Understanding Link Changes in LOD via the Evolution of Life Science Datasets

André Gomes Regino[1], Julio Kiyoshi Rodrigues Matsoui[1], Julio Cesar dos Reis[1], Rodrigo Bonacin[2], Ahsan Morshed[3] and Timos Sellis[3]

[1] Institute of Computing, University of Campinas, Campinas - SP, Brazil
andregregino@gmail.com, julio.k.r.matsoui@gmail.com, jreis@ic.unicamp.br
[2] UNIFACCAMP and Center for Information Technology Renato Archer, Campinas - SP, Brazil
rodrigo.bonacin@cti.gov.br
[3] Swinburne University of Technology, Australia
amorshed@swin.edu.au, tsellis@swin.edu.au

Abstract. RDF data has been extensively deployed for the interlinking of health-related data in a structured way. The definition of link statements between distinct resources plays a key role to interconnect several life science repositories. However, RDF assertions are subject to change, which can affect existing links. In this article, we conduct extensive experiments to understand the evolution of links in the Linked Open Data (LOD). The objective is to empirically associate changes in the semantic definition of data resources with modifications observed in predefined links. We consider two versions of the *Agrovoc* RDF repository to calculate different types of change operations and associate them to link change actions. Obtained results indicate the existence of the cases investigated in this study. We demonstrate that RDF changes impact the evolution of established links.

Keywords: LOD; Web of Data evolution; Link evolution; Change Operations; Link changes; Link Repair, RDF life science datasets

1 Introduction

Linked data technologies have become increasingly relevant for semantic interoperability and knowledge discovery in the life sciences. Their data sources are more and more interconnected via links between Resource Description Framework (RDF) data descriptions. The interconnection of RDF statements via explicit links plays a central role to assure data linkage. Datasets focusing on life sciences, such as Bio2RDF, Mesh and *Agrovoc* are part of the LOD cloud. These datasets are subject to a constant evolution via the publication of new releases.

In order to couple with domain updates, RDF statements defining real-world resources are subject to change. RDF triples can be updated, added or removed to keep the repositories up-to-date. Although the implementation of change operations in RDF datasets is essential to assure structured data evolution, these

operations can negatively affect established links. This hampers data linkage consistency over time. Indeed, the connection between nodes can be broken by a number of reasons, which changes the state of the link to broken or invalid [8].

Literature has superficially studied the maintenance of linked datasets [8] even though mapping evolution phenomena between ontologies has further advanced in the last years [1]. Previous studies have investigated semi-automatic approaches to adapt ontology mappings when at least one of the mapped ontologies evolves [3]. Dos Reis *et al.* conceptualized the *DyKOSMap* framework [2] for supporting the adaptation of mappings. However, it lacks thorough investigations empirically grounded to unveil how links evolve in LOD.

In this article, our thorough and original experiments analyze the way change operations in RDF repositories correlate to changes observed in links. We systematically investigate the key factors of link evolution according to RDF change operations. We assume that the results of our analyses must allow understanding and predicting the consequences of changes performed in RDF datasets over links. We aim to understand the necessary elements regarding the changes detected in the bases to adapt links between instances of two datasets, *i.e.*, external links from one dataset to another. This study might support the definition of automatic mechanisms for link evolution in LOD.

Our experiments consider change operations in RDF repositories and the modifications observed in the level of links. This investigation performs several analyses to measure the extent to which the change operations are associated to the observable changes in links. For instance, how the removal of internal triples affect existing links to external datasets. We conduct experiments with versions of the *Agrovoc*[4] linked dataset.

Our findings demonstrate that changes in links (like addition, removal and modification) are caused as effects of modifications in predicate or object of internal triples. We found that in most cases that an internal triple is added, a link is created. However, when an internal triple is removed, only few cases of links associated to the triple is removed, leading to the problem of broken links.

The remaining of this article is organized as follows: Section 2 presents the related work; Section 3 reports on the description of the experimental study; Section 4 presents the obtained results, which are further discussed in Section 5. Finally, Section 6 draws conclusion remarks.

2 Background

The evolution of ontologies and mappings between them have been studied over the last years. Dos Reis *et al.* [1] proposed thorough studies to research the evolution of mappings between biomedical ontologies and defined techniques to adapt affected mappings [2].

There has been a number of studies on specific aspects of LOD evolution. Literature has addressed techniques to track changes in RDF repositories for the

[4] http://aims.fao.org/standards/agrovoc/linked-data

automatic identification of deltas between versions [10]. However, comprehending the evolution aspect of LOD deserves further investigations, specially the aspect of maintaining the links up-to-date.

Concerning the RDF link aspect, few studies approached the link integrity problem aiming to monitor and preserve data quality [9]. Pourzaferani and Nematbakhsh [9] defined a tool for fixing broken links considering the similarity between the associated entities.

Liu and Li [6] proposed techniques to keep link integrity based on the metadata sources as a way to detect data changes. Their proposal considered the use of metadata to detect and notify real-time changes in the dataset, without any need to scan the entire dataset periodically.

