As a conclusion, we can finally say that the completeness as calculated here does not provide us with the relevant value regarding the real representation of scientists in the DBpedia dataset. Hence, we need to overcome this issue by inventing and exploring to get an idea about how they are actually described and which properties are used. In [2], the authors propose a new approach to compute the completeness of instances based on similar ones in Wikidata. For each instance, they find the most frequent properties among instances that have the same type, and find the percentage of missed properties to calculate the completeness. This approach sometimes does not work well when the instance has several values of the property *instance of* as a class such as Writer and Player.

1.2 Conciseness

Conciseness is one aspect of linked data quality dimension, which basically aims to avoid repetition through elements having the same meaning with differ-

[2] `http://mappings.dbpedia.org/server/ontology/classes/`

[3] Performed on: `http://dbpedia.org/sparql`

ent identifiers or names. The eliminating of the synonymously used predicates aims to optimize the dataset to speed up processing.

Our research on conciseness dimension was inspired by the existing Synonym Analysis for Predicate Expansion [1]. However, Abedjan et Naumann proposed a data-driven synonym discovery algorithm for a predicate expansion by applying both schema analysis and range content filtering.

Range content filtering aims to represent a transaction as a distinct object with several predicates. For example, the object *Lyon* city is connected with several predicates such as (*birthPlace*, *deathPlace* and *location*). The authors suppose that synonym predicates have a similar sense. They also share a similar group of object values. For this reason, the proposed approach finds that the frequent sets pattern of predicates is dominated by object values.

Thus, it is not sufficient to discover the predicates that are used synonymously depending on Range Content Filtering alone. For example, the predicates *birthPlace* and *deathPlace* share the significant co-occurrences with the same object values but they are definitely used differently. However, the authors have proposed another filter in order to overcome this problem and to find the synonym predicates more correctly. They expect that the synonym predicates should not co-exist for the same instance. According to schema analysis, transactions of distinct subjects with several predicates are represented. By applying negative association rules, the synonym predicates appear in different transactions. For instance, the subject *Michael_Schumacher* does not have two synonymously used predicates such as *born* and *birthPlace* in the same dataset.

Now, our objective is to discover synonym predicates by applying the proposed approach. We clarify its drawbacks through applying the next example (see Table 1), and we would like to apply the previous approaches on a sample of facts from DBpedia to discover the synonym predicates.

Based on range content filtering, all predicates will be gathered into groups by each distinct object. Thus, results can be as illustrated in Table 2 in order to retrieve the frequent candidates. As a result, we can see that the *nationality* and *sourceCountry* predicates are already in the same transaction. By applying FP-growth algorithm [8], frequent itemsets have been mined, thus *nationality* and *sourceCountry* are the consequences. The next step is applying schema analysis as a subject in a context and we will get the following transactions (see Table 3). We can notice that by applying negative association rules, there is no co-occurrence between *sourceCountry* and *nationality* predicates.

Subject	Predicate	Object
Adam_Hadwin	*type*	*GolfPlayer*
Adam_Hadwin	*birthPlace*	$Moose_jaw$
Adam_Hadwin	*nationality*	*Canada*
White_River	*sourceCountry*	*Canada*
White_River	*riverMouth*	$Lake_superior$
White_River	*state*	*Ontario*

Table 1: Facts in SPO structure from DBpedia

Object	Predicate
GolfPlayer	*type*
$Moose_jaw$	*birthPlace*
Canada	*nationality, sourceCountry*
$Lake_superior$	*riverMouth*
Ontario	*state*

Table 2: Range Content Filtering

Subject	Predicate
Adam_Hadwin	*type, birthPlace , nationality*
White_River	*sourceCountry,riverMouth,state*

Table 3: Schema analysis

Therefore, the algorithm proposed the *nationality* and *sourceCountry* as synonym predicate pairs, which is not correct because we cannot replace *nationality* predicate that is related to *Person* class as its Domain with *sourceCountry* predicate, which is related to *Stream* class as its Domain.

