Using Provenance for Quality Assessment and Repair in Linked Open Data

Giorgos Flouris[1], Yannis Roussakis[1], María Poveda-Villalón[2], Pablo N. Mendes[3], and Irini Fundulaki[1,4]

[1] FORTH-ICS, Greece, [2] UPM, Spain, [3] FUB, Germany, [4] CWI, The Netherlands
{fgeo,rousakis,fundul}@ics.forth.gr, mpoveda@fi.upm.es,
pablo.mendes@fu-berlin.de

Abstract. As the number of data sources publishing their data on the Web of Data is growing, we are experiencing an immense growth of the Linked Open Data cloud. The lack of control on the published sources, which could be untrustworthy or unreliable, along with their dynamic nature that often invalidates links and causes conflicts or other discrepancies, could lead to poor quality data. In order to judge data quality, a number of quality indicators have been proposed, coupled with quality metrics that quantify the "quality level" of a dataset. In addition to the above, some approaches address how to improve the quality of the datasets through a repair process that focuses on how to correct invalidities caused by constraint violations by either removing or adding triples. In this paper we argue that provenance is a critical factor that should be taken into account during repairs to ensure that the most reliable data is kept. Based on this idea, we propose quality metrics that take into account provenance and evaluate their applicability as repair guidelines in a particular data fusion setting.

1 Introduction

The Linked Open Data (LOD) cloud is experiencing rapid growth since its conception in 2007. Hundreds of interlinked datasets compose a knowledge space which currently consists of more than 31 billion RDF triples. In this setting, data constantly evolves, invalidating previous links between datasets and causing quality problems; similarly, changes in the world itself are not simultaneously reflected in all related datasets, causing conflicts and other discrepancies among overlapping datasets during data fusion.

Such data quality problems come in different flavors, including duplicate triples, conflicting, inaccurate, untrustworthy or outdated information, inconsistencies, invalidities and others [23,1,14], and cost businesses several billions of dollars each year [7]. Therefore, improving the quality of datasets in an evolving LOD cloud is crucial.

Quality is generally defined as *fitness for use* [14]. Therefore, the interpretation of the quality of some data item depends on who will use this information, and what is the task for which they intend to employ it. While one user may consider the data quality sufficient for a given task, it may not be sufficient for another task or another user. Thus, it has been argued that the concept of quality is multi-dimensional, as well as context- and application-specific [23]. To assess quality, a non-exhaustive list of *quality dimensions* such as timeliness, trustworthiness, conciseness and validity has been considered in [23,1,26]. A conceptual model for quality assessment is described, that is composed of *quality indicators*, *quality assessment metrics* and *scoring functions* that quantify the quality of a dataset along a given (set of) quality dimension(s) [26].

Apart from evaluating a dataset's quality, it is also important to improve it, i.e., *repair* the dataset. We focus on evaluating automated repair methods [24,28] which rely on a set of *preferences* that can be used as "guidelines" by the system to determine how to resolve quality problems. For example, in the case of conflicting information, one could opt to keep the most recent information to resolve the conflict (in accordance to the Principle of Primacy of New Information often employed in evolution settings [17]). The purpose of this paper is to determine *how provenance can help in devising useful preferences for improving the quality of LOD datasets*.

Provenance refers to the origin or source of a piece of data and encodes *from where* and *how* the piece of data was obtained [32]. Provenance is of paramount importance, as in some cases it is considered more important than the data itself [8]. It is essential in many applications, as it allows to effectively support trust mechanisms, digital rights and privacy policies, and is also a means to assess the reliability of information; we exploit the latter property to devise useful metrics for quality assessment and repair.

We focus on *validity*, which requires that the dataset conforms to a set of custom constraints, expressed as logical rules. Validity is one of the most flexible and important metrics, because it encodes context- and application-specific requirements. It has been used in different contexts to express constraints like transitivity or functionality of properties, cardinality constraints, foreign key constraints etc [3,18,25,30].

