Link Maintenance in the Semantic Web

Andre Gomes Regino[1], **Julio Cesar dos Reis**[1]

[1]Institute of Computing – University of Campinas (Unicamp)
Campinas – SP – Brazil

{andre.regino,jreis}@ic.unicamp.br

Abstract. *Links among data elements represent the core of the Semantic Web. The links are built with semi-automatic linking algorithms using a variety of similarity calculus. The data interconnected by these algorithms demands automatic methods and tools to maintain its consistency. These changes occur mainly in datasets that represent knowledge in areas that evolve drastically, like biology and medicine. Even though the constant change of these links is an important task for the evolution of these structured datasets, such changing operations can negatively influence the well-established links, which turns difficult the consistency of the connections over time. In this work, we aim to investigate new methods responsible for fixing and updating links among ontologies in the Linked Open Data context.*

1. Introduction

Resources described in Semantic Web format using the Resource Description Framework (RDF) have been extensively implemented. In recent years, a large number of knowledge bases interconnected on the Web have appeared describing different types of resources in a structured way. The explicit connection between resources belonging to different bases is a key aspect of the interconnection between repositories. The links allow previously isolated bases to be explored in a combined manner, avoiding that the bases - also known as datasets - become isolated islands of knowledge. However, data described in RDF is subject to change. Property values, predicates, and objects are removed and added among several other change operations. These changes may turn existing links invalid, as previously elements that are relevant to the definition of the link have been modified. Manual maintenance of connections can be difficult to perform due to the huge number of links available, as long as the size of the datasets.

A similar problem has been studied in the literature in order to maintain mappings between updated ontology concepts. However, the literature does not yet address the context of links between instances of concepts. The first difficulty with the problem is to establish clear criteria for when connections become invalid. For example, there may be some types of changes that do not affect the accuracy of the links established, while some other types invalidate. The second challenge is to make it possible that from the different types of possible changes, the connections are adapted accordingly.

Although alert techniques for possible corrupted links are found in the literature, there is still no knowledge of how to implement automatic actions that update them.

This project aims to handle the problem of broken link using link maintenance algorithms. To achieve it, we propose the development of a framework, LODMF - Linked Open Data Maintenance Framework.

The remaining of this paper is organised as follows: Section 2 presents the general and specific goals of the ongoing research; Section 3 shows the related work about link maintenance. Section 4 presents the methodology and the framework, while 5 presents what has been done and achieved so far. Finally, Section 6 explains the next steps and challenges.

2. Goal

2.1. Generic Goal

We aim to investigate, formalize and implement semi automatic link maintenance actions in order to recognize affected links and turn them up-to-date. These actions will address cases of structurally broken links, which part of the link was removed, and semantically broken links, which the meaning of the resources changed.

Figure 1 shows an evolution of a removal of a given triple and the absence of a link removal associated to that triple. A well-established link connecting r_a from dataset $\mathcal{R}^S$ to r_b from dataset $\mathcal{R}^T$ at a time j might have gotten broken after the removal of r_a at time $j + 1$.

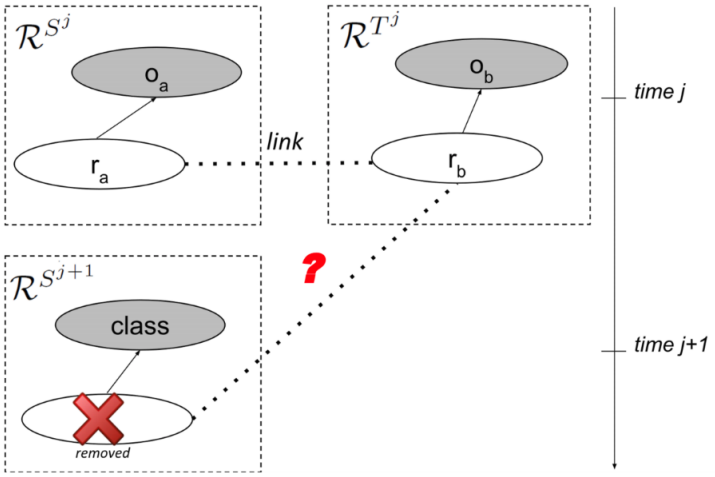

Figura 1. Problem Characterization [Regino et al. 2019]