Studies have addressed the broken link problem in the context of LOD. Popitsch and Haslhofer proposed a event detection framework (*DSNotify*) [8] for dynamic datasets applied to linked data sources to inform actors about various types of events (create, remove, update). This allows actors to maintain links to resources in a distributed environment, fixing broken links in their local data, while preserving the links' integrity. In particular, this work focused on identifying events that can lead to broken mappings.

The framework MeLinDa [11] aimed to map and apply existent tools to join two ontological datasets, based on its URIs and ontologies. A total of 6 link building tools proposed in other studies were used to optimize the achieved results. For each one of them, the level of automation, domain specificity (some of them tends to work better in certain domains), types of similarities techniques were compared.

In contrast, the Diachron platform [7] managed the preservation of evolving linked data ecosystems. This can help publisher to decide how frequently their datasets need to be updated, by taking into consideration the domain they target or the type of link.

By using links with predicates as "same as" and "see also", Vesse *et al.* [12] designed an algorithm for retrieving linked data about the broken URI. This process was based on link maintenance of the traditional hypermedia. These contributions resulted in a doctoral thesis that developed a framework for handling broken links based on two solutions for structural broken links implemented in hypermedia.

Silk [13] refers to a framework responsible for keeping alive links between two datasets as both of them evolves. The Silk generates links between the datasets, evaluates them and track future links that have to be created.

Very few studies have attempted to empirically understand the dynamics of Linked Data. Kafer *et al.* [5] defined the *Dynamic Linked Data Observatory* (DyLDO) to monitor a fixed set of Linked Data documents. Their findings show the stability of RDF statements revealing how often their content changes. However, their work still does not perform corrective actions over links to keep them up-to-date. In addition, the behaviour of links based on the way associated triples changes were not investigated.

Although these studies present valuable attempts to somehow reach link maintenance, they fail in supporting the link evolution and remain unable to fully tackle the impact of the RDF changes in link evolution. In this research, we carry out empirical studies contributing to pave the way to the definition of automatic techniques of link evolution in LOD. Our long-term goal is to track the behaviour of links in the datasets in order to keep the links up-to-date as automatically as possible, minimizing the occurrence of broken links.

3 Design of the Study

This study aims to comprehend how changes related to RDF facts impact the evolution of links in LOD. We rely on the main building blocks of the Semantic Web paradigm, referring to the *Universal Resource Identifier* (URI), RDF and ontologies [4].

RDF dataset. A dataset in the context of Linked Data is an conglomeration of a finite number of RDF triples in a domain. Formally, $\mathcal{R} = \{t_1, t_2, t_3, ..., t_n\}$.

Triple. A RDF triple refers to a data entity composed of subject, predicate and object defined in the form of $t = (s, p, o)$ where:

- **Subject:** (s) is either an URI reference or a blank node.
- **Predicate:** (p) is a URI reference that defines a relation between a subject and a predicate.
- **Object:** (o) is either an URI reference, a literal, or a blank node.

A literal is a string combined with either a language identifier (plain literal) or a data-type (typed literal). Blank nodes are those nodes representing the resources for which a URI or literal are not given.

Ontology. An ontology $\mathcal{O}$ describes a domain in terms of concepts (general understanding of something), attributes and relationships [4]. Formally, an ontology $\mathcal{O} = (\mathcal{C}_\mathcal{O}, \mathcal{S}_\mathcal{O}, \mathcal{A}_\mathcal{O})$ consists in a set of concepts $\mathcal{C}_\mathcal{O}$ interrelated by directed relations $\mathcal{S}_\mathcal{O}$. Each concept $c \in \mathcal{C}_\mathcal{O}$ has an unique identifier and it is associated to a set of attributes $\mathcal{A}_\mathcal{O}(c) = \{a_1, a_2, ..., a_p\}$.

Link. Besides the use of triples inside a dataset, the linkage among several datasets interconnect the datasets. To this end, it is necessary that a predicate establishes a relation between a subject in the first dataset (source) and an object in the second (target). Formally, we define a link as $l =< r_a, p, r_b >$ connecting a pair of resources r_a and r_b, in which $r_a \in \mathcal{R}^S$ and $r_b \in \mathcal{R}^T$, such that $\mathcal{R}^S$ differs from $\mathcal{R}^T$. For the definition of p, we consider well-established properties to express the predicates of links including: $owl : SameAs$, $rdfs : seeAlso$, $owl : DifferentFrom$ and SKOS mapping properties vocabulary[5]. This study relies on the notion of the following types of links, consisting of predicates connecting the *Agrovoc* datasets with other datasets:

- $l_1 =< r_a, owl : sameAs, r_b >$

[5] https://www.w3.org/TR/skos-reference/#mapping

- $l_2 =< r_a, rdfs:seeAlso, r_b >$
- $l_3 =< r_a, owl:differentFrom, r_b >$
- $l_4 =< r_a, skos:exactMatch, r_b >$
- $l_5 =< r_a, skos:closeMatch, r_b >$
- $l_6 =< r_a, skos:broaderMatch, r_b >$
- $l_7 =< r_a, skos:narrowMatch, r_b >$

From now on, we use the notation $l(r_a \rightarrow r_b)$ to denote a link. We define a set of links between $\mathcal{R}^S$ and $\mathcal{R}^T$ as $\mathcal{L}_{\mathcal{ST}} = \{l_0, l_1, l_2, ..., l_n\}$. A complete dataset refers to the union of internal triples and links, such as $\mathcal{D} = \mathcal{R} \cup \mathcal{L}_{\mathcal{ST}}$.