Previous work on repairing LOD datasets has focused on preferences related to either the data itself [28], or to the metadata thereof (called *fusion functions*) [24]. This paper combines the repairing algorithm of [28] and the approach advocated in [24] and proposes the use of complex preferences that consider both data and metadata with an emphasis on provenance to *evaluate how this combination performs in a real setting where data from disparate sources are fused to produce a LOD dataset*. In a nutshell, the main contributions of this paper are the following:

- The description of a set of provenance-based assessment metrics (Section 4).
- The extension of the repairing algorithm in [28] to support preferences on provenance and other metadata (Section 5).
- The evaluation of the extended repairing algorithm under the proposed preferences in a LOD fusion setting (Section 6).

2 Motivating Example

For illustration purposes, we consider a user who wants to find information about Brazilian cities by fusing DBpedia dumps from different languages. Dumps overlap, have different coverage, and not all of them contain correct or up-to-date information. Therefore, the result of fusion may contain redundant, conflicting or inconsistent information (cf. Section 6), i.e., data of poor quality. To improve data quality, one could remove conflicting information keeping one value for each city. Given that the user has no access to some authoritative data source (if he did, then searching in the DBpedias would be unnecessary), the only way to choose the correct value is to use heuristics, expressed as preferences, that determine the most reliable information. Such preferences could involve the trustworthiness of information (based on provenance metadata), recency, common sense, or some combination of the above metrics.

Doing so manually may be difficult or impossible, given that DBpedia contains 1400 conflicts of this type for Brazilian cities alone (cf. Section 6). Thus, it makes more sense to use an automated repairing algorithm such as those proposed in [28,24] to perform the repairing based on user preferences. Unfortunately, existing automated approaches for repairing LOD datasets are restricted in the kind of preferences they support. In this paper, we show how existing approaches can be extended and employed to perform this task, and evaluate our methods using the above fused dataset on Brazilian cities.

3 Preliminaries

3.1 Quality Assessment

Data quality is a multi-dimensional issue, whose exact definition depends on the dataset's expected use [14]. To model these facts, a list of *quality dimensions* were collected in [23,1,26] to capture the most common aspects of data quality, such as timeliness, verifiability, completeness, relevancy, validity etc. In addition, [23,26] proposed a generic quality assessment methodology based on associating each dataset with a numerical value that represents its quality along the given dimension(s). This framework is based on *quality indicators*, *scoring functions* and *quality metrics*.

A *quality indicator* is an aspect of the considered dataset that indicates the suitability of the data for some intended use. Indicators are very diverse and can capture several types of information, including both the information to be assessed itself, and its metadata. As an example, the last modification date can be used as a quality indicator of "freshness" (related to the timeliness dimension). A *scoring function* is a numerical assessment representing the suitability of the data for the intended use, as determined by the quality indicator. Continuing the above example, a scoring function for freshness could return the number of days between two specified dates. *Quality assessment metrics* (or metrics for short) are used to measure information quality. Essentially, metrics combine information from various quality indicators and scoring functions to determine a numerical value that represents the quality of the data.

3.2 Quality Repair

Our repairing approach is based on [28] where data violating one or more *validity rules*, expressed in formal logic, are repaired in a manner that respects a set of user-defined specifications expressed as *preferences*. Due to space considerations, we will only briefly describe that approach here, and refer the reader to [28] for details.

The approach of [28] considers DED rules (Disjunctive Embedded Dependencies) [5], which are rules expressed in first-order logic that can capture several types of constraints, including transitivity and functionality of properties, cardinality constraints etc. In [28], a simple methodology for identifying DED violations and the possible ways to resolve them is described. A violation can usually be resolved in several different manners, which necessitates the use of *preferences* to determine the optimal repair solution.

Preferences are based on the idea of formal preference models employed by the database community [11], and are declarative specifications allowing one to describe the ideal solution in an abstract manner (e.g., “minimum number of schema changes”). Such a specification can be used by the system to automatically determine the preferred resolution by comparing each of the options against the “ideal” one. Formally, preferences are based on *features* which are functions that assess the repairing options under some dimension relevant for the preference (e.g., “number of schema changes”); features are then either minimized or maximized, and combined using operators to form more complex preferences. Due to space limitations, the reader is referred to [28] for more details on the formal specification of preferences; more examples of preferences are given in Sections 5, 6.

It should be noted that this specification allows various complex preferences to be defined, but their applicability is limited on repairing options because features are defined upon repairing options; thus, this definition cannot support preferences taking into account the repair result and/or metadata information. This is addressed in Section 5.