2 Relevancy

Nowadays we are witnessing an increase in data accessible on the internet. There are large amounts of data being generated daily. It plays a crucial role in companies, organization and individual decisions. This data, although rich in content, is often incomplete, lacks metadata or even suffers from redundancy. As our goal is to improve Linked Data quality, the problem is relevant for Linked Data publishers, contributors and consumers. Users look forward to getting information with a high quality which means that data is "fitness to use" [11].

3 Related work

Several metrics and tools have been proposed to assess Linked Data and improve its quality [4,13,12]. Unfortunately, there were obstacles related to the absence of a clear definition of the word "Quality" since it has different meaning from a domain to another. However, data quality is commonly conceived to suite our use so that it has several aspects or dimensions, such as accuracy, completeness and interlinking. In 2016, Zaveri et al. [17] identified a set of 18 different data quality dimensions, each dimension has at least one indicator or a metric to assess the given dimension. Some of the proposed approaches deal with one dimension [7,15] or several dimensions [13,12].

Completeness is one of the essential dimensions of data quality, which refers to the amount of the presented information. Pipino et al. [14] divided completeness into: Schema completeness that is the degree to which classes and properties are not missing in a schema, property completeness which is the extent of the missing property values of a specific kind of property, and population completeness that refers to the ratio of objects represented to real-world objects. Since several works provide metrics for the three completeness classifications [13,5], their defined metrics evaluate the completeness by comparing it with a predefined schema that could not provide an accurate value of dataset completeness.

On the other hand, Mendes et al. [13] categorized conciseness dimension into intensional and extensional conciseness. The first type, which is the intensional conciseness, measures a number of unique dataset elements to the total number of schema elements, thus this measurement is represented on the schema level. In a similar manner but on the instance level, extensional conciseness measures the number of unique objects to the real number of objects in the dataset. In the similar sense but with another naming "uniqueness", Füber et Hepp [5] defined the elements of representation like classes, properties and objects. Their definition suggested uniqueness of breadth at the schema level and uniqueness of depth at the instance level. In [1], the authors proposed an algorithm to discover the synonym predicates for query expansions. They depended on mining similar predicates according to their subjects and objects. However, their approach works well when dealing with a dataset that has a limited number of instances.

Our goal is to enhance the dimensions of linked data quality that do not have enough metrics (i.e. completeness and conciseness). We aim to propose new metrics from different perspectives such as inferring a reference schema from data source and using semantic analysis to understand the meaning of predicates.

4 Research questions

Completeness calculation requires a reference schema to be compared with. The gold-standard or predefined schema does not always represent a good reference. So, there is a need to explore instances to have a suitable reference schema (ontology). Also, dataset ontology contains semantic features which represent an explanation of the meaning of each predicate.

- **Completeness dimension:** Is it possible to calculate completeness values using inferred schema from data source? How can we assess the completeness of Linked Data?
- **Conciseness dimension:** Can we enhance the conciseness dimension of liked datasets by analyzing the semantics of predicates?

5 Hypotheses

Our hypotheses are derived directly from the questions above:

H1 Exploring instances to get an idea about how they are actually described and which properties, besides considering the importance of each one, are used. This provides more suitable schema to use as a reference one in order to calculate completeness value of a dataset.

H2 A deep semantic analysis of data, beside to the statistical analysis, can enhance the conciseness of linked datasets by discovering repeated predicates. Where the semantic analysis will reduce the false positive results.

6 Preliminary results

6.1 Completeness assessment

On the basis of our belief that a suitable schema (e.g. a set of properties) needs to be inferred from the data source, the experiments were performed on the well-known real-world datasets, DBpedia, publicly available on the Linked Open Data (LOD). DBpedia, is a large knowledge base composed of structured information extracted collaboratively from Wikipedia. It describes currently about 14 million things.

