UpLOD: A Tool for Inconsistent Links Repairment in the LOD

André Gomes Regino[1], Enio de Jesus Pontes Monteiro[1], Andressa Cristina dos Santos[1], Julio Cesar dos Reis[1,2]

[1] Institute of Computing, University of Campinas, Brazil
andre.regino@students.ic.unicamp.br,
enio.monteiro@students.ic.unicamp.br,
andressa.santos@students.ic.unicamp.br, jreis@ic.unicamp.br
[2] Nucleus of Informatics Applied to Education, University of Campinas, Brazil

Abstract. The amount of interconnected RDF data has expressively grown as a result of the adoption of Semantic Web technologies. Changes in RDF datasets are essential to guarantee data evolution. However, changes may affect well-formed and validated "sameAs" links, impacting the real meaning intended by ontology maintainers and/or the link creators. The manual maintenance is unfeasible due to the data volume. In this article, we describe a software tool capable of correcting links between RDF datasets based on the evolution of the underlying datasets. Our tool automatically detects broken links caused by RDF changes. On this basis, it provides maintenance actions assisted by users for repairing links and turning them adequate. We present an architecture integrating these features and how the user interacts with the software.

Keywords: Link Maintenance; Linked Data; Semantic Web tools

1 Introduction

The semantic definition of RDF entities tends to undergo modifications over time. These changes can influence "sameAs" links to other datasets and potentially decrease the quality and consistency of the links. Maintaining their accuracy is critical because applications for data search and integration are based on them for their proper functioning. Due to the large volume of existing links leveraged by the growing number of RDF repositories, manual correction becomes arduous and costly. In this sense, correcting links manually is a hard and error-prone task. This justifies the development of novel computational tools capable of assisting domain specialists in this maintenance task.

Literature has presented studies to address the problem of link maintenance arising [2] due to the constant reviews and changes carried out in RDF datasets. Our literature analysis indicates that existing approaches are not fully prepared to perform automatic detection of changes in RDF repositories [2], and subsequently, carry out a correction in artifacts associated with them, such as links and semantic annotations [2].

In this article, we present a software tool capable of identifying and applying maintenance actions on links affected by the evolution of RDF datasets as automatically as

possible. Our maintenance process comprises the execution of three steps: 1) identify changes between RDF dataset versions; 2) discover broken links among these changes; and 3) repair these links (*cf.* Section 2). Our software solution integrates these features allowing dataset maintainers to visualize and analyse changes and conduct repairment actions via the tool.

2 Link Maintenance Software Tool

The development of our tool was based on the conceptualization of the *Linked Open Data Maintenance Framework* (LODMF) [1] to support the maintenance of links between RDF datasets. Our tool receives two different versions of the RDF dataset and detects changes in the triples from one version to the other. On this basis, the tool suggests maintenance actions to correct found cases of broken links. The output is a set of fixed links. Fig. 1 presents the interaction flow to perform maintenance actions.

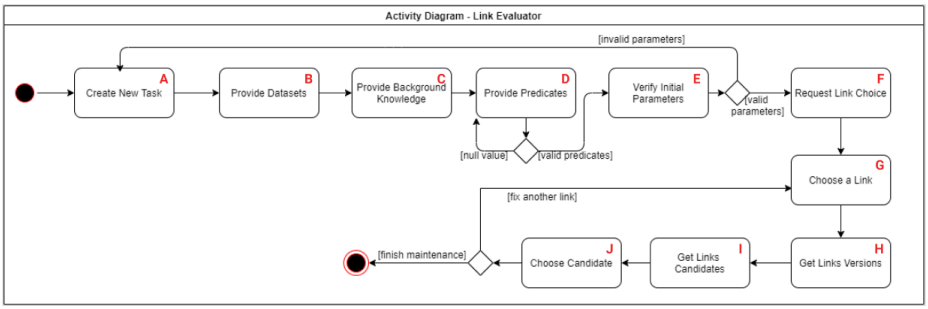

Fig. 1. User interaction flow throughout the system.

The users can "Create a New Task" (*cf.* A in Fig. 2 – Figure A) to start a new dataset maintenance; or the user can "Open a Task" (*cf.* B in Fig. 2 – Figure A) to keep a RDF verification already started. The left side menu in the first screen allows to access the recent tasks performed (*cf.* D in Fig. 2 – Figure A).