4 Quality Assessment Using Provenance

We describe three basic and two complex quality assessment metrics based on both data and metadata information (e.g., provenance). The presented metrics are not the only possible ones; instead, we focused on a particular set that is applicable to our evaluation scenario. In the future, we plan to evaluate alternative scenarios/metrics.

More specifically, we use the provenance-related indicators of *source reputation*, *freshness* and *plausibility*. The first relates the quality of data with the perceived trustworthiness of its source. The second associates the quality of data with its recency, assuming that old information is more likely to be outdated and unreliable. The last is data-related and is used to identify typos, outliers or other obviously incorrect values.

Each of these indicators defines a simple quality assessment metric, but can also be combined to form more complicated ones; we describe two options below.

The first is based on the idea that freshness alone may not be an adequate metric; depending on the application, a piece of data may be up-to-date for several years (e.g., total area of a country), for a few minutes or hours (e.g, temperature), or only for a few seconds (e.g., location of a moving car). Thus, a refined metric could first determine whether a certain piece of data is outdated (depending on the application); if so, then its assessment would be based solely on freshness (assessing “how much” it is outdated); if not, then source trustworthiness should be used. We call this metric *weighted freshness*.

The second is based on the idea that the reliability of a source sometimes depends also on the data itself. In our example, we could assume that the Portuguese Wikipedia is the closest to the domain (Brazilian cities) and is likely to be better for more esoteric things (small cities), whereas the English one, being the largest and most edited, is likely to be more reliable for larger cities [31]. Thus, a sophisticated metric could take into account the data itself before determining source trustworthiness. In our example, the perceived reliability of the Portuguese Wikipedia (compared to the English one) should increase when the triple considered is related to a small city, and decrease for large ones. We will call this metric *conditional source trustworthiness*.

5 Quality Repair Using Provenance

To evaluate the use of provenance-related metadata as a means to identify the preferences that determine the optimal repairing options during a repair process, we should lift the limitations of [28] which defined the features of preferences to be applicable only on repairing options. This does not present major challenges: all we need to do is extend the features' definition. The new features can then be seamlessly added in the existing framework of [28] and used for improved quality repair. Here, we explain the ramifications of this extension and compare the extended version with the original one.

The first extension to be considered allows features to be applied on the result of the repair process rather than the repairing options. This way, the repairing process selects how to resolve invalidities based on the dataset (repair) that these choices lead to, instead of based on the choices themselves. For example, to model the preference "I want the resulting class hierarchy to have minimum depth", one should use a feature that measures the depth of the class hierarchy; such a feature makes sense to be defined on the repair, not the repairing options (which are of the form: "add/delete triple t").

The results in [28] indicate that this extension does not change the expressive power of features. Indeed, given the original dataset D and the repairing options, one can compute the repair; similarly, given D and the repair, the repairing options that led to this repair can be computed. Therefore, for any given D there is a one-to-one correspondence between a repair and the repairing options that led to it. Thus, a preference defined upon repairing options can always be equivalently rewritten into a preference defined upon the repairs, and vice-versa [28].

Note that this correspondence is true for any given D; thus, given a preference on repairs, and in order to define the equivalent one on repairing options, one should devise a complicated expression that takes into account D. The same is true for the opposite rewriting. As an example, the preference "I want the resulting class hierarchy to have minimum depth" makes more sense to be expressed upon repairs, even though it can, in theory, be expressed as a preference upon repairing options if the input data D (to be repaired) is considered. On the other hand, the preference "I want minimum number of schema changes" makes more sense to be expressed upon repairing options, even though, again, a complicated equivalent preference upon repairs that takes into account the original dataset D is possible. Thus, even though the two options are equivalent in theory, they are complementary from the practical perspective.

The second extension allows features to consider metadata related to either the input data, or the repair, or the repairing options. This is a clear and very important extension to the original definition, because it allows important features of data to be considered, such as provenance, trustworthiness, reliability, timeliness and others, thus increasing the range and variety of preferences that a user can define.

