Link maintenance for integrity in linked open data evolution: literature survey and open challenges

Andre Gomes Regino [a], Julio Cesar dos Reis [a,d], Rodrigo Bonacin [b], Ahsan Morshed [c] and Timos Sellis [e]

[a] *Institute of Computing, University of Campinas, SP, Brazil*
E-mails: andregregino@gmail.com, jreis@ic.unicamp.br
[b] *, UNIFACCAMP and Center for Information Technology Renato Archer, SP, Brazil*
E-mail: rodrigo.bonacin@cti.gov.br
[c] *, Central Queensland University of Technology,Melbourne, Australia*
E-mail: a.morshed@cqu.edu.au
[d] *Nucleus of Informatics Applied to Education, University of Campinas, SP, Brazil*
E-mail: jreis@ic.unicamp.br
[e] *, Swinburne University of Technology,Melbourne, Australia*
E-mail: tsellis@swin.edu.au

Editor: Oscar Corcho, Universidad Politécnica de Madrid, Spain
Solicited reviews: Mikel Emaldi Manrique, Universidad de Deusto, Spain; Two anonymous reviewers
Open review: Mohamed Ahmed Sherif, University of Leipzig, Germany

Abstract. RDF data has been extensively deployed describing various types of resources in a structured way. Links between data elements described by RDF models stand for the core of Semantic Web. The rising amount of structured data published in public RDF repositories, also known as Linked Open Data, elucidates the success of the global and unified dataset proposed by the vision of the Semantic Web. Nowadays, semi-automatic algorithms build connections among these datasets by exploring a variety of methods. Interconnected open data demands automatic methods and tools to maintain their consistency over time. The update of linked data is considered as key process due to the evolutionary characteristic of such structured datasets. However, data changing operations might influence well-formed links, which turns difficult to maintain the consistencies of connections over time. In this article, we propose a thorough survey that provides a systematic review of the state of the art in link maintenance in linked open data evolution scenario. We conduct a detailed analysis of the literature for characterising and understanding methods and algorithms responsible for detecting, fixing and updating links between RDF data. Our investigation provides a categorisation of existing approaches as well as describes and discusses existing studies. The results reveal an absence of comprehensive solutions suited to fully detect, warn and automatically maintain the consistency of linked data over time.

Keywords: link integrity, link maintenance, RDF evolution

1. Introduction

In recent years, a large number of knowledge bases interconnected on the Web have emerged describing various types of resources in a structured way. In particular, Linked Data (LD) refers to machine-readable data connecting datasets across the Web [1], by exploring Semantic Web technologies such as Resource Description Framework (RDF)[1] and computational on-

[1] https://www.w3.org/TR/WD-rdf-syntax-971002/

tologies [2]. RDF refers to a graph-oriented data model suited to represent metadata about Web resources.

Links between RDF descriptions are at the heart of the Web of Data. The growing number of structured data published as RDF repositories in the Web confirms the real potentiality of the global data space proposed by the Semantic Web vision. Indeed, connections between data elements described via RDF models are in the center of the Semantic Web [3]. The interconnection of RDF statements, via explicit links, plays a central role in this scenario to assure data linkage. The links allow previously isolated bases to be explored in combination.

RDF statements defining real-world resources are subject to change when the domain updates. This evolution comes from emerging number of contributions by governments, private institutions, wiki databases (such as DBpedia)[2] to create a big and well-formed Linked Open Data (LOD), available for anyone to consume and contribute. In this scenario, RDF triples [4], which are the basic statements performed in RDF datasets, can be 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 [5], these operations can affect several established links, which might turn them invalid or inconsistent. In addition, ontologies, vocabularies and data schemes can change the definition and structure of RDF data. This might entail consequences on the respective data definition as well as links between datasets.

This scenario hampers data linkage consistency over time. The manual maintenance remains hardly accomplishable due to the overwhelming number of links available. Often external links disappear without notifying their dependants. Since links should not be recomputed each time a change occurs, novel methods are required to adequately consider the evolution. The evolution of the RDF datasets should be as much automated as possible, even though the possibility provided to users edit their content and validate changes should be assured [6]. This evolutionary characteristic of the LOD causes the link maintenance problem which is in the root of a research line known as the broken links.