When the users select the option of "Create a New Task" (A in Fig 1 – Figure A), they encounter an interface describing the required data for input (*cf.* Figure B of Fig. 2). The user informs a name for the task and includes a list of parameters (B, C and D in Fig. 1). It starts with the source RDF dataset (*cf.* B in Fig. 2 – Figure B) and its second version. The user indicates a target RDF dataset (*cf.* C and D in Fig. 2 – Figure B). The source RDF dataset is the location of the outgoing links and the target is the location of the incoming links. The user can insert a "Background Knowledge" as a semantic network used by our tool to compute semantic similarity between RDF resources (*cf.* E in Fig. 2 – Figure B). This is required for computing candidate resources in the maintenance process. The user can inform the types of predicates of the links that should be analysed (*cf.* F in Fig. 2 – Figure B). As default, we consider "owl:sameAs".

Fig. 2 presents a dashboard with key information regarding the input datasets. It enables the user to view the detailed percentage of the identified links and other relevant statistics about the datasets. The user can access the list of changed links separated by modified subject, predicate and object through the left side menu (*cf.* A, B and C in Fig. 2). A graphic (*cf.* D in Fig. 2) enables to analyze the percentage of links that were already verified by the user, the number of discarded and to be verified. The system presents information about the involved datasets (E in Fig. 2) as long as the changing operations are found in links between the processed RDF versions (F in Fig. 2).

The analysis is made link by link to complete the maintenance process. At this stage, we consider only those affected links by the computed RDF changes. This is a result of a procedure from our system that categorizes an evolved link as affected. In summary, a link is considered affected if the semantic similarity between subject and object decreased from version v0 to version v1 [3]. The user might select one of the affected links (steps F and G in Fig. 1) displayed in the left side menu (*cf.* A, B and C in Fig. 2). This classifies the links based on the type of changes that affected them. By selecting one link, our software tool presents details regarding such link in an interface dedicated to its maintenance (*cf.* Fig. 3).

Fig. 3 presents the interface in which the left side contains a fixed menu with the list of changed links. It was designed with different panels illustrating useful information about the selected link. The panel (E in Fig. 3) shows an example of the evolution of a link between versions V0 and V1 of the dataset. In this example, the first link (with subject represented by the ID 2643743) evolved to a new version (with subject represented by the ID 11609024). After this evolution, our tool categorized the link as affected.

The panel (D in Fig. 3) shows the suggested actions that the user can take to fix this affected link (*cf.* step I in Figure 1). These actions are automatically computed by the system. In the running example, the first suggestion is the replacement of the subject from the resource with ID *11609024* to *2643744*. In this case, a subject consisting of a higher degree of similarity with the object ("*London*").

Alternatively, if the user does not agree with the tool's suggestion, there is a possibility to completely remove the affected link from the dataset (option "Discard link" – D in Fig. 3) or apply other maintenance actions suggested by the tool (step J in Fig. 1). The list of suggested actions varies from choosing another subject to replacing the predicate, for instance. To this end, the user should choose the option "Show more link suggestions" (*cf.* D in Fig. 3).

Following the repairment process, two graphs are shown to help users in understanding the actions suggested by the tool (*cf.* H in Fig. 3). They support people comprehending visually what will be the result after applying one or other link maintenance action. The panel (*cf.* G in Fig. 3) shows the values of similarity of each similarity algorithm used by our tool. The example (*cf.* G in Fig. 3) illustrates three algorithms/background knowledges: *Levenshtein*, *WordNet* and *Nasari*.

The selected links (*cf.* F in Fig. 3) (green and orange background) are compared using the three similarity algorithms (*cf.* G in Fig. 3). Values closer to 1 indicate that the subject and object of the links are more syntactically and semantically similar when compared to values closer to 0. The graph view shows the affected links and some of

its connected resources (*cf.* H in Fig. 3). The orange graph (left side) shows the link in a broken state; the green graph (right side) presents the link after repairment, if the suggestion made by the tool is accepted by the user. Both panels G and H serve as guidance to the user in the final decision. The user chooses the adequate action and commits the changes by clicking on "Apply change" (*cf.* D in Fig. 3). If the users need to analyze another link, they select the desired link on the left side panel, returning to step G (cf. Figure 1) of the system.