6 Evaluation

Our experiments focus on evaluating the usability of the preferences described in Section 4. To do so, we used the algorithm described in [28] to identify and repair inconsistent information found in a fused DBpedia dataset on Brazilian cities coming from five different input datasets (Wikipedias). The result of the repair was compared against an authoritative dataset (the *Gold Standard*) in order to evaluate its quality. Our results show that all preferences perform relatively well in terms of the accuracy of the repairs performed, and that complex preferences are not necessarily better than simple ones: the simple preference based on source reputation performs very well in all cases.

6.1 Experimental Setting

Both the Gold Standard and the input datasets[1] contain data about the area, population and founding date of Brazilian cities (*areaTotal*, *populationTotal* and *foundingDate* properties respectively).

The *Input datasets* contain data extracted from the English, French, German, Spanish, and Portuguese Wikipedia infoboxes. Each piece of data is associated with its provenance (recording the Wikipedia version of the corresponding source article, using RDF named graphs) and last modification date (of the corresponding article).

The *Gold Standard dataset* (GS) was obtained from an independent and official source, namely the Instituto Brasileiro de Geografia e Estatística (IBGE)[2]. We converted the data to RDF, generating a LOD dataset accessible via a SPARQL endpoint[3].

The only requirement imposed for the validity of the fused dataset is that the properties *areaTotal*, *populationTotal* and *foundingDate* are functional, i.e., that each city should have at most one value for its area size, population and founding date. This can be expressed as a DED rule using the formal expression: $\forall city, pop1, pop2 : (city, populationTotal, pop1) \wedge (city, populationTotal, pop2) \rightarrow (pop1 = pop2)$. Similar DED rules can be written for the other properties. Each of those rules was violated several times, as several duplicate conflicting records for these properties appeared in the various Wikipedias. When such a rule is violated, there are only two resolution options, namely deleting population count $pop1$ or $pop2$. The evaluation process used the five preferences described below, based on the quality metrics discussed in Section 4.

PREFER_PT is based on the *source reputation metric*, that imposes a preference order on the information found in different input datasets. In our case, we used the following trust order for the Wikipedias: Portuguese, English, Spanish, German, French. In case of conflicts, the most trustworthy information prevails. To implement this preference, we can define a feature that gives a "trust rank" to each triple based on its source and, in case of conflict, uses the preference to select the triple that has the highest rank.

PREFER_RECENT is based on *freshness* and provides a bias towards recent information: in case of conflict, the most recent information is kept. It can be modeled using a feature that assigns to each triple a value indicating how long ago it was last edited.

[1] All datasets can be found at: www.oeg-upm.net/files/mpoveda/ISWC2012Main

[2] geoftp.ibge.gov.br

[3] geo.linkeddata.es/brasil/sparql

PLAUSIBLE_PT is based on the *plausibility metric* that considers the actual data and is used to determine whether a piece of information is “irrational”. In particular, if the population is less than 500, the area less than 300 km^2 and the founding date earlier than year 1500, the triple is considered “irrational” and will always be dropped in case of conflict. If both conflicting triples are “rational”, then we resort to the PREFER_PT preference to choose one of the two.

WEIGHTED_RECENT: this preference is based on the “weighted freshness” metric. In case of conflict, it evaluates the last update date of the two conflicting triples: if they are found to be close (less than 3 months apart) they are considered equally up-to-date, so the preference PREFER_PT is used to choose the one to keep; otherwise, the most up-to-date information prevails (i.e., PREFER_RECENT is used).

CONDITIONAL_PT: this preference is based on the “conditional source trustworthiness” metric. Intuitively, it states that for small cities (less than 500.000 citizens) the Portuguese Wikipedia is more reliable so the PREFER_PT preference is used; for larger cities, we use PREFER_EN, which is similar to PREFER_PT except that the trustworthiness order of the English and Portuguese Wikipedias is swapped.

For the repair process, we used the algorithm of [28] with the above validity rules and preferences. The repair process initially searches in the dataset for rule violations, and resolves each violation independently using one of the above preferences. Each repair result was compared against the GS and its quality was evaluated under various dimensions (conciseness, consistency, validity, completeness and accuracy).

6.2 Experimental Results

Our analysis considered five different quality dimensions, namely conciseness, consistency, validity, completeness and accuracy [23]. These dimensions were evaluated against the final result of the repair under the different preferences.