The constant and evolving process of updating datasets demands the study and development of novel methods and software tools [7]. In general, RDF resources are linked by semi-automatic algorithms [8] [9] and these links are manually evaluated, which involves huge amount of labour cost and time. Usually RDF links between data sources are updated only sporadically, which leads to dead (broken) links pointing to URIs that are no longer maintained [3]. Currently, there is a huge mass of interconnected data that requires automatic methods and tools to deal with consistency aspects. In this context, Web dynamics tends to update data definitions on an isolated basis [10], [11]. This aspect sets up a challenging research scenario to deal with the controlled evolution of interconnected datasets. The design and implementation of novel software tools must concern these factors and address them in various perspectives. These aspects represent serious obstacles towards a fully automatic solution, requiring a complete and exhaustive survey of existing approaches focused on addressing this issue.

With the advent of the LOD, researches emerged to study problems caused by its exponential expansion, ranging from scalability issues [12], URI synonymity [13] as well as dataset quality [14]. However, the literature has superficially studied the maintenance of linked datasets [15] and the mapping evolution phenomena between biomedical ontologies [16]. To the best of our knowledge, it lacks thorough investigations empirically grounded to unveil how links evolve in the context of linked datasets. In our previous work, we empirically analysed several cases of link modifications by studying the evolution of datasets in the domain of life sciences [17]. Our study investigated if there were correlations between changes in triples and changes in links by considering 12 scenarios involving addition, removal and modification of triples and links in the Agrovoc[3] dataset. Nevertheless, our investigation shows a lack of a systematic literature analysis addressing such research problems to help understanding relevant properties that might support the definition of automatic mechanisms for link evolution in LOD.

In this article, we provide a systematic literature survey to thoroughly understand existing contributions addressing link maintenance. In summary, we make the following contributions:

- We formally define and illustrate the link maintenance problem, highlighting the complexity of the problem. We illustrate examples to clarify the involved issues and explore them throughout this article.

[2] https://wiki.dbpedia.org/

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

- We systematically review the literature on the link maintenance problem, offering a comprehensive state-of-the-art by presenting, comparing and discussing existing proposals in several categories identified via our literature analysis.
- We analyse lacks of existing approaches discussing open issues that the literature fails to address towards a fully automatic link maintenance. This allows us to underscore open research challenges.

In the view adopted in this paper, a systematic literature review identifies the extent and form the literature on a topic to obtain a broad review of key studies from a specific topic during the initial examination of a new domain. A literature review is valuable to gather existing information about a subject in a formal, complete, impartial and meticulous manner [18]. Our methodology was conducted based on a series of steps responsible for planning, retrieving and analysing the scientific papers selected on the search on huge scientific databases. The methodology involved defining research questions to be answered through this investigation, establishing the terms and in which databases the queries are performed. We defined inclusion and exclusion criteria for selecting the relevant retrieved papers. Our literature analysis provided an organisation of the papers based on distinct categories as approaches related to link maintenance problem.

The results achieved via a careful analysis of the literature indicate that there are solutions for detecting broken links - some of them with scalability issues given the size of the datasets. However, none of them are able to fix broken links of any kind, in the context of Linked Data, without the assistance of human throughout the process.

The remaining of this paper is organized as follows: Section 2 defines and formalizes the link maintenance problem; Section 3 describes in details the methodology conducted in our systematic literature review; Section 5 presents the obtained results indicating our literature analysis and explaining the categories of solutions found related to the problem; Section 6 carefully discusses our findings and reports on open challenges of different nature with unsolved research questions. Finally, in Section 7, we wrap up the article with concluding remarks and outline future work.

2. The link maintenance problem

Before we formally define the link maintenance problem, we provide basic concepts referring to Linked Data and its essential elements.

Linked Data refers to "*[...] a set of best practices for publishing and connecting structured data on the Web in a way that data is machine-readable, its meaning is explicitly defined, it is linked to other external datasets*" [1]. The Linking Open Data initiative, known as LOD cloud, started its activity in 2007 with the premise of being a "grassroots community effort to bootstrap the Web of Data by interlinking open-license datasets" [1]. From that period, the LOD cloud has grown substantially. Nowadays, a huge quantity of RDF datasets has been published and present interconnections from one dataset with others.