In [10], for evaluating the completeness of different versions of DBpedia, we chose three relatively distant versions. The first one (v3.6) was generated in March/April 2013, the second one (v2015-04) in February/March 2015 and the third one (v2016-10) in October 2016. For each dataset, we have chosen a couple of classes from different natures. We studied the completeness of resources that have classes as the following ones: $C = \{Film, Organisation, Scientist, PopulatedPlace\}$. For the properties used in the resources descriptions, we have chosen the English datasets "mapping-based properties", "instance types" and "labels".

The experiments revealed that datasets completeness could increase or decrease due to changes made to existing data or to the new added data. We also noticed that often this evolution does not benefit from the initial data cleaning as the set of properties continue evolving over time. Our approach could be helpful for data source providers to improve, or at least to keep a certain completeness of their datasets over different versions. It could be particularly useful for datasets constructed collaboratively by applying some rules for contributors when they update or add new resources.

6.2 Conciseness assessment

The automatic generation of RDF knowledge bases might lead to several semantic and syntactic errors in addition to incomplete metadata.

Because publisher commonly do not respect the semantics including misuses of ontological term and undefined classes and properties [9], this leads to the lack of semantic features in DBpedia ontology as illustrated in Table 4, only the property domain-range restrictions can be applied. Unfortunately, only 30 functional properties have been defined. Furthermore, DBpedia ontology neither defines min and max cardinality nor the functional predicates nor transitive properties nor the symmetric ones. In addition, we noted that according to the last version of DBpedia (October-2016), 16.3% of predicates are represented without domains and 10.2% of predicates are without ranges in DBpedia ontology. For this reason, based on the approach that has been proposed in [16], we infer missed domains (and/or ranges) of predicates in DBpedia ontology. In case of instances which have more than one *rdf:type*, only the class with the highest value will be defined as the domain (or range) of the property if this value is greater than a selected threshold. When the highest value is smaller than the threshold, *owl:Thing* will be selected as the domain (or range). We applied our approach on the last version of DBpedia ontology v2016-10. Table 5 shows top 10 results from the DBpedia dataset ranked by schema analysis. We chose support thresholds 0.1% for the content filtering part.

Feature	existence
Domain	83.7%
Range	89.8%
Functional properties	1%
Transitive properties	0%
symmetric properties	0%
max cardinality	0%
min cardinality	0%

Table 4: Characteristics predicates of DBpedia dataset (v10-2016)

Predicate 1	Predicate 2
musicComposer	composer
author	writer
creator	author
starring	artist
city	locationCity
education	almaMater
headquarter	city
occupation	personFunction
musicComposer	musicalArtist
musicBy	musicComposer

Table 5: Top 10 matched predicate pairs

7 Approach

According to the research questions and the hypotheses formulated, we address two aspects of linked data quality dimensions.

7.1 Completeness assessment

We represent, in this section, our approach [10] that addresses a completeness aspect of linked data by posing the problem as an itemset mining problem. In fact, the completeness at the data level assesses missing values [14]. This vision requires a schema (e.g. a set of properties) that needs to be inferred from the data source. However, it is not relevant to be considered for a subset of resources. However, the schema is as the union of all properties used in their description as seen in Section 1.1. Indeed, this vision neglects the fact that missing values can express inapplicability.

Our mining-based approach includes two steps:

1. **Properties mining**: Given a dataset $\mathcal{D}$, we first represent the properties, used for the description of the $\mathcal{D}$ instances, as a transaction vector. We then apply the well-known FP-growth algorithm [8] for mining frequent itemsets (we chose FP-growth for efficiency reasons, any other itemset mining algorithm could obviously be used). Only a subset of these frequent itemsets, called "Maximal" [6], is captured. This choice is motivated by the fact that, on one hand, we are interested in important properties for a given class that should appear often, and on the other hand, the number of frequent patterns could be exponential when the transaction vector is very large.
2. **Completeness calculation**: Once the set of maximal frequent itemsets $\mathcal{MFP}$ is generated, we use the apparition frequency of items (properties) in $\mathcal{MFP}$ to give each of them a weight that reflects how important the set of properties is considered for the description of instances. Weights are then exploited to calculate the completeness of each transaction (regarding the presence or absence of properties) and, hence, the completeness of the whole dataset.