Link change actions. Our study was structured to understand the link evolution in the LOD. To this end, we assume that there might exist different types of changes in links. Given two versions of the same RDF repository, l_k a link, we define the link change actions as follows:

- Unchanged link ($l_k \in {\mathcal{L}_{\mathcal{ST}}}^j \wedge l_k \in {\mathcal{L}_{\mathcal{ST}}}^{j+1}$)
- Added link ($l_k \notin {\mathcal{L}_{\mathcal{ST}}}^j \wedge l_k \in {\mathcal{L}_{\mathcal{ST}}}^{j+1}$)
- Removed link ($l_k \in {\mathcal{L}_{\mathcal{ST}}}^j \wedge l_k \notin {\mathcal{L}_{\mathcal{ST}}}^{j+1}$)
- Modified link ($l_k \in {\mathcal{L}_{\mathcal{ST}}}^j \wedge l_k \in {\mathcal{L}_{\mathcal{ST}}}^{j+1}$, $p^j \neq p^{j+1}$ or $r_b^j \neq r_b^{j+1}$)

RDF change operations. Considering two different versions of the same dataset, it is necessary to compute differences between RDF graphs. At this stage, we introduce the notion of time $j \in N$. For a RDF dataset, we denote $\mathcal{R}^{S^j}$ the initial version of the dataset and $\mathcal{R}^{S^{j+1}}$, the evolved version. Change operations are represented by an operator Δ, such that $\Delta(\mathcal{R}^{S^j}, \mathcal{R}^{S^{j+1}})$ (also known as a Diff operation) produces low-level changes as triple additions and removals in the RDF graph [10]. A set of simple changes is defined by $SCH = \{ch_1, ch_2, ..., ch_n\}$, such that $ch = (chp, t+, t-)$, as follows:

- chp refers to the change parameter defined by the type of change operation: addition or removal.
- $t+$ and $t-$ stand for the old and new triples of the updated RDF graph.

Complex changes refer to a set of well delimited simple changes; $CCH = \{SCH_1, SCH_2, ..., SCH_k\}$. A combination of simple change operations is necessary to obtain a complex change. For example, the replacement of a resource is the combination of the removal of a triple with the addition of a new one.

The evolution of RDF datasets in terms of changes affecting their triples may invalidate previously determined links. RDF statements (triples) remain the smallest manageable piece of knowledge. Modifications including the addition, removal, or update of a resource might affect triple statements, which can lead to a link integrity issue. In order to maintain the consistency of RDF datasets, its links should remain in a integrity state, even with underlying changes in data [8].

Consider a link l^j at time j and a link l^{j+1} based on distinct releases of the associated datasets. Modifications occurring in related resources of the link (r_a) from a release of the dataset in a specific time j to a time $j+1$ can invalidate such link $l^j(r_a \rightarrow r_b)$. For example, r_a can exist at time j ($r_a \in \mathcal{R}^{S^j}$), but be removed

at time $j+1$ ($r_a \notin \mathcal{R}^{S^{j+1}}$). In this sense, the link is considered structurally broken and l^j should be updated to a new state $l^{j+1}(r_a \rightarrow r_c)$, such that, in this case as an example, $r_a \neq r_c$.

In this study, for each change in Δ affecting triples of a RDF dataset associated to links, we investigate the way changes occured in links. To this end, we selected two distinct versions of the same RDF dataset. Our study explored the *Agrovoc*[6], a well-known dataset in life sciences related to agriculture, food and environment. *Agrovoc*'s maintainers regularly publish their open data on the web. Our criteria for such selection was considering a well-known dataset with characteristics of evolving over time. From one release version $\mathcal{R}^j$ to another $\mathcal{R}^{j+1}$, we calculated ontology change operations $\Delta(\mathcal{R}^{S^j}, \mathcal{R}^{S^{j+1}})$. We used this information to correlate with changes in the links $\mathcal{L}_{\mathcal{ST}}$.

Table 1 presents the releases of the datasets used in our investigation. $\mathcal{D}^j$ corresponds to the release before evolution, and $\mathcal{D}^{j+1}$ stands for a new release of the same dataset.

Table 1: Overview of dataset releases.