RDF dataset. A dataset in the context of Linked Data is a conglomeration of a finite number of RDF triples in a domain [19]. RDF triples are statements composed of unique URIs[4][4], which identify resources in a dataset. Formally, $\mathcal{R} = (t_1, t_2, t_3, ..., t_n)$.

Triple. In a dataset, a triple unites two nodes (or resources) using a property (A node, that can also be called a resource, is the instance of a given class). A resource can be anything described in the "real world", either a physical thing like a computer, or a concept like "theory of relativity" [19]. In RDF, the resources are represented as *Uniform Resource Identifier* (URI). An RDF triple [4] 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 a URI reference or a blank node.
- **Predicate:** (p) is a URI reference as property defining characteristics of an individual in an ontology class.
- **Object:** (o) is either a 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. As an example of triple considering the notation (s, p, o), we describe the Abraham Lincoln' birthday date and place as follows:

[4] https://www.w3.org/wiki/URI

- *dbr:Abraham_Lincoln dbo:birthDate "1809-02-12"^^xsd:date*;
- *dbo:birthPlace dbr:Hodgenville,_Kentucky*.

Ontology. An ontology $\mathcal{O}$ describes a domain in terms of concepts, attributes and relationships [2]. Formally, an ontology $\mathcal{O} = (\mathcal{C}_{\mathcal{O}}, \mathcal{S}_{\mathcal{O}}, \mathcal{A}_{\mathcal{O}})$ consists in a set of classes $\mathcal{C}_{\mathcal{O}}$ interrelated by directed relationships $\mathcal{S}_{\mathcal{O}}$. Each concept $c \in \mathcal{C}_{\mathcal{O}}$ has a unique identifier and it is associated to a set of attributes $\mathcal{A}_{\mathcal{O}}(c) = \{a_1, a_2, ..., a_p\}$.

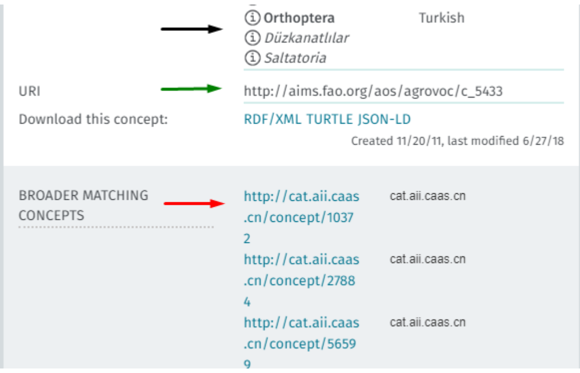
Link. Besides the use of triples in a dataset, the linkage among several datasets is essential for Linked Data. There is a link joining two distinct datasets if a predicate is established between a subject in the first dataset (source) and an object in the second (target). Formally, we define a link 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]. The following list shows some examples of links discovered and processed by existing methods and tools analyzed in our study:

- $l_1 =< r_a, owl : sameAs, r_b >$
- $l_2 =< r_a, rdfs : seeAlso, r_b >$
- $l_3 =< r_a, owl : differentFrom, r_b >$
- $l_4 =< r_a, skos : exactMatch, r_b >$
- $l_5 =< r_a, skos : closeMatch, r_b >$

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 $\mathcal{L}_{\mathcal{ST}} = \{l_0, l_1, l_2, ..., l_n\}$.

The connection between nodes of a source and destination node can be broken by several reasons, which changes the state of the link to broken or invalid. According to Popitsch and Haslhofer, a link is broken when "[...] *the representations of the target resource were updated in such a way that they underwent a change in meaning the link-creator had not in mind*" [9].

Some authors categorize the broken links in two main groups. A link is structurally broken, as stated by Singh, Brennan and O'Sullivan, "if either source or target are no longer dereferenceable" [20]; and also by Popitsch and Haslhofer "if its target resource had representations that are not retrievable anymore" [15]. A link is semantically broken when the semantic of data in the target dataset is different from the semantic of the source [21]. Semantically broken links are harder to detect and fix [15].