2.2. Specific Goals

Our research will provide the following contributions:

- An exhaustive study of the behavior regarding broken links and its correlation with simple and complex changes in Linked Open Data datasets;
- A framework that outputs the broken links found between two versions of the same dataset, in addition to the fixing, rebuild or removal of these links;

Each of the steps mentioned at Section 4 retrieves relevant knowledge of the maintenance of links timeline: In Step 1, we can retrieve a list of changed concepts between the versions of the datasets; In Step 2, a list of links that may be broken; In Step 3, an up-to-date dataset.

3. Related Work

Previous investigations addressed the problem of broken links using different methods. Literature has shown that most of the proposals only handle the detection phase of the process and are not concerned with the fixing phase. It has also shown that there is no evidence of a solution that can automatically detect and fix these links without human intervention.

Dealing with broken links in the traditional Web is not a novelty problem. Existing work attempted to adapt traditional Web techniques of dealing with broken links to the Semantic Web environment [Vesse et al. 2010]. Existing implemented tools send notifications to the maintainer of the dataset when detecting that a resource has changed. This approach suffers from scalability issues since it is dependant on the number of notifications sent [Popitsch and Haslhofer 2011].

Another approach stores the versions based on changes in the dataset and, in the case of inconsistency detection, it is easier to manually restore previous versions and find the source of the potentially broken link. One of the limitations in this approach is the number and size of deltas, which are the mapping of differences between two versions of a given dataset [Kondylakis et al. 2017]. The use of metadata was studied to store relevant data with the nodes of the datasets, which is a case of a change, revisiting the metadata of the nodes helps on identifying what happened [Meehan et al. 2016].

4. Methodology

In order to keep to links up-to-date, we are building a framework composed of three main steps, listed as:

- **Step A:** Detect the changes that occurred in a given period of time based on two releases of the involved datasets that follow the LOD principles. These changes can be simple changes (atomic changes like addition or removal actions) or complex changes (non-atomic changes like update action, a change composed by a removal followed by an addition actions) of the knowledge stored in the datasets;
- **Step B:** Recognize which of the changes found at Step A turned into an affected link (semantically or structurally broken link). This link could be also created, removed, updated or remained untouched (which can also be an example of a inconsistency, given that the dataset had evolved, but the link remains the same);
- **Step C:** Apply corrective actions in the recognized affected and broken links. This action can be a reconnection with an unbroken link (like the children or parents of the outdated link) or the removal of the link.

Figure 2 shows the steps mentioned above and subsections 4.1, 4.2 and 4.3 explains the purpose of each of them.

4.1. Step A: Detection of Changes

The initial step (Step A of Figure 2) is responsible for detecting the changes that occurred in a given period of time based on two releases of the involved LOD datasets. These changes can be simple changes (like simple addition or removal of triples) or complex changes (update action) of the knowledge stored in the datasets.

As an example, suppose there is a link connecting two resources in two different datasets. The first dataset has a resource entitled “Dubai”. The second dataset also has a

