Leveraging Linked Open Data: A Link Maintenance Framework

André Gomes Regino
andre.regino@students.ic.unicamp.br
Institute of Computing - Unicamp
Campinas, São Paulo

Julio Cesar dos Reis
jreis@ic.unicamp.br
Institute of Computing - Unicamp
Campinas, São Paulo

Abstract

Connections among RDF (Resource Description Framework) data elements represent the core of LOD (Linked Open Data). These connections are built with semi-automatic linking algorithms using a variety of similarity methods. Interconnected data demand automatic methods to maintain their consistency. Constant update of RDF connections is relevant for the evolution of RDF datasets. However, changing operations can influence well-formed links, which turns difficult the consistency of the connections over time. This study investigated new methods responsible for fixing and updating links among structured data following ontologies rules and properties. We contribute with the design and development of an automatic method that updates RDF links based on changing operations in RDF datasets. The framework that implements our method - named LODMF - was evaluated in terms of discovering broken links in big and well-known Linked Open datasets.

Keywords: link maintenance; linked open data.

1 Introduction

Links between data elements described via the RDF model are at the heart of the Semantic Web. The growing number of structured data published in RDF repositories confirms the potential of the global data space proposed by the Semantic Web view. Usually, links are built using (semi)-automatic instance matching algorithms based on pre-defined heuristics and similarity calculation methods.

Currently, there is a huge mass of interconnected RDF data that requires methods and tools to deal with their consistency. Although implementing change operations in RDF datasets is essential to guarantee the evolution of structured data, such operations can influence established links, which provides difficulties in maintaining the consistency of data connections over time.

This paper describes new methods and algorithms capable of correcting RDF links between RDF data (organized according to ontologies) based on the evolution of dataset properties. This involves designing techniques and formalizing RDF change operations from one version to another concerning links between instances of ontology concepts.

Our investigation contributes to the following aspects: (1) a survey discussing an extensive study of strategies to debug, discover, and fix links inside and outside LOD datasets (Section 2); (2) a tool to analyze correlations between changes in RDF triples and changes in links (Section 3); (3) a technique that automatically discovers semantically broken links based on the evolution of RDF datasets (Section 4.2); (4) a (semi)-automatic method that, supported by change operations on RDF datasets, indicates maintenance actions on links, turning inconsistent links healthy again (Section 4.3).

2 Link Maintenance

State-of-the-art studies have studied the broken link problem using the following techniques: dataset versioning, notification of changes to dataset maintainers, and backlinks storage [3]. However, to the best of our knowledge, the literature lacks investigations dealing with the maintenance of links established among RDF datasets. The results of our literature review [3] indicate that there are solutions for detecting broken links - some of them with scalability issues, given the size of the dataset. However, none of them can fix broken links of any kind, structurally or semantically, in the context of Linked Data, without the assistance of humans throughout the process. Although the literature has addressed techniques to track changes in RDF repositories to identify deltas between versions [5], the task of maintaining the links up-to-date deserves further investigation.

3 Experimental Analysis of Correlation Between Changes in Triples and Links

This investigation aimed to study the effects of changes in LOD data instances and their impacts on previously established links. In this way, we defined a series of analyzes to verify how changes in a dataset influence the links at the instance level. More precisely, this experiment aimed to study how links are affected when a change occurs, affecting instances from one version of a dataset to another.

Figure 1 presents the problem of the study. Given two knowledge bases - R^S and R^T - at a time j, there is a link connecting two resources r_a and r_b. Assuming that resource r_a has been removed from knowledge base R^S in time $j+1$.

In: IV Concurso de Teses e Dissertações (CTD 2022), Curitiba, Brasil. Anais Estendidos do Simpósio Brasileiro de Sistemas Multimídia e Web (WebMedia). Porto Alegre: Sociedade Brasileira de Computação, 2022.

ISSN 2596-1683

This link, if left unchanged, may have become inconsistent. We want to find out with this study what types of changes most affect the links in these knowledge bases.

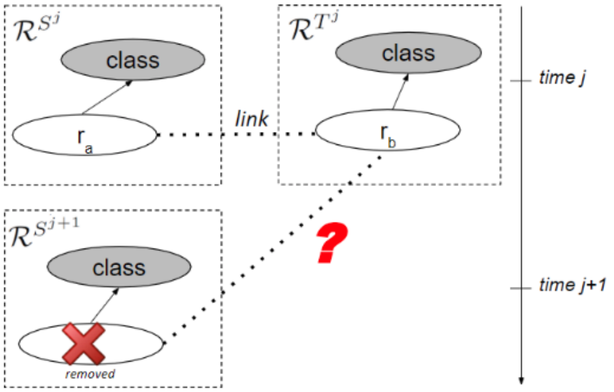

Figure 1. Link maintenance Problem

For this analysis, two versions of the knowledge base Agrovoc[1] were used, a base of the widely known LOD network on life sciences, covering resources, triples, and links about agriculture, food, and the environment. The first version, from April 2018, contains 4,254,655 triples, and the second version, from April 2019, contains 4,540,205 triples.