Notation	**Triples**	**Dataset release**
$\mathcal{D}^j$	4.254.655	Agrovoc release of April 2018
$\mathcal{D}^{j+1}$	4.540.205	Agrovoc release of April 2019

We describe some facts regarding the releases:

- The total number of triples is 4.254.655 in the initial dataset release $\mathcal{D}^j$ and 4.540.205 in the newer release $\mathcal{D}^{j+1}$;
- In $\mathcal{D}^j$, there are 64.977 links (also named here external triples) - divided into "*closeMatch*", "*exactMatch*", "*narrowMatch*", and "*broadMatch*" predicates - representing 1.52% of the whole dataset;
- The $\Delta(\mathcal{D}^{S^j}, \mathcal{D}^{S^{j+1}})$ (changes computed between the two dataset releases) consists of 535.467 changes, such that: 202 removals (0.03%), 6.990 additions (1.3%) and 528.275 modifications (98.65%);

We defined three types of analyses to investigate the evolution of the links. The first focuses on the relation of removed triples with links; the second emphasizes the behaviour of added triples; the third aims to map the relation between modified triples. For all these analyses, we assume that only one side of the dataset evolves per time. In this sense, we can assure that the observed effects in the links occur due to the observable modifications from one version to other of the RDF repository.

Analyses of removed triples over links. Figure 1 (a and b) presents the cases related to the removal of triples from one version to another.

- **Case R:1** (Figure 1 (a)). In this case, there is a link $l^j(r_a \rightarrow r_b)$ and the triple associated to the resource r_a is removed. In this sense, the resource

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

$r_a \in t_k$ such that $t_k \in \mathcal{R}^{S^j}$ and $t_k \notin \mathcal{R}^{S^{j+1}}$. This analysis searches for a link removal associated to the triple t_k. We consider that exists a coherence between the removal of the triple and the link.

– **Case R:2** (Figure 1 (b)). In this case, a triple t_k is removed from the RDF dataset $\mathcal{R}^{S^j}$ to $\mathcal{R}^{S^{j+1}}$, but the analysis detects an operation of unchanged link related to the triple t_k. We consider this case an inconsistency because a broken link is created due to the dataset evolution.

Analyses of added triples over links. Figure 1 (c and d) presents the two cases related to the addition of triples from one version to another.

– **Case A:1** (Figure 1 (d)). A new triple t_k is added in the $\mathcal{R}^{S^{j+1}}$; as a consequence, this analysis aims to identify new links associated to the new triple as a coherence in their evolution.
– **Case A:2** (Figure 1 (c)). This case presents the scenario in which a new triple is added, but no link is inserted from one version to another of the dataset associated to the new triple.

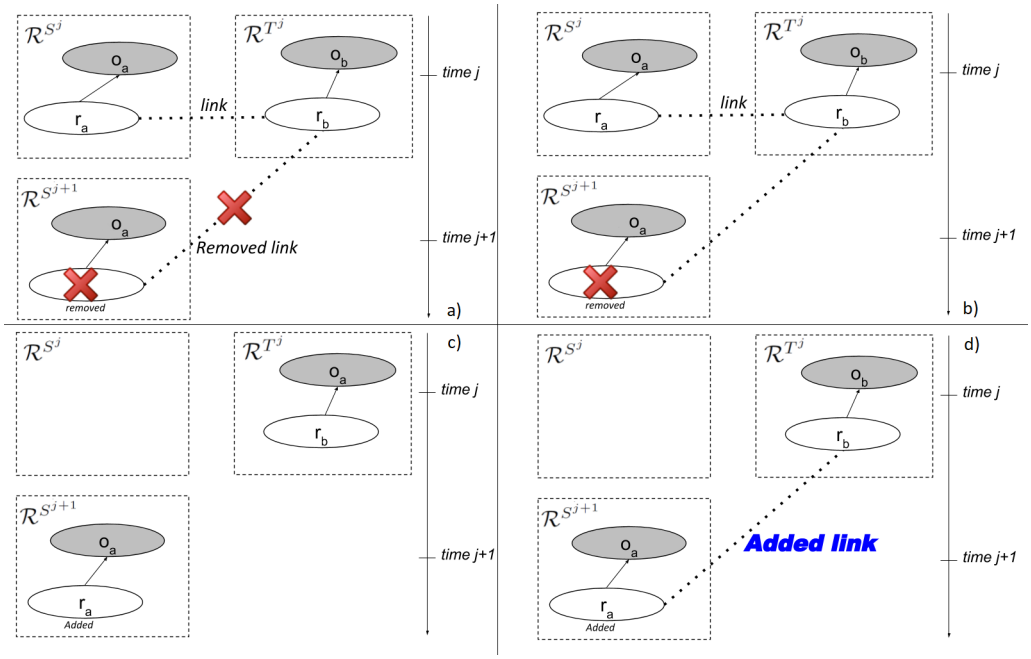

Fig. 1. Analyses of link evolution based on removed and added triples

Analyses of modified triples over links. Figure 2 presents the four cases related to the modification of elements in triples from one version to another.

- **Case M:1** (Figure 2 (e)). It presents the scenario in which based on a change in the predicate or object of a triple, the analysis recognizes an added link action in the dataset.
- **Case M:2** (Figure 2 (f)). In this case, a triple t_k has its predicate or object changed. This analysis aims to detect a removed link action associated to this change in the internal triple.
- **Case M:3** (Figure 2 (g)). This case investigates the behaviour of modified predicate or object of triples and the modification of a predicate or object in the link (triple and link share the same subject).
- **Case M:4** (Figure 2 (h)). It investigates the occurrences of modified predicate or object of triples that did not trigger modification in the link associated.