Conciseness requires that no duplicates appear in the result, and is guaranteed to be perfect as our algorithm trims duplicate triples as part of the repair process. *Consistency* is defined as the lack of conflicting triples, whereas *validity* is defined as the lack of rule violations. These two notions coincide for the particular rules considered. Again, our algorithm guarantees perfect consistency and validity, as it eliminates all conflicts. *Completeness* is the coverage of information about the domain of interest that the fused dataset exhibits. For the particular input, it has been measured to be 77,02%. Completeness is not affected by our process, because our algorithm only deletes conflicting triples, and does not add or change triples.

The *accuracy* dimension is the most interesting of all; accuracy is defined as “being as close as possible to the actual state of affairs”, where the “actual state of affairs” in our case is taken by the GS. Accuracy is evaluated by comparing the output resulting from each preference against the GS. However, it is important here to distinguish between “inherent” inaccuracy and inaccuracy caused by the repair process.

For example, the population of Aracati (per the GS) is 69159, whereas in the fused dataset it has two population counts: 69159 and 69616. The repair process will drop one of the two values to resolve the conflict; if it chooses to keep the correct value (69159) then the accuracy of the dataset is improved, otherwise the repair process contributes

Table 1. Evaluation Results

populationTotal

Preference	Good	Bad	Good+Bad	Optimal	Sub-optimal	Approximate	Total
PREFER_PT	224	10	234	654	56	710	944
PREFER_RECENT	208	26	234	580	130	710	944
RATIONALITY_PREFER_PT	224	10	234	663	47	710	944
WEIGHTED_PREFER_RECENT	221	13	234	641	69	710	944
WEIGHTED_PREFER_RECENT	220	14	234	642	68	710	944

areaTotal (raw)

Preference	Good	Bad	Good+Bad	Optimal	Sub-optimal	Approximate	Total
PREFER_PT	0	11	11	14	419	433	444
PREFER_RECENT	2	9	11	87	346	433	444
RATIONALITY_PREFER_PT	0	11	11	9	424	433	444
WEIGHTED_PREFER_RECENT	1	10	11	25	408	433	444
WEIGHTED_PREFER_RECENT	0	11	11	32	401	433	444

areaTotal (modified)

Preference	Good	Bad	Good+Bad	Optimal	Sub-optimal	Approximate	Total
PREFER_PT	5	2	7	156	91	247	254
PREFER_RECENT	5	2	7	151	96	247	254
RATIONALITY_PREFER_PT	5	2	7	155	92	247	254
WEIGHTED_PREFER_RECENT	4	3	7	152	95	247	254
WEIGHTED_PREFER_RECENT	5	2	7	150	97	247	254

foundingDate

Preference	Good	Bad	Good+Bad	Optimal	Sub-optimal	Approximate	Total
PREFER_PT	6	3	9	0	3	3	12
PREFER_RECENT	6	3	9	0	3	3	12
RATIONALITY_PREFER_PT	6	3	9	0	3	3	12
WEIGHTED_PREFER_RECENT	6	3	9	0	3	3	12
WEIGHTED_PREFER_RECENT	6	3	9	0	3	3	12

to the inaccuracy of the data. The first case (i.e., when the algorithm keeps the correct data) is called a *good choice*, whereas the second is called a *bad choice*.

On the other hand, the city Oiapoque has two conflicting population counts, namely 20226 and 20426 in the fused dataset; its actual population, per the GS, is neither 20226 nor 20426, but 20509. Thus, the inaccuracy is inherent in the data, and cannot be affected by the repair process, regardless of the repairing choice made. In this case, the algorithm's choice only affects accuracy in terms of "closeness" to the actual value: 20426 is closer to the reality, and is therefore better in terms of accuracy. If the algorithm chooses 20426, we say that we have an *optimal approximation choice*, otherwise we have a *sub-optimal approximation choice*.