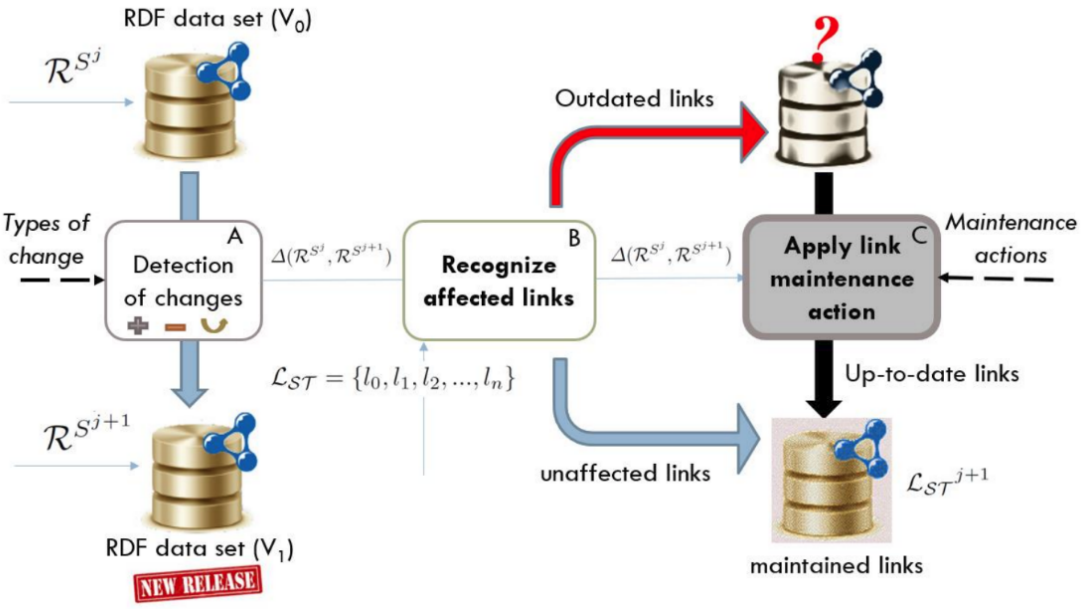

Figura 2. LODMF Framework [Regino et al. 2020]

resource with the same title. They are connected by a "sameAs" link, stating that "Dubai" from the first dataset is the same "Dubai" from the second dataset. However, the maintainer of the first dataset changed the resource "Dubai" to "Emirates of Dubai". Step A of LODMF aims to detect these changes that may affect the links.

This step derives some research questions: How to detect changes between two LOD datasets? Which kind of changes most occurs in LOD datasets? Is there any relation between the domain of the dataset and the most frequent changes?

4.2. Step B: Recognize Affected Links

Step B is responsible for categorizing the set of links as inconsistent or consistent, depending on the change it suffered. Following the example presented at subsection 4.1, in Step B the changed link ("Dubai" connected to "Dubai" changed to "Emirates of Dubai" connected to "Dubai") is evaluated. If the change did not alter the meaning intended by the link's author, then the change is categorized as valid and the link as unaffected. However, this is not the case of the modification from "Dubai" to "Emirates of Dubai", which changed the meaning of the link. "Emirates of Dubai" from first dataset can not be linked to "Dubai" resource of the second dataset by the predicate "sameAs", given that these resources do not represent the same thing in the real world. This semantic inconsistency is detected and the set of affected links proceed to Step C.

The research questions related to these steps are: How to precisely categorize a link as inconsistent? Is there a correlation between the number of affected and unaffected links? The proportion of broken links over the total links is the same between the many versions of the dataset through time?

4.3. Step C: Apply Link Maintenance Action

At Step C we apply maintenance actions on the affected links detected at Step B (subsection 4.2). In our example, given that the link between "Emirates of Dubai" and "Dubai"

is inconsistent, it should be repaired at Step C. The repairment can be a reconnection to another resource (connect "Emirates of Dubai" from the first dataset to a synonym in the second dataset), a replacement of the predicate connecting the resources ("sameAs" can be changed to "differentFrom") or, in the last case, the complete removal of the link.

The derived questions in these steps are as follows: As long as an invalid link is found, which actions to take? How can we assure that the newly created link fits in more than the older one? How to semantically reconnect a link?

5. Initial Results

Tables 1, 2 and 3 shows the collected results in a conducted study [Regino et al. 2019] regarding Step A of Section 4. At this work, we aimed to correlate changes that occurred in triples and resultant changes in links associated with that triples. We used Agrovoc, a well-known dataset in life sciences related to agriculture, food, and environment. We collect two releases of this dataset: April 2018's, with 4.254.655 triples and April 2019's with 4.540.205 triples. For each changed link, we try to correlate these changes (addition, removal, and modification) with changes in triples based on the versions of the dataset.