Orthoptera Turkish
Düzkanatlılar
Saltatoria

URI http://aims.fao.org/aos/agrovoc/c_5433

Download this concept: RDF/XML TURTLE JSON-LD

Created 11/20/11, last modified 6/27/18

BROADER MATCHING CONCEPTS

http://cat.aii.caas.cn/concept/10372	cat.aii.caas.cn
http://cat.aii.caas.cn/concept/27884	cat.aii.caas.cn
http://cat.aii.caas.cn/concept/56599	cat.aii.caas.cn

Fig. 1. Real-world example of broken link

When a broken link is found by someone in traditional hypermedia, the one who found it might search for alternative paths to the same or similar location, different from the machine-to-machine communication present in linked data [15]. Vesse *et al.* [22] divided the link integrity in two main categories: (1) the dangling link, where the destination node no longer exists and takes to nowhere; and (2) the editing problem, in which the destination node still exists, but the semantics of this node changed over time and the link does not provide any reliable information.

As an example of broken link, we found a real scenario at Agrovoc linked open data[6]. When we search for the term "orthoptera" (which is an animal) using the Agrovoc web page[7], we find results similar to as presented in Figure 1. It shows the Linked Data associated to the searched resource. The black arrow indicates some translations of the term; the green arrow shows the unique URI of that resource in the Agrovoc dataset[8]. The red arrow shows connections to other datasets, linked to Agrovoc, and their corresponding matching related to "orthoptera". The first one refers to a broken link because the target resource is no longer available.

At this stage, we introduce the notion of time $j \in \mathbb{N}$. Consider a link l^j at time j and a link l^{j+1} at different specific time based on distinct releases of the associated datasets. Modifications occurring in the re-

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

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

[7] http://aims.fao.org/standards/agrovoc/functionalities/search

[8] represented by the end of the URI, *c*_5433

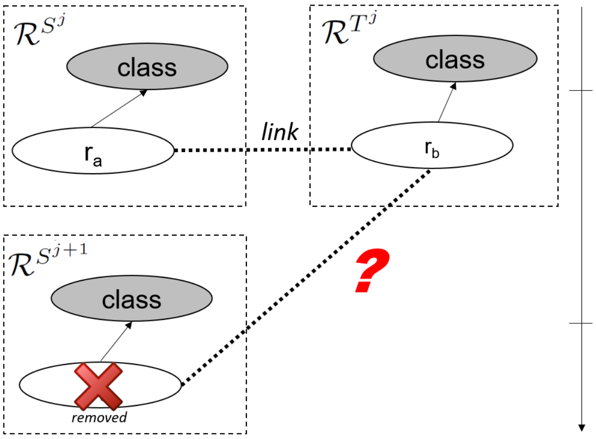


Fig. 2. Link maintenance problem

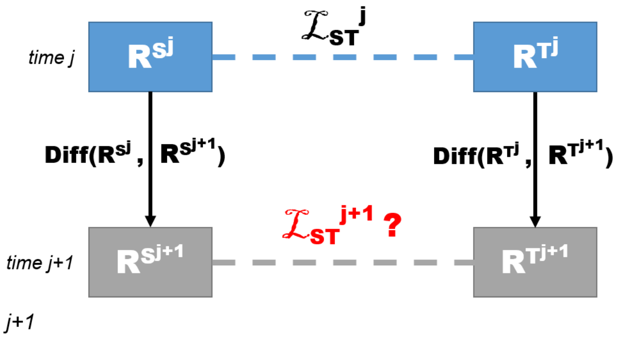

time j
R^{Sj}
$\mathcal{L}_{ST}^{j}$
R^{Tj}
Diff(R^{Sj}, R^{Sj+1})
Diff(R^{Tj}, R^{Tj+1})
$\mathcal{L}_{ST}^{j+1}$?
time j+1
R^{Sj+1}
R^{Tj+1}

Fig. 3. Problem modelling

lated resources (r_a or r_b) from a release of the dataset at a specific time j to a time $j+1$ can invalidate a 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 $l^{j+1}(r_c \rightarrow r_b)$, such that, in this case, $r_a \neq r_c$. Figure 2 illustrates the link maintenance problem.

An example of semantically broken link caused by the evolution of the dataset is the "London" example. In dataset A we have the resource with label "London", as long as in dataset B. They are connected by a "sameAs" predicate, forming a link between the "London" resources. The maintainer of the ontology A decides to change the resource from "London" to "Greater London". Now, "Greater London" from dataset A is linked to "London" in dataset B. London and Greater London are different in many aspects such as population number, area and climate. The link became semantically broken. It is not structurally broken because the resources still exist in the datasets.