The following cases were analyzed:

a) Removal of an instance with the removal of a link.

b) Removing an instance without removing a link.

c) Adding a new instance without creating a link.

d) Addition of a new instance and the creation of a link.

e) Modification of a predicate/object of the triple resulting in the addition of a link related to the subject of the triple.

f) Modification of a predicate/object of the triple resulting in the removal of a link related to the subject of the triple.

g) Modification of a predicate/object of the triple by modifying the link related to the subject.

h) Modification of a predicate/object of the triple without modifying the link related to the subject.

The results from the analysis indicate that complex changes (Modification of triples) impact links more than superficial changes, such as adding and removing triples, in the context of the Agrovoc knowledge base [4]. Based on this result, we started focusing on modified links. Based on this result, we build the LODMF framework to cover modification cases (changes in resources and links) found in the datasets.

4 Linked Open Data Maintenance Framework

We developed a framework for executing a maintenance strategy after the evolution of a LOD dataset. This process is key to keeping the links unbroken and updated, considering new versions of these datasets. With the LODMF framework, the maintainers - people responsible for maintaining the LOD datasets - can verify the link integrity of their datasets.

The proposed framework to deal with the broken link problem was organized into three steps: (1) change detection, (2) recognition of inconsistent links, and (3) link repair. The objective is to obtain a version of the knowledge base with consistent links, given as input two different versions of this knowledge base. Each step, input, and output are in Figure 2 and the following sections.

4.1 Detection of Changes

In this step (A in Figure 2), LODMF detects changes that occurred in a given period of time based on two different versions of the LOD datasets. These changes can be simple (add or remove) or complex (update). Step A requires as input the LOD knowledge base in two versions: before evolution and after evolution. Step A computes all changes made between versions. The framework generates as output from Step A a list containing the modified links, whether they are negatively affected by the change or not.

4.2 Recognize Affected Links

In this step (B in Figure 2), links are found between resources of two different datasets that, with the evolution of one of them, turned the link semantically inconsistent.

Figure 3 shows two links as an example of an evolution of the subject of a link. Specifically, triples are composed of a *DBpedia* subject, a "sameAs" predicate, and a Wikidata object. The first triple (upper part) represents the triple before evolution (in time *j*) with the subject "alcoholic beverage". The second line (bottom part) represents the triple after the evolution (in time $j + 1$) with the subject "alcoholic drink".

The methodology to recognize affected links is composed of 5 steps. As input, the dataset in two different versions is expected, and as output, the list of links is categorized by heuristics, including the category of broken links.

In the first step (reading the datasets), two different versions of the same dataset are required. These datasets contain multiple RDF triples and links connecting to an external dataset. In the second, links are separated from other triples. First, the method detects which predicates connect subjects and objects in different datasets. Subsequently, all triples that have predicates categorized as links are retrieved.

In the third step, our solution compares each link between the versions, retrieving the links with changed subjects but keeping their predicate and object. We note that there are consistent and inconsistent changes to the links among these changes. We perform similarity analyzes between the affected resources to determine whether these changes break links. In the fourth step, the solution applies a similarity calculation to measure the degree of relationship between the resources. The output of this step is a normalized value that ranges from 0 to 1, with values close to 1 representing