Finally, note that non-conflicting records that are inaccurate compared to the GS are ignored because our algorithm does not deal with non-conflicting records. In the following, we present the number of good, bad, optimal and sub-optimal choices for each of the preferences considered (Table 1) and also compare the accuracy of the dataset be-

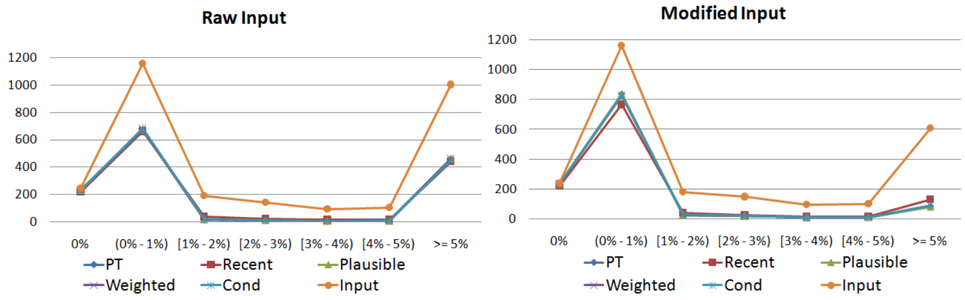

Fig. 1. Comparing the Accuracy of the Input and the Repairs

fore and after the repair (Figure 1). For clarity, our analysis is split in the three important properties of our dataset (*populationTotal*, *areaTotal*, *foundingDate*).

Our experiments showed that 944 cities contained a duplicate population entry. In 234 of these cases (24,8%), the correct value could be found in the input, and in most of these cases the algorithm managed to select it correctly (208-224 times, depending on the preference – cf. Table 1). The simple PREFER_PT, as well as the PLAUSIBLE_PT preferences give the best results, indicating that the Portuguese Wikipedia is indeed the most reliable when it comes to Brazilian cities' population. Surprisingly, PREFER_RECENT performs poorly, despite the fact that population is a dynamic property where up-to-date information is usually more reliable. In the remaining 710 cases where the actual value could not be found in the input, PLAUSIBLE_PT had the best performance again with 663 optimal choices, followed by PREFER_PT with 654. The worst behavior was exhibited by PREFER_RECENT, with only 580 optimal choices.

In the case of *areaTotal*, we note that the results were poor for all preferences. Upon further investigation, we realized that the problem was partially caused by the format of the area values. In particular, the area measurements in the Portuguese Wikipedia were generally off the actual one by 3 orders of magnitude. This was most probably related to the different use of the thousands separator (",") and the decimal places separator (".") in the English and Portuguese Wikipedia. Unfortunately, this was not consistent throughout the values, as some values in the Portuguese Wikipedia used the English notation, whereas others used the Portuguese one.

To evaluate this hypothesis, we pre-processed all area values coming from the Portuguese Wikipedia by multiplying them with 1000, and re-ran the experiment. The results of the modified input look much better as the number of total violations dropped significantly (from 444 to 254), indicating that many of the original violations were caused by said extraction problem. The results for the various preferences are relatively good, the best performance being exhibited again by the PREFER_PT and RATIONALIZED_PREFER_PT preferences (with minor differences from the rest).

The results related to the *foundingDate* property are the least interesting of the three, because, as can be seen in Table 1, all the preferences exhibited the same behavior.

In addition, we evaluated the accuracy of the dataset before and after the repairing process, by determining how much the value of each property (*populationTotal*, *areaTotal* and *foundingDate*) differs from the corresponding value in the GS (before and after the repair), as a percentage of the value in the GS. The results are shown in Figure 1. The orange line represents the total accuracy of the input (for all 3 properties), whereas the other five lines represent the accuracy of the repair for each preference. The x axis represents the accuracy (0% indicates accurate values); the y axis represents the number of triples that have the corresponding accuracy. Obviously, the values differ depending on whether the raw or the modified *areaTotal* input was considered. The figure shows that the repair process improves accuracy, and that all 5 preferences give similar accuracy.

7 Related Work

Several works on quality assessment have appeared in the literature (see [1] for a survey). In [2] users are allowed to express quality assessment policies to filter information from the Web. In [16] customizable assessment processes are formalized through a Web Quality Assessment model. In [10] SPARQL queries were used to identify various quality problems, whereas in [22] an XML-based model is proposed and used by a web service cooperation broker to select the best data from different services.

In the context of the Semantic Web, repairing addresses malformed values and datatypes or accessibility and derefencability issues [12], entity matching and disambiguation [13], and resolving inconsistencies, incoherencies or invalidities (*ontology debugging*) [29]. The latter type is the most relevant to our work.