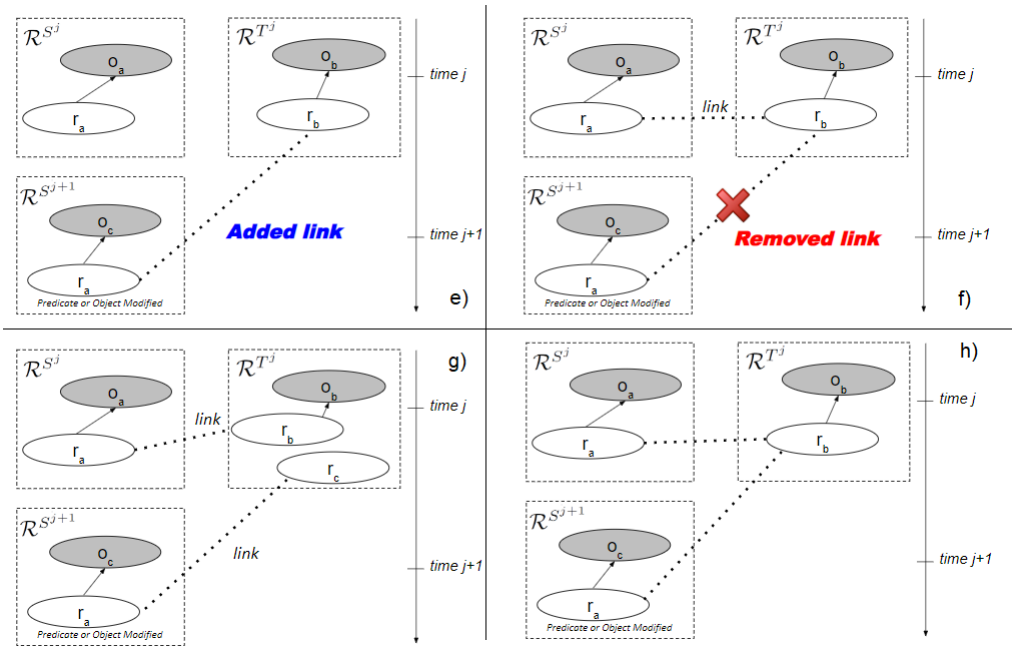

Fig. 2. Analyses of link evolution based on modified triples

4 Results

This section presents the results organized by the three types of analyses. Firstly, subsection 4.1 reports on the results of the analyses concerning the way removed triples affect links; subsection 4.2 reports on effects of triple addition on links, and subsection 4.3 reports on the effects of modified triples on links.

4.1 Results of the effects of removed RDF triples

Table 2 presents the encountered results in the analyses of association between removed triples with connected links. The columns entitled **Case R:1** and **Case R:2** represents the scenarios illustrated by case (a) and (b) in Figure 1.

Table 2: Results of the analyses of removed triples.

Id	Type	R:1	R:2
(a)	**Internal triples removed with link removal**	3	-
(b)	**Internal triples removed without link removal**	-	75
(c)	**External Triples (links) Removed**	124	124
(d)	**Internal Triples Removed**	78	78
(e)	**Total of Triples Removed (c+d)**	202	202
(f)	**Triples in $\mathcal{D}^{S^j}$**	4.254.655	4.254.655
(g)	**Triples in $\mathcal{D}^{S^{j+1}}$**	4.540.205	4.540.205
(h)	**Percentage (a OR b)/(c)**	2.41%	60.48%
(i)	**Percentage (a OR b)/(d)**	3.85%	96.15%
(j)	**Percentage (a OR b)/(e)**	1.49%	37.13%
(k)	**Percentage (a OR b)/(g)**	$< 0.01\%$	$< 0.01\%$

Case R:1. We detected 78 removed internal triples. The total triples present in $\mathcal{D}^{S^j}$, but removed in $\mathcal{D}^{S^{j+1}}$ is 202, in which 124 (61.38%) refer to links and 78 (38.61%) are internal triples. We found that 3.85% (i) of the removed triples (a) is also related to the removal of a link. This number found in (a) also represents 1.49% of the total triples removed (j) and less than 0.01% of the total of triples found in the dataset before evolution $\mathcal{D}^{S^j}$ (k).

Case R.2. The search of (b) returned 75 occurrences of the removal of internal triples, without the removal of associated links, representing 96.15% of internal triples removed (i), 37.13% of both internal and external triples removed (j), less than 0.01% of the total of triples in the dataset before evolution $\mathcal{D}^{S^j}$ (k). Figure 3 presents an example, in which the triple related to the resource 'Lavandula' (represented by the ID 'c_4228' in Agrovoc dataset) is removed, and the link connecting '*Agrovoc*' with an external dataset named 'Chinese Agricultural Thesaurus'[7] using the predicate 'broadMatch' is not removed.