The evolution of RDF datasets in terms of changes affecting its triples may invalidate previously determined links. Figure 3 presents the general scenario of investigation. Since we consider RDF datasets evolution, it is necessary to examine different versions of each dataset. A $Diff(\mathcal{R}^{S^j}, \mathcal{R}^{S^{j+1}})$ refers to the operation necessary to identify modifications from one dataset version to another.

In order to maintain the consistency of the dataset, its links should remain in an integrity state, even with recurrent changes in the data. Popitsch and Haslhofer define link integrity as "[...] *a qualitative property that is given when all links within and between a set of data sources are valid and deliver the result data intended by the link creator*" [23].

In this investigation, we concern the research problem named link maintenance, whose aims are:

- To provide innovative mechanisms to guarantee that there is a minimum occurrence of broken links;
- To consider change operations from one dataset to another aiming to inform operations of update in links, based on information on how the nature of these changes is affecting the links;
- To consider the history of changes to retrieve older links and maximize data reliability [24];
- If unsolved occurrences of broken links remain, the responsible for maintaining data integrity should be notified.

3. Methodology for the systematic literature review

This work investigates the research literature related to link integrity and maintenance. This review provides a key contribution to a better understanding and existing solutions on this topic. A complete investigation about the subject provides a concise understanding of common approaches and advancements in the scenario under investigation. Our study aims to guide researchers concerning the state-of-the-art status in addition to unveil the drawbacks and open issues in recent studies.

We adopted the guidelines proposed by Budgen and Brereton to perform the review [18]. Our methodology considers two key phases named planning and conduct. Figure 4 presents the steps in these two phases of the methodology.

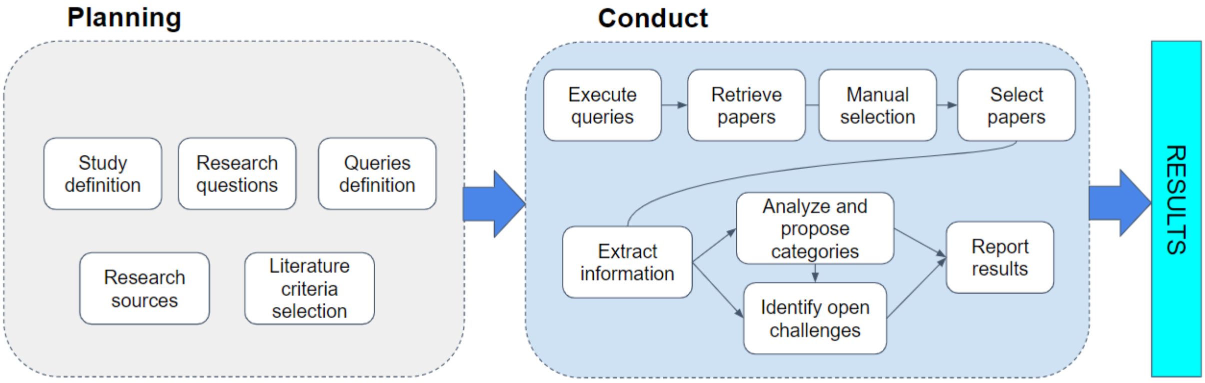


Fig. 4. Systematic literature review process

The planning phase defines activities related to the review protocol, organising the steps of the literature research. The protocol defined for the literature search identifies the research questions and a establishes search strategy comprising inclusion, exclusion and quality criteria to evaluate the studies. The conduct phase executes the defined protocol via specific activities to retrieve, select and analyse the papers. Section 3.1 describes the steps involved in the planning phase, while Section 3.2 reports the activities in the conduct phase.

3.1. Planning phase

Study definition. The first step was determining whether there was a need for a review and ensuring the relevance of conducting a literature study. An initial and exploratory study confirmed that some relevant publications had introduced the subject and presented related literature. For example, some investigation defined the broken link phenomenon and investigated specific concerns in link integrity. However, the existing literature did not provide a complete and updated overview of the area. In addition, no previous systematic literature reviews were found on the topic. Table 1 presents the key general questions that define this study.