Most works on ontology debugging consider some Description Logic as the underlying language and address the problem of removing logical contradictions. In our setting, we consider custom validity rules over RDF data. Moreover, most works focus on the problem of identifying inconsistencies, rather than resolving them [9]. In the cases where the active resolution of inconsistencies is supported, the resolution is usually done manually or semi-automatically [15], possibly with the help of an interactive tool (e.g., ORE [19], PROMPT [27], or Chimaera [21]). To the best of our knowledge, the only automated repairing approaches in the area appear in [24] and [28] regarding data fusion and ontology repair respectively.

The work described in [24] considers repairing in the context of data fusion and is very similar to ours. The authors propose a configuration-based approach where repairing is based on a user-defined conflict resolution strategy (similar to a preference) that uses quality metrics and fusion functions. However, such strategies can only consider metadata information, and cannot be combined to form complicated strategies. Moreover, the approach of [24] is restricted to handling conflicting information, i.e., it cannot support arbitrary validity rules (e.g., DEDs). Thus, the present paper can be seen as taking the best of both [24] and [28]: on the one hand, it allows more complicated validity rules in the spirit of [28]; on the other it lifts the limitations of both [24] and [28] on preferences, by allowing complicated preferences over both the data and metadata.

Most works record provenance by either associating triples with a named graph [4,33] (as done here) or by extending an RDF triple to a quadruple where the fourth element represents the triple's provenance [6,20]. These works vary in the semantics of the fourth element, which can be used to represent provenance, context, access control, trust, or other metadata information. The work of [8] addressed the problem of recording provenance in the presence of RDFS inference, where the provenance of an implicit triple is defined as the "combination" of the provenance of the implying ones.

8 Conclusions

Quality is an important issue in modern datasets, especially in the context of LOD where several dynamic and potentially unreliable sources are being interlinked, with no central control or curation. Quality assessment allows the evaluation of the quality of datasets, whereas quality repair aims at improving quality, with emphasis on the validity dimension. Apart from its value as a stand-alone process, repairing is also an integral process in various contexts, such as data integration/fusion [24] and evolution [17].

This paper deals with quality assessment and repair of LOD datasets. We proposed a number of quality metrics and applied them as preferences in the repairing process defined in [28]. Our approach is similar in spirit to [24], but extends [23,28,24] by providing more sophisticated provenance-based assessment metrics and by combining data and metadata information in the definition of complicated preferences that are used as guidelines for repair.

Our focus was not on developing a general approach to the problem (which can be found at [23,24,26,28]), but on evaluating the usefulness of the proposed metrics and preferences. As the extensive literature on data quality has shown, the usefulness of such metrics is application- and context-dependent, so we focused our evaluation on a specific data fusion setting. This is, to our knowledge, the first work evaluating an automated repair algorithm in a LOD fusion setting using provenance metadata.

In the future, we plan to consider and evaluate more diverse and/or complicated quality assessment metrics (and their associated preferences) for both the quality assessment and the quality repair problems in different settings. Moreover, we plan to consider integrating the above automated method with some kind of interactive preference elicitation interface as an aid for users to formulate complicated preferences.

Acknowledgments

This work was partially supported by the projects PlanetData and BabelData.

References

1. C. Bizer. *Quality-Driven Information Filtering in the Context of Web-Based Information Systems*. PhD thesis, Freie Universitat Berlin, 2007.
2. C. Bizer and R. Cyganiak. Quality-driven information filtering using the wiqa policy framework. *Web Semantics*, 7(1):1–10, January 2009.
3. A. Calì, G. Gottlob, and A. Pieris. Advanced processing for ontological queries. *Proceedings of VLDB Endowment*, 3:554–565, 2010.
4. J. Carroll, C. Bizer, P. Hayes, and P. Stickler. Named graphs, Provenance and Trust. In *WWW*, 2005.
5. A. Deutsch. Fol modeling of integrity constraints (dependencies). In *Encyclopedia of Database Systems*. 2009.
6. E. Dumbill. Tracking Provenance of RDF Data. Technical report, ISO/IEC, 2003.
7. W. Fan. Dependencies revisited for improving data quality. In *PODS-08*, 2008.
8. G. Flouris, I. Fundulaki, P. Pediaditis, Y. Theoharis, and V. Christophides. Coloring rdf triples to capture provenance. In *ISWC-09*, 2009.