[7] http://cat.aii.caas.cn

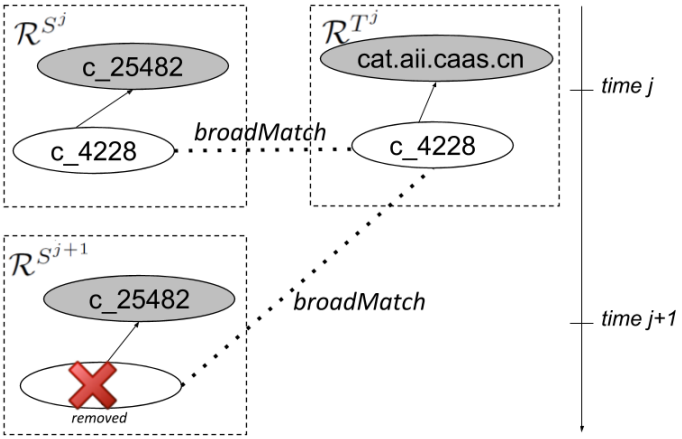

Fig. 3. Example of internal triple removal without link removal

4.2 Results of the effects of added RDF triples

We investigate the scenario in which added triples are related to new links associated to them. Table 3 presents the encountered results of our analysis. The columns entitled **Case A:1** and **Case A:2** represent the scenarios referred in cases (c) and (d) described in Figure 1, respectively.

Table 3: Results of the analyses of added triples.

Id	Type	A:1	A:2
(a)	**Internal triples added with links added**	6.909	-
(b)	**Internal triples added without links added**	-	29
(c)	**External Triples (links) Added**	494	494
(d)	**Internal Triples Added**	6.496	6.496
(e)	**Total of Added Triples (c+d)**	6.990	6.990
(f)	**Triples in $\mathcal{D}^{S^j}$**	4.254.655	4.254.655
(g)	**Triples in $\mathcal{D}^{S^{j+1}}$**	4.540.205	4.540.205
(h)	**Percentage (a OR b)/(d)**	106.35%	0.45%
(i)	**Percentage (a OR b)/(e)**	98.84%	0.41%
(j)	**Percentage (a OR b)/(g)**	0.15%	$< 0.01\%$

Case A:1. We found out 6909 internal triples added in $\mathcal{D}^{S^{j+1}}$ that has the same subject of other 494 subjects categorized as external triples (links). Figure 4 presents an example of this case, in which a new triple related to 'Lycopersicon' (represented by the ID 'c_4474' in Agrovoc dataset) is added, as long as a link to the external dataset named '*DBPedia*'[8] with the predicate 'closeMatch' is also created. We found 6990 (e) triple additions in $\mathcal{D}^{S^{j+1}}$, which 494 (7.07%) (c) are links and 6496 (92.93%) (d) are internal triples. We found that around 106.35% (h) of the internal triples added (a) produced an addition of a link. This number of occurrences that is higher than 100% means that an internal triple can be related to one or more links. This number found in (a) represents 98.84% of all triples added (i), and 0.15% of the total of triples in $\mathcal{D}^{S^{j+1}}$ (j) .

[8] https://dbpedia.org/

Case A:2. Our analysis found 29 occurrences of this case (b in Table 3), representing 0.45% of the internal added triples (h in Table 3), 0.41% of both triples added (i), and less than 0.01% of the total of triples in $\mathcal{D}^{S^{j+1}}$ (j).

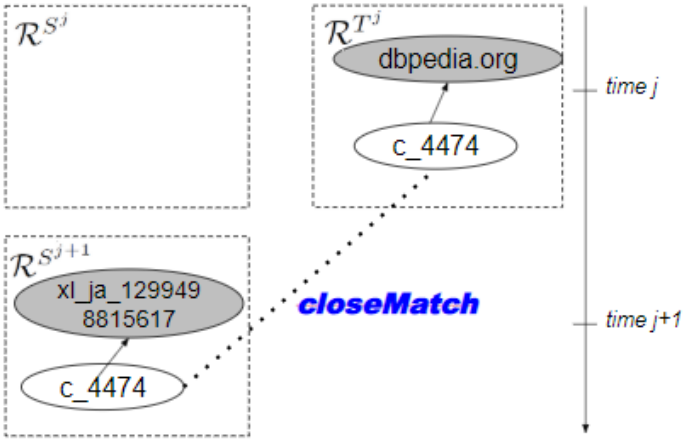

Fig. 4. Example of triple addition associated to link addition

4.3 Results of the effects of modified RDF triples

We show the results regarding the way modifications affecting an internal element of a triple influence addition, removal or modification of links. Table 4 presents the results for the cases M:1, M:2, M:3 and M:4 (*cf.* Figure 2).

Table 4: Results of the analyses of modified triples.

Id	Type	M:1	M:2	M:3	M:4
(a)	**Mod. Triples with Add. of links**	0	-	-	-
(b)	**Mod. Triples with Rem. of links**	-	245	-	-
(c)	**Mod. Triples with Mod. of links**	-	-	21.899	-
(d)	**Mod. Triples with no Mod. of links**	-	-	-	474.253
(e)	**Mod. internal triples**	496.397	496.397	496.397	496.397
(f)	**Added links**	494	-	-	-
(g)	**Removed links**	-	124	-	-
(h)	**Mod. links**	-	-	31.878	-
(i)	**Total of Modified Triples (e+h)**	528.275	528.275	528.275	528.275
(j)	**Percent. (a OR b OR c OR d)/(e)**	0.00%	0.04%	4.41%	95.55%
(k)	**Percent. (a OR b OR c OR d)/(i)**	0.00%	0.04%	4.14%	95.82%
(l)	**Percent. (a/f)—(b/g)—(c/h)**	0.00%	197.58%	68.69%	-

Case M:1. Results in Table 4 (a) show that 496.397 triples were modified. However, these modification did not produce association with the addition of links, sharing the same subject of the internal triple.