[1] http://aims.fao.org/agrovoc/releases

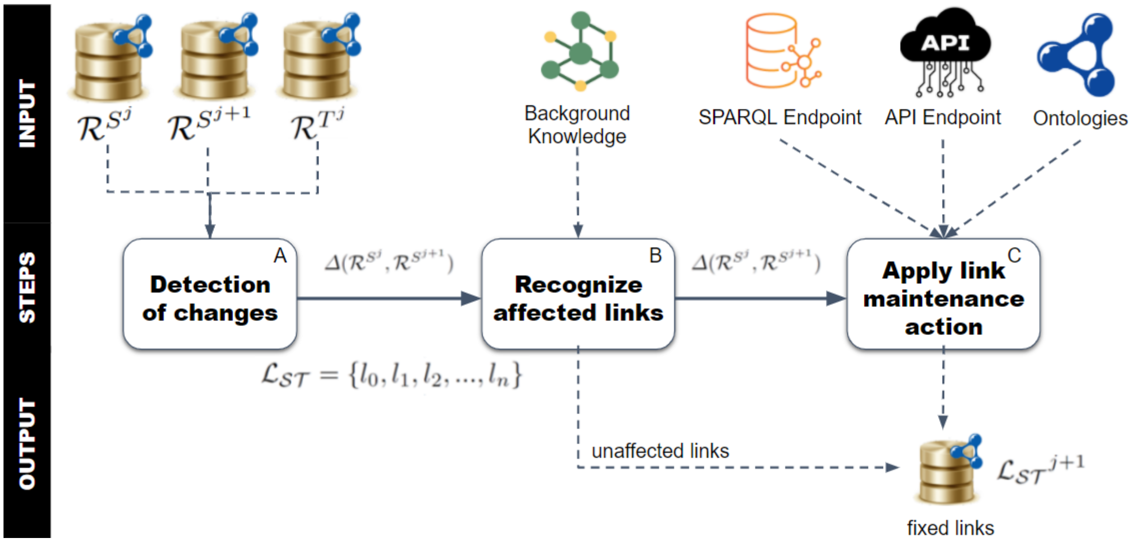

Figure 2. LODMF Framework

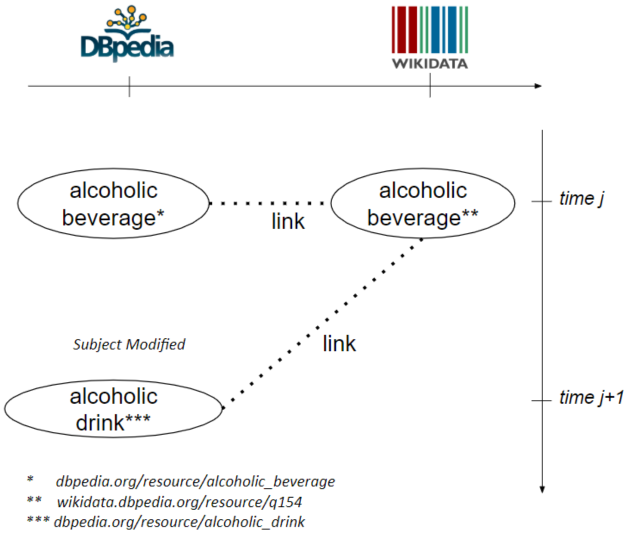
DBpedia
WIKIDATA
alcoholic beverage*
link
alcoholic beverage**
time j
Subject Modified
link
alcoholic drink***
time j+1
* dbpedia.org/resource/alcoholic_beverage
** wikidata.dbpedia.org/resource/q154
*** dbpedia.org/resource/alcoholic_drink

Figure 3. Changed link connecting DBpedia and Wikidata

more similar results. In the fifth step, we define heuristics taking into account the similarity values and the calculations between the resources in the fourth step.

Our objective is to classify each affected link in one of the cases expressed by the following heuristics: Heuristic 1 (unknown): This case predicts the impossibility of calculating similarity for unknown reasons. This includes scenarios where labels associated with the resource are unavailable and scenarios where the similarity algorithm cannot compute the similarity aspect; Heuristic 2 (unaffected): The change did not semantically affect the link. The resulting resource after the change may have become a synonym for the old resource; Heuristic 3 (affected): Possible case of a broken link. We understand that a link becomes semantically broken if the similarity between the involved resources decreases compared to the original link in time j; Heuristic 4 (positive refinement): Link quality has improved. We detect this case based on the complement assumption of heuristic 3: the semantic similarity of the link before evolution is less than the similarity value calculated after evolution.

We applied the methodology to evaluate our solution from the point of view of searching for semantically affected links in real-world RDF databases. To carry out this evaluation, we used DBpedia as the source dataset of the links and the Wikidata and GeoNames datasets as the destination. Two versions of DBpedia were used, one dated October 2015 and another from October 2016. One hundred seventy-four links were extracted between DBpedia and Wikidata and 3975 between DBpedia and GeoNames. Among the results obtained, around 30% of the cases of heuristic 1, approximately 45% of the cases of heuristic 2, 14% of heuristic 3, and 11% of heuristic 4 were found. These results indicate that, although several cases were found where the link quality improved in terms of semantics (heuristic 4) and that also many cases, the change did not affect the semantics of the link (heuristic 2), many cases of possible links affected negatively were discovered (heuristic 3). This shows that there is still room for improvement in updating resources and links [1].

4.3 Apply Link Maintenance Action

This step (C in Figure 2) applies corrective actions to the links [2]. These actions can be reconnecting to other resources or removing the link. The links not affected in Step B are added to the final list of links with the fixed links. The last step of the framework takes the list of modifications from Step B as input and applies maintenance actions suited for each link. The output from Step C is a list of suggested fixes.

Initially, a list of candidates to replace the broken resource is generated. One of the resources on this list may replace the broken link resource under verification by the dataset maintainer, making this link healthy again.