Triples / Links	Add	No Add
Add	**98.84%**	0.41%

Tabela 1. Added Cases

Triples / Links	Remove	No Remove
Remove	3.85%	**96.15%**

Tabela 2. Removed Cases

Triples / Links	Add	Remove	Modify	No Change
Modify	0%	0.04%	4.41%	**95.55%**

Tabela 3. Modified Cases

Table 1 shows that Agrovoc dataset applies the concept of Linked Data, linking 99% of their newly added triples to an external dataset. At Table 2, however, 96.15% of identified removed cases show that if an internal triple is removed, the connecting link remained untouched, generating cases of structurally broken links. Table 3, regarding modification, shows that the fourth sub-case concerns the most frequent one, in which the modification of triples led to unchanged links. This case needs additional studies to further observe to which extend these unchanged links remained semantically inconsistent due to the modifications of the associated RDF triples.

We also performed a literature survey to understand to which extent the link maintenance problem for integrity in Linked Open Data was studied. We discovered that most of the developed techniques focus on the discovery part of the broken links, not in the fixing part. We also discovered that, to the best of our knowledge, there is no evidence of a tool that can discover and fix semantically broken links automatically. This study is not yet published.

6. Future Work

We are now focusing on developing novel strategies to address the challenges of identifying broken links and maintaining them (Steps B and C of Section 4). In addition, we are evaluating state-of-art methods to detect semantic inconsistencies in the links, such as the usage of genetic programming [Isele and Bizer 2011] and background knowledge (WordNet [Fellbaum 2012] and BabelNet [Navigli and Ponzetto 2012]). For evaluation purposes, we are developing a gold standard dataset sample containing links that are semantically inconsistent, links that changed but are still consistent, and links that are unchanged. We will measure each of the 3 steps of the framework in terms of precision, recall, and f-measure using this gold standard dataset to verify the quality of the links modified by our approach.

Referências

Fellbaum, C. (2012). Wordnet. *The encyclopedia of applied linguistics*.

Isele, R. and Bizer, C. (2011). Learning linkage rules using genetic programming. In *Proceedings of the 6th International Conference on Ontology Matching-Volume 814*, pages 13–24. CEUR-WS. org.

Kondylakis, H., Despoina, M., Glykokokalos, G., Kalykakis, E., Karapiperakis, M., Lasithiotakis, M.-A., Makridis, J., Moraitis, P., Panteri, A., Plevraki, M., et al. (2017). Evordf: A framework for exploring ontology evolution. In *European Semantic Web Conference*, pages 104–108. Springer.

Meehan, A., Kontokostas, D., Freudenberg, M., Brennan, R., and O Sullivan, D. (2016). Validating interlinks between linked data datasets with the summr methodology. In *International Conferences On the Move to Meaningful Internet Systems*, pages 654–672. Springer.

Navigli, R. and Ponzetto, S. P. (2012). Babelnet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250.

Popitsch, N. and Haslhofer, B. (2011). Dsnotify: A solution for event detection and link maintenance in dynamic datasets. *Web Semantics: Science, Services and Agents on the World Wide Web*, 9(3):266–283.

Regino, A. G., dos Reis, J. C., and Bonacin, R. (2020). Lodmf: A linked open data maintenance framework. In *Proceedings of the Workshop on Semantic Technologies for Smart Information Sharing and Web Collaboration (Web2Touch) co-located with 29th IEEE International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE) (accepted for publication)*.

Regino, A. G., Matsoui, J. K. R., dos Reis, J. C., Bonacin, R., Morshed, A., and Sellis, T. (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)*, volume 2477 of *CEUR Workshop Proceedings*, pages 40–54.

Vesse, R., Hall, W., and Carr, L. (2010). Preserving linked data on the semantic web by the application of link integrity techniques from hypermedia. In *Linked Data on the Web (LDOW2010)*. Event Dates: 27th April 2010.