   Definition 1. *(*Completeness*) Let $\mathcal{I}'$ a subset of instances, $\mathcal{T}$ the set of transactions constructed from $\mathcal{I}'$, and $\mathcal{MFP}$ a set of maximal frequent pattern. The completeness of $\mathcal{I}'$ corresponds to the completeness of its transaction vector $\mathcal{T}$ obtained by calculating the average of the completeness of $\mathcal{T}$ regarding each pattern in $\mathcal{MFP}$. Therefore, we define the completeness $\mathcal{CP}$ of a subset of instance $\mathcal{I}'$ as follows:*

$$\mathcal{CP}(\mathcal{I}') = \frac{1}{|\mathcal{T}|} \sum_{k=1}^{|\mathcal{T}|} \sum_{j=1}^{|\mathcal{MFP}|} \frac{\delta(E(t_k), \hat{P}_j)}{|\mathcal{MFP}|} \tag{1}$$

$$\textit{such that: } \hat{P}_j \in \mathcal{MFP}, \textit{ and } \delta(E(t_k), \hat{P}_j) = \begin{cases} 1 \textit{ if } \hat{P}_j \subset E(t_k) \\ 0 \textit{ otherwise} \end{cases}$$

7.2 Conciseness assessment

The objective of the semantic analysis is to find the meaning of the predicate. Depending on systematic analysis alone is not sufficient to discover the synonymously used predicates, also too many false positive results are represented, especially when we deal with a large dataset. As the previous example illustrated in Section 1.2, the predicates *nationality* and *sourceCountry* can have the same object like *Canada*. They also never appear or co-occur together for the same subject. Obviously, the *nationality* is a predicate of *Person* class and *sourceCountry* is a predicate of *Stream* class.

We add an important extension to the previous work by studying the meaning of each candidate. In addition, we study some conditions to examine them exploring their meanings so that we mathematically prove on a basis of Description Logic that a predicate cannot be a synonym of another predicate if they have disjoint domains or ranges. Through taking the same previous example of *nationality* and *sourceCountry* predicates, we will analyze the domain and range of each one of them. On one hand, the predicate *nationality* has a domain as *Person* class and a range as *Country* class, and on the other hand, the predicate *sourceCountry* has a domain as *Stream* class and a range as *Country* class. According to DBpedia ontology *Stream* class is a subclass of *Place* class, as well as *Place* and *Person* classes are completely disjointed. Consequently, we cannot consider *nationality* and *sourceCountry* as synonym predicates.

To promote our arguments, we will prove that a predicate cannot be a synonym of other predicate in some cases according to the semantic features of each one, such as: Disjoint properties based on their domains and ranges, Symmetric/Asymmetric Property, Inverse Functional property, Functional property and max cardinality. We illustrate these arguments using Description Logic formalization.

8 Evaluation plan

Our goal is to compare the completeness and conciseness of dataset with our presented approach to the state of the art such as Sieve [13]. In the future, we plan to enrich our investigation with other data sources such as Yago, IMDB, etc. In addition, for conciseness we would compare our approach to the Abedjan approach [1] to judge the importance of the semantic analysis, we aim to prove that the excluded candidates cannot be synonyms predicates. As the results show that DBpedia dataset misses lots of metadata, we plan to find an approach to infer the features of the predicates since we believe that we can help to improve the use of the semantic part.