Research questions definitions. We precisely defined the research questions necessary to be answered as the outcome of this study. The research questions require inquiring subtopics needing enough studies for detailed system review. Table 2 presents the research questions, which where produced on the base of discussions among the authors of this paper. These key questions aim to better drive the search, extraction and analyses of results.

Query strings definition. We gathered terms used to search for papers. The search terms used were as simple as possible. We avoided complex search terms because the goal was to recover a substantial quantity of papers at the initial stage of the survey.

The search terms were built upon strings related to the link maintenance problem. Approaches to maintain links matter for this work (*e.g.*, create new links, reconstruct erased ones, modify semantically wrong links or even detect and notify about them). Table 3 presents the defined search queries.

We clarify that the order of the search terms and the search logic connectors used in the entire process of search, such as double quotes and connectors like AND/OR/+/-, are relevant to retrieve the exact papers referred to in this survey. We set parameters at the research according to database specificities (defined in the next step). In particular, we considered 100 as the maximum number of retrieved results in Google Scholar.

Research sources. The fourth step was responsible for defining the scientific databases in which the searches were run to retrieve an expressive number of high-quality research papers. We selected the following databases: Web of Science, Scopus, ACM Digital Library, IEEE Xplore, Elsevier Science Direct and Springer. Other sources like Google Scholar were further explored in our literature research. The databases were chosen on the base of their reputation and due to the publication type variety. We assumed that these repositories cover high quality publications with some

Table 1
Study Definition.

Question	Definition
Why?	Investigate and understand the state of the art concerning how the link maintenance problem is solved in the Linked Data context
Where?	Via the existing literature related to the area of study, such as: Web Semantics, Ontologies, Linked Data
What?	Uncover the unsolved open research questions and potential solutions revealed in the literature on link integrity in LOD

Table 2
Research Questions.

RQ	Question
RQ-01	What are the benefits of having an RDF dataset with no or very few broken links?
RQ-02	What are the types of broken links?
RQ-03	What are the existing proposals for solving broken links outside the Linked Open Data community?
RQ-04	What are the existing solutions for the link integrity and link maintenance problem in the Linked Open Data community?
RQ-05	Are there fully automated approaches to maintain links up to date?

Table 3
Search queries.

Query	Terms description
Q-01	"web semantics" + "link integrity"
Q-02	"semantic web" + "link integrity"
Q-03	"link maintenance" + "semantic web"
Q-04	"evolution knowledge" + rdf + "data integrity"
Q-05	"evolution knowledge base" + "rdf dataset"
Q-06	"broken links" + "web of data"
Q-07	rdf + "link maintenance"
Q-08	"rdf link integrity"
Q-09	"link integrity" + rdf -hypertext
Q-10	"link broken" + "semantic web"
Q-11	"evolution of rdf" + "rdf dataset"
Q-12	"interlink problem" + "semantic web"
Q-13	"linked data" + "data fusion"
Q-14	"linked data" + "data linking"
Q-15	"link changes" + "linked data"
Q-16	"semantic web" + "data fusion" + "broken link"
Q-17	"dataset evolution" + rdf
Q-18	survey + "link integrity" + "linked data"
Q-19	survey + "link integrity" + "linked open data"
Q-20	"mapping maintenance" + "linked data"
Q-21	"mapping links" + "linked data"
Q-22	"rdf annotation" + "linked data"
Q-23	"change detection" + rdf
Q-24	"record linkage" + "semantic web"
Q-25	"ontology alignment" + rdf

degree of impact, guaranteeing information quality for the research.

Literature criteria selection. This step of the review concerned which papers should contribute to this survey. The study selection criteria determine which studies are included in or excluded from the review. Table 4 presents the defined exclusion criteria and Table 5 reports on the inclusion criteria.

The motivation for the exclusion criteria was based on the necessity to retrieve a well-defined set of papers, meeting a quality and content criteria. We understand that the defined exclusion criteria help obtain high quality papers in the literature considering: papers that are not too old (EC-01), papers noncompliant with academic best-practices (EC-02 and EC-03); and articles whose addressed problem does not contribute to the understanding of the link maintenance challenges (EC-04).