Our solution performs a series of comparisons between each candidate in the list and the object in the triple. These comparisons generate similarity values using semantic similarity algorithms. Semantic similarity calculates whether two terms share the same meaning, even though they are lexically different. The five comparisons performed are (1) Labels: Each label of the candidate list is compared with the label of the link object; (2) Instance of/Type: It compares which classes instantiate the resources of the candidate list with the class that instantiates the object's class (Figure 4); (3) sameAs string: Using the transitive property of the *owl:sameAs* property, the resources of the candidate list are compared with the resources that the object is externally connected to; (4) Scopes: Using properties of wider scope *skos:broader* and narrower scope *skos:narrower*, we compare the resources coming from these properties in each of the candidates with resources that are also broader and narrower in the scope of the object; and (5) User Input: Compares property values entered by the user between the candidate list and the object.

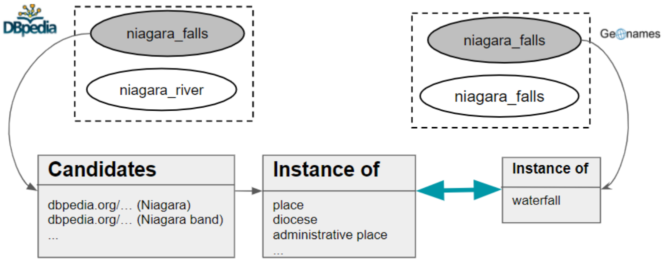

Figure 4. Instance of/Type Comparison between resources of DBpedia and Geonames

Each comparison generated a similarity value between each candidate in the list with the triple object, generating a matrix of candidate similarity values for each comparison.

The average of each candidate is generated so that it is possible to rank the top K resources most similar to the object. This step represents the beginning of the action suggestion process displayed to the dataset maintainer user. For this choice, the average reached by each candidate is compared with a threshold pre-established by the user.

Then, our solution starts to define the best action(s) the user can take. Among the possible actions, we can mention (1) Subject reconnection: the action of replacing the subject of the existing link with another resource from the list; (2) Predicate reconnection: the action of replacing the existing predicate (*owl:sameAs*) with another one that better fits the existing subject and object in the link. This action is valid only when no resource in the candidate list has reached the threshold value; (3) Complete removal: the action of deleting the link in question. (4) No action: Users can choose not to take any action on the link, as they can understand that the link is not broken or that the repair is ineffective.

More than one of these actions is displayed to the user, but only one of them can be chosen and carried out by the LODMF framework in each link. After the list of actions returns, the user chooses the desired action. The system performs this action, ending the link correction cycle.

5 Conclusion

Recognition and update of semantically broken RDF links is a challenging task. In this investigation, we proposed the LODMF framework for applying corrective actions necessary to make the RDF bases consistent. We contributed with a state-of-art link maintenance framework and software tools that handle broken links throughout the process of their maintenance. Our framework was experimentally evaluated using real-world and well-known datasets from the LOD cloud, such as DBpedia, Geonames, Wikidata, and Agrovoc. The obtained results reveal that our solution for maintaining established links is suited to help maintainers of RDF datasets keep links updated according to RDF change operations affecting resources.

6 Acknowledgments

This work was financially supported by the São Paulo Research Foundation (FAPESP) (grants #2017/02325-5 and #2018/14199-7)[2].

References

[1] André Gomes Regino and Júlio Cesar dos Reis. 2020. Discovering Semantically Broken Links in LOD Datasets.. In *Workshop Managing the Evolution and Preservation of the Data Web (MEPDaW'20) co-located at the 19th International Semantic Web Conference (ISWC'20)*. 17–26.

[2] André Gomes Regino, Julio Cesar dos Reis, and Rodrigo Bonacin. 2020. Lodmf: A linked open data maintenance framework. In *2020 IEEE 29th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*. IEEE, 263–268.

[3] Andre Gomes Regino, Julio Cesar dos Reis, Rodrigo Bonacin, Ahsan Morshed, and Timos Sellis. 2021. Link maintenance for integrity in linked open data evolution: Literature survey and open challenges. *Semantic Web Journal* (2021).

[4] André Gomes Regino, Julio Kiyoshi Rodrigues Matsoui, Júlio César dos Reis, Rodrigo Bonacin, Ahsan Morshed, and Timos Sellis. 2019. Understanding Link Changes in LOD via the Evolution of Life Science Datasets. In *Proceedings of the Workshop on Semantic Web Solutions for Large-Scale Biomedical Data Analytics co-located with 18th International Semantic Web Conference (ISWC 2019), Auckland, New Zealand, October 27th, 2019 (CEUR Workshop Proceedings, Vol. 2477)*. 40–54.

[5] Yannis Roussakis, Ioannis Chrysakis, Kostas Stefanidis, Giorgos Flouris, and Yannis Stavrakas. 2015. *14th International Semantic Web Conference (ISWC)*. Springer, Chapter A Flexible Framework for Understanding the Dynamics of Evolving RDF Datasets, 495–512.

[2] The opinions expressed in here are not necessarily shared by the financial support agency.