9. G. Flouris, D. Manakanatas, H. Kondylakis, D. Plexousakis, and G. Antoniou. Ontology change: Classification and survey. *Knowledge Engineering Review (KER)*, 23(2), 2008.
10. C. Fürber and M. Hepp. Using semantic web resources for data quality management. In *EKAW-10*, 2010.
11. P. Georgiadis, I. Kapantaidakis, V. Christophides, E. Mamadou Nguer, and N. Spyratos. Efficient rewriting algorithms for preference queries. In *ICDE-08*, 2008.
12. A. Hogan, A. Harth, A. Passant, S. Decker, and A. Polleres. Weaving the pedantic web. In *LDOW-10*, 2010.
13. A. Hogan, A. Zimmermann, J. Umbrich, A. Polleres, and S. Decker. Scalable and distributed methods for entity matching, consolidation and disambiguation over linked data corpora. *Journal of Web Semantics*, 10, 2012.
14. J. Juran. *The Quality Control Handbook*. McGraw Hill, New York, 3rd edition, 1974.
15. A. Kalyanpur, B. Parsia, E. Sirin, and B. Cuenca Grau. Repairing unsatisfiable concepts in owl ontologies. In *ESWC-06*, 2006.
16. T. Knap and I. Mlýnková. Web quality assessment model: trust in qa social networks. In *8th international conference on Ubiquitous intelligence and computing*, 2011.
17. G. Konstantinidis, G. Flouris, G. Antoniou, and V. Christophides. A formal approach for rdf/s ontology evolution. In *ECAI-08*, 2008.
18. G. Lausen, M. Meier, and M. Schmidt. Sparqling constraints for rdf. In *EDBT-08*, 2008.
19. J. Lehmann and L. Buhmann. Ore - a tool for repairing and enriching knowledge bases. In *ISWC*, 2010.
20. R. MacGregor and I.-Y. Ko. Representing Contextualized Data using Semantic Web Tools. In *Practical and Scalable Semantic Systems*, 2003. In conjunction with ISWC.
21. D.L. McGuinness, R. Fikes, J. Rice, and S. Wilder. An environment for merging and testing large ontologies. In *KR*, 2000.
22. M. Mecella, M. Scannapieco, A. Virgillito, R. Baldoni, T. Catarci, and C. Batini. Managing data quality in cooperative information systems. In *Meaningful Internet Systems*, 2002.
23. P. Mendes, C. Bizer, Z. Miklos, J.-P. Calbimonte, A. Moraru, and G. Flouris. D2.1: Conceptual model and best practices for high-quality metadata publishing. PlanetData Del., 2012.
24. P. Mendes, H. Muhleisen, and C. Bizer. Sieve: Linked data quality assessment and fusion. In *LWDM-12*, 2012.
25. B. Motik, I. Horrocks, and U. Sattler. Bridging the gap between owl and relational databases. In *WWW-07*, 2007.
26. F. Naumann. *Quality-driven query answering for integrated information systems*. Springer-Verlag, Berlin, Heidelberg, 2002.
27. N.F. Noy and M.A. Musen. Prompt: Algorithm and tool for automated ontology merging and alignment. In *AAAI*, 2000.
28. Y. Roussakis, G. Flouris, and V. Christophides. Declarative repairing policies for curated KBs. In *Proceedings of the 10th Hellenic Data Management Symposium (HDMS-11)*, 2011.
29. S. Schlobach and R. Cornet. Non-standard reasoning services for the debugging of description logic terminologies. In *IJCAI-03*, 2003.
30. G. Serfiotis, I. Koffina, V. Christophides, and V. Tannen. Containment and minimization of rdf(s) query patterns. In *ISWC-05*, 2005.
31. E. Tacchini, A. Schultz, and C. Bizer. Experiments with wikipedia cross-language data fusion. In *SFSW-09*, 2009.
32. W.C. Tan. Provenance in databases: Past, current, and future. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, 2007.
33. E. Watkins and D. Nicole. Named Graphs as a Mechanism for Reasoning About Provenance. In *Frontiers of WWW Research and Development - APWeb*, 2006.