3.2. Conduct phase

Retrieval and selection of papers. The queries were run in the determined scientific databases. Their initial selection relied on their content including: title, abstract and conclusion. Some retrieved articles were discarded in this step because their content was not close enough to the goals in this investigation, based on the exclusion and inclusion criteria presented, respectively, in Table 4 and Table 5.

This initial selection considered finding key terms in the title and abstract, based on the keywords in Table 3. For instance, "*Repairing Broken RDF Links in the Web of Data*" and "*An Approach for Discovering and Maintaining Links in RDF Linked Data*", which present terms such as "broken rdf links" and "maintaining links in rdf".

Those papers meeting all the inclusion criteria and not fitting the exclusion criteria were retrieved in our selection. For a given paper, whether one of the inclu-

Table 4

Exclusion Criteria.

EC	Type	Definition
EC-01	**Date**	Papers 15 years older than the execution of the queries in January 2019
EC-02	**Abstract**	Papers without abstract
EC-03	**Language**	Papers not written in English
EC-04	**Application**	Papers that just mention a tool that handles broken links, but which do not explain how and why it was used

Table 5

Inclusion Rules.

EC	Type	Definition
IC-01	**Application**	Book chapters, conference papers (full and short articles), journals and thesis
IC-02	**Application**	Papers of type as listed at IC-01 that create, use or theoretically define a way to detect, notify or fix broken links to keep link integrity and maintenance

sion criteria had not been met or one exclusion criteria had been met, then such paper was discarded. The selection was made after the common agreement of the researchers. The Appendix in Section 7 presents details regarding the returned results for each query in the research sources.

In addition to the retrieval based on the defined queries, we performed an additional manual step in retrieving and selecting relevant articles, respecting the inclusion and exclusion criteria. In this step, we searched for additional articles in Google scholar, specific conferences and workshops whose papers are not indexed in the considered databases. Our goal in this step was to obtain and analyse articles not retrieved from the chosen sources of information. We consider this step relevant to evaluate additional investigations that can contribute to improve this survey. Additionally, we considered the selection of key correlated surveys connected to our topic of interest. The collected and presented surveys can help in clarifying the open challenges and organizing existing outcomes.

Information extraction. This activity consisted in the analysis of textual content in the articles of the initial retrieved sample. The data was extracted by reading and processing the following sections of the articles: references, title, abstract, keywords, conclusion and when necessary the full content of the article.

Categories definition. The rigorous analysis of selected papers via their careful reading enabled the proposition of categories to classify and organise the selected papers. Table 6 shows the created categories and briefly describes their goals. For each paper of our selection, we assigned only one category from those categories defined. This step of the conduct phase generated 9 categories, 8 of which containing research papers and 1 containing survey papers.

Section 5 presents the description of the articles obtained in each category. We clustered them depending on how they propose a solution based on the research questions presented in Table 2.

Identification of open challenges. At this stage, we carefully analysed the conducted survey to highlight a set of open issues concerning the link maintenance problem. The authors in a common agreement detected and categorised major research issues that still deserve further research. We provided unaddressed research challenges to address link maintenance (cf. Section 6.3).

Description of results. The last step involved the adequate description of results from the retrieved and analysed investigations in the different categorisations (*cf.* Section 5). We provide a comparative analysis carefully conducted to further understand the achievements and limitations from the literature (*cf.* Section 6.1). We present and illustrate the open issues in different topics (*cf.* Section 6).

4. Publication analysis

This section presents a numeric publication analysis of the papers in our survey.

Our selection and analysis led to a total of 28 papers by considering the defined inclusion and exclusion criteria. Regarding the year of publication, the oldest article was published in 2006 whereas the newest one was published in 2018. This indicates that the problem investigated in this survey and the literature around it is mature, but there are still researchers seeking to solve open issues.

Table 6

Categories of approaches.

Category	Definition
Change Detection	Generates deltas (diffs) based on changes detected at the dataset
Metadata Storage	Custom data stored along with the nodes of the dataset to help detecting and fixing links
High Level Modifications	Represents the changes in a more human-readable group of modifications over time
Ontology-driven Change Representation	Builds a temporary or permanent ontology to support the detection of broken links
Hypermedia-based Approaches	Approaches handling link integrity problem at the traditional Web
Link Management Mechanisms	Track, discover, build links and fix them