9 Reflections

Since we believe that poor-quality data affects negatively on the decision that can lead to catastrophic consequences, improving the quality of linked data is our research main aim because Web of Data is worthlessness without good quality. We are concerned to concentrate on two dimensions, which do not have a lot of metrics or indications to be evaluated, according to what Zaveri suggested. We believe that extracting a reference schema from data source is more suitable to calculate the completeness of dataset, beside, we prove the importance of semantic part in addition to the statistical one to enhance the conciseness of dataset. Our proposed approach takes into account only properties disjoint and functional proprieties because semantic features are not sufficiently present as explained in Section 6.2. Therefore, our plan is to infer all possible semantic features of LOD datasets too.

Acknowledgements I would like to thank my supervisors Prof. Dr. Samira Si-said Cherfi and Dr. Fayçal Hamdi for their support and the opportunity for the realization of this work.

References

1. Abedjan, Z., Naumann, F.: Synonym analysis for predicate expansion. In: Extended Semantic Web Conference. pp. 140–154. Springer (2013)
2. Balaraman, V., Razniewski, S., Nutt, W.: Recoin: Relative completeness in wikidata. In: Companion of the The Web Conference 2018 on The Web Conference 2018. pp. 1787–1792. International World Wide Web Conferences Steering Committee (2018)
3. Batini, C., Cappiello, C., Francalanci, C., Maurino, A.: Methodologies for data quality assessment and improvement. ACM Computing Surveys (CSUR) 41(3), 16 (2009)
4. Debattista, J., Auer, S., Lange, C.: Luzzu–a framework for linked data quality assessment. In: Semantic Computing (ICSC), 2016 IEEE Tenth International Conference on. pp. 124–131. IEEE (2016)
5. Fürber, C., Hepp, M.: Swiqa-a semantic web information quality assessment framework. In: ECIS. vol. 15, p. 19 (2011)
6. Grahne, G., Zhu, J.: Efficiently using prefix-trees in mining frequent itemsets. In: Proceedings of the ICDM 2003 Workshop on Frequent Itemset Mining Implementations, 19 December 2003, Melbourne, Florida, USA (2003)
7. Guéret, C., Groth, P., Stadler, C., Lehmann, J.: Assessing linked data mappings using network measures. In: Extended Semantic Web Conference. pp. 87–102. Springer (2012)
8. Han, J., Pei, J., Yin, Y., Mao, R.: Mining frequent patterns without candidate generation: A frequent-pattern tree approach. Data Min. Knowl. Discov. 8(1), 53–87 (Jan 2004)
9. Hogan, A., Harth, A., Passant, A., Decker, S., Polleres, A.: Weaving the pedantic web. LDOW 628 (2010)
10. Issa, S., Paris, P., Hamdi, F.: Assessing the completeness evolution of dbpedia: A case study. In: Advances in Conceptual Modeling - ER 2017 Workshops AHA, MoBiD, MREBA, OntoCom, and QMMQ, Valencia, Spain, November 6-9, 2017, Proceedings. pp. 238–247 (2017)
11. Joseph M., J., Richard S., B., Frank M., G.: The Quality Control Handbook. Rainbow-Bridge, 3 edn. (1974)
12. 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)
13. Mendes, P.N., Mühleisen, H., Bizer, C.: Sieve: linked data quality assessment and fusion. In: Proceedings of the 2012 Joint EDBT/ICDT Workshops. pp. 116–123. ACM (2012)
14. Pipino, L.L., Lee, Y.W., Wang, R.Y.: Data quality assessment. Communications of the ACM 45(4), 211–218 (2002)
15. Ruckhaus, E., Baldizán, O., Vidal, M.E.: Analyzing linked data quality with liquate. In: OTM Confederated International Conferences" On the Move to Meaningful Internet Systems". pp. 629–638. Springer (2013)
16. Töpper, G., Knuth, M., Sack, H.: Dbpedia ontology enrichment for inconsistency detection. In: Proceedings of the 8th International Conference on Semantic Systems. pp. 33–40. ACM (2012)
17. Zaveri, A., Rula, A., Maurino, A., Pietrobon, R., Lehmann, J., Auer, S.: Quality assessment for linked data: A survey. Semantic Web 7(1), 63–93 (2016)