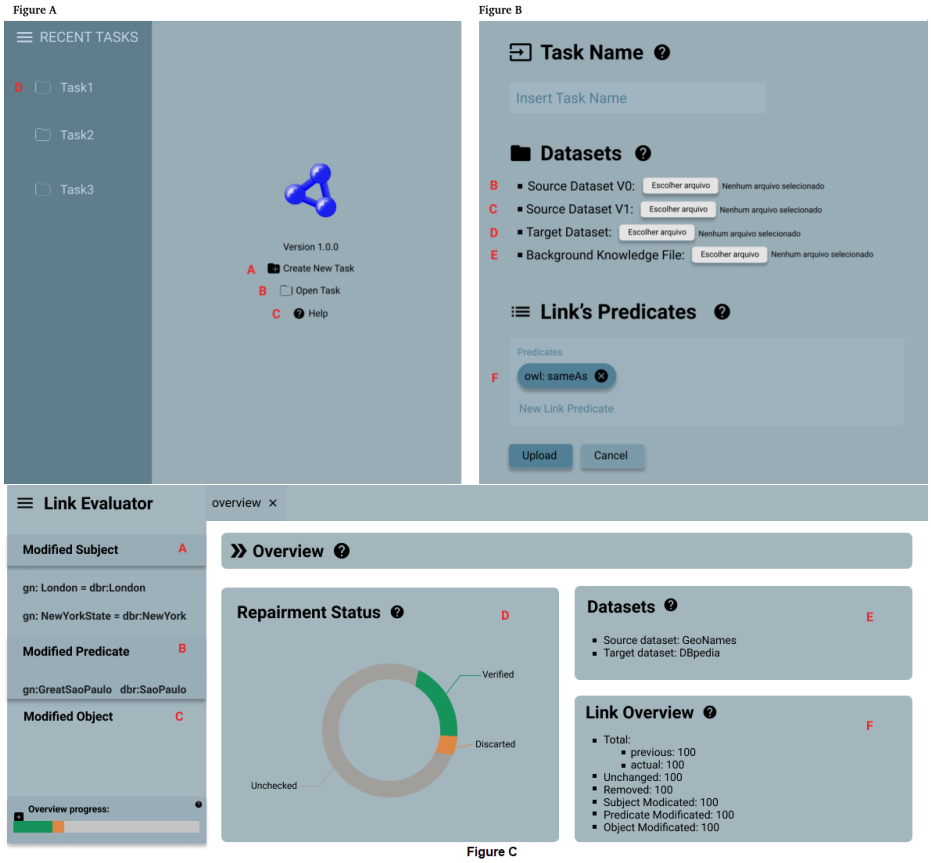

Fig. 2. Figure A - System home screen. Figure B - Initial and configuration interface of the Up-LOD. Figure C - Link evaluator dashboard interface.

3 Conclusion

The real value of semantic-enabled computer systems lays on the reliability of links. This study investigated how to keep links updated according to the evolution of RDF data repositories. We presented a software tool for the semi-automatic maintenance of RDF links affected by data evolution. Our defined maintenance process works on the

basis of change operations automatically identified in the evolution of RDF datasets. We are currently investigating additional features in the tool for the adaptation of RDF-based semantic annotations. Our next steps involve the development of a module responsible for maintaining semantic annotations. We plan to conduct complete case studies applying the use of the system in real-world scenarios.

Fig. 3. Link maintenance interface.

Acknowledgements

This work was financially supported by the São Paulo Research Foundation (FAPESP) (grants #2017/02325-5, #2018/14199-7, #2019/14582-8, #2020/12466-8)[3].

References

1. Regino, A.G., dos Reis, J.C., Bonacin, R.: Lodmf: A linked open data maintenance framework. In: 2020 IEEE 29th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE). pp. 263–268. IEEE (2020)
2. Regino, A.G., dos Reis, J.C., Bonacin, R., Morshed, A., Sellis, T.: Link maintenance for integrity in linked open data evolution: Literature survey and open challenges. Semantic Web **12**, 517–541 (2021)
3. Regino, A.G., dos Reis, J.C.: Discovering semantically broken links in lod datasets. In: Workshop Managing the Evolution and Preservation of the Data Web (MEPDaW) co-located at the 19th International Semantic Web Conference (ISWC), virtual conference. pp. 17–26 (2020)

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