Case M:2. The second column of Table 4 shows that 245 internal triples that were modified caused the removal of links (b). This represents 0.04% (j) of the 496.397 modified internal triples (e), 197.58% (l) of the removed links (g), implying that links removed were associated to more than one internal triple, and 0.04% (k) of all the 528.275 modified triples (i).

Case M:3. As the third column of Table 4 shows, we found that 21.899 internal triples that were modified caused the modification of external triples (c). This represents 4.41% (j) of the 496.397 modified internal triples (d), 68.69% (l) of the modified links (g) and 4.14% (k) of all the 528.275 modified triples (i). Figure 5 shows an example of an internal triple with object that changed from 'Fund' to 'Credit', which caused an effect of changing the external dataset connected by 'exactMatch' from 'USA National Agricultural Library LOD' dataset to the 'Germany National Library LOD'[9] dataset.

Case M:4 The fourth column of Table 4 shows the total number of times that modifications in triples did not cause modification in links. Even if there is an evidence of change in predicate or object of the triple, no change was detected in the predicate or object of the external link. This is a case which deserves further investigation, because it can lead to links semantically inconsistent.

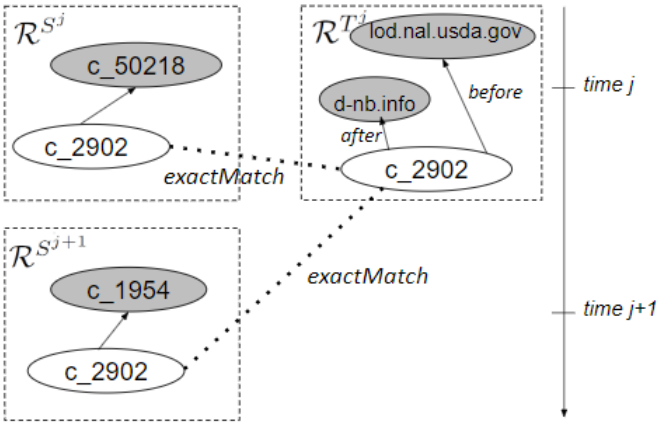

Fig. 5. Example of link modification after triple modification

[9] https://portal.dnb.de

5 Discussion

This investigation aimed to detect unexplored situations regarding the way RDF simple and complex changes in triples are associated with link change actions in LOD datasets. We found that it is possible to interrelate distinct change operations of RDF triples with the way links change. This is a key aspect to inform automatic techniques to update the link statements or to support the editors of controlled sources to update their models. Our original findings are thus useful to the design of link evolution mechanisms, which might be essential to enable the management of the LOD evolution and insure data quality over time. Our investigation highlighted the potentialities of our findings for managing the evolution of life science LOD.

We discovered that, for *Agrovoc*, simple changes as addition or removal of triples had a minor impact on the dataset, because these changes corresponded 1.33% of the computed changes. The analysis of the addition indicated that almost all the internal triples added had an external link added simultaneously with the same subject. The case representing the addition with internal triples with no link being added represented only 1% of the instances found. This implies that *Agrovoc* dataset clearly applies the concept of Linked Data, linking 99% of their newly added triples to an external dataset. The impact analysis of the exclusion change operation showed the opposite situation. Only 3.85% of the internal triples removed between the *Agrovoc* releases resulted in an exclusion of the associated links to an external dataset. The other 96.15% identified cases showed that if an internal triple is removed, the connected link remained untouched, generating a broken link.

Our results indicated that the biggest impact is caused by complex modifications of internal triples, which responded in 98.67% of all the computed changes. The first sub-case regarding modifications was related to the addition of a link when an internal triple is modified. We did not find an instance of this case, which demonstrates that in *Agrovoc* when there is a modification of a predicate or object of a triple, no link to a dataset is created.

The second sub-case was related to the removal, showing that there is a relation between modified triples leading to an exclusion of links. In particular, the modification of the predicate or the object of 245 triples, resulted in the removal of 124 links. Further investigations are required to comprehend if the removal of links happened with the influence of an internal triple with the same subject.

The third sub-case of the effects of modified RDF triples indicated that modification of triples can lead to modifications in links sharing the same subject. The fourth sub-case concerns the most frequent one, in which the modification of triples lead to link unchanged. This case needs additional studies to further observed to which extend these unchanged links remained semantically inconsistent due to the modifications of the associated RDF triples.

These results indicate the need to improve the LOD maintenance mechanisms. The *Agrovoc* LOD dataset[10] was released connected to several external vocabularies (*e.g.*, NALThesaurus, Chinese Agricultural Thesaurus and DBpedia). This allows the interchange of information across multiple systems, as well as human and machine interpretation of interconnected vocabulary data. However, the results obtained in this investigation suggest that such initiatives are affected by link maintenance issues. Broken and outdated links may introduce inconsistency in the use of LOD datasets whereas the manually maintenance is a hard and costly task. This indicates the need for investigations of new mechanisms to support the maintenance of links.

A removal or update of internal triples generate the need for inspecting links aiming to evaluate whether they remain consistent or not. Usually, this demands manual work, which are subject to resource constrains and fails (as our results indicate). The implementation of computational mechanisms that support this task can be valuable for LOD maintainers. The implementation of these mechanisms relies on the research of advanced techniques that are able to suggest changes and to identify inconsistencies in established links. Machine learning techniques, for instance, can be explored to suggest maintenance actions based on the structure of the linked datasets and previously constructed training sets.

We plan to improve the algorithms implementing our analyses for comparing more than two versions of the same dataset. This might enable handling several releases of the same dataset and enriching the results of our analyses. In addition, we aim to address novel challenges concerning the effects of class level changes on links. The study of these issues must pave the way for the definition and development of mechanisms suited to the maintenance of links without requesting the reapplication of link discovery techniques. This must benefit the RDF dataset maintainers and underlying software application relying on links to favor their execution and effectiveness.

6 Conclusion

Handling the impacts of RDF datasets evolution in established links is essential to maintain the consistency and benefits of LOD over time. This is specially true and valuable for the life science RDF datasets, which are heavily interconnected. In this paper, we investigated the behaviour of links based on the evolution of RDF datasets. We conceived and developed analyses based on the computation of RDF dataset deltas to understand their influence in the links of the dataset.

We found that complex changes (modification of triples) generate further impact in the links than simple changes, such as simple (atomic) addition and removal of triples. This result stands for a starting point in the investigation of a solution to preserve links automatically over time. Future work involves the definition and implementation of additional analyses and their execution with larger datasets in other domains. We plan to investigate the correlation

[10] http://aims.fao.org/standards/agrovoc/linked-data

between changes in ontological level and their corresponding instances. We will investigate the elaboration of a link maintenance mechanism informed by the lessons learned in these analyses.

Acknowledgements

This work was financially supported by the São Paulo Research Foundation (FAPESP) (grants #2017/02325-5, #2018/08082-0, #2018/05357-8 and #2018/14199-7)[11].

References

1. Dos Reis, J.C., Pruski, C., Da Silveira, M., Reynaud-Delaître, C.: Understanding Semantic Mapping Evolution by Observing Changes in Biomedical Ontologies. Journal of Biomedical Informatics **47**, 71–82 (2014)
2. Dos Reis, J.C., Pruski, C., Da Silveira, M., Reynaud-Delaître, C.: DyKOSMap: A framework for mapping adaptation between biomedical knowledge organization systems. Journal of Biomedical Informatics **55**, 153–173 (2015)
3. Groß, A., Reis, J.C.D., Hartung, M., Pruski, C., Rahm, E.: Semi-Automatic Adaptation of Mappings between Life Science Ontologies. In: Proceedings The 9th International Conference on Data Integration in the Life Sciences. pp. 90–104 (2013)
4. Gruber, T.R.: Toward principles for the design of ontologies used for knowledge sharing. International Journal of Human-Computer Studies **43**, 907 – 928 (1995)
5. Käfer, T., Abdelrahman, A., Umbrich, J., O'Byrne, P., Hogan, A.: 10th Extended Semantic Web Conference, chap. Observing Linked Data Dynamics, pp. 213–227. Springer (2013)
6. Liu, F., Li, X.: Using metadata to maintain link integrity for linked data. In: 4th Int. Conference on Cyber, Physical and Social Computing. pp. 432–437 (2011)
7. Papastefanatos, G., Stavrakas, Y.: Diachronic linked data: Capturing the evolution of structured interrelated information on the web. Ercim news **52**, 35–37 (2014)
8. Popitsch, N., Haslhofer, B.: 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 (2011)
9. Pourzaferani, M., Nematbakhsh, M.A.: Repairing broken rdf links in the web of data. Int. J. Web Eng. Technol. **8**(4), 395–411 (2013)
10. Roussakis, Y., Chrysakis, I., Stefanidis, K., Flouris, G., Stavrakas, Y.: 14th International Semantic Web Conference, chap. A Flexible Framework for Understanding the Dynamics of Evolving RDF Datasets, pp. 495–512. Springer (2015)
11. Scharffe, F., Euzenat, J.: Melinda: an interlinking framework for the web of data. arXiv preprint arXiv:1107.4502 (2011)
12. Vesse, R., Hall, W., Carr, L.: Preserving linked data on the semantic web by the application of link integrity techniques from hypermedia (2010)
13. Volz, J., Bizer, C., Gaedke, M., Kobilarov, G.: Discovering and maintaining links on the web of data. In: Bernstein, A., Karger, D.R., Heath, T., Feigenbaum, L., Maynard, D., Motta, E., Thirunarayan, K. (eds.) International Semantic Web Conference (ISWC). pp. 650–665. Springer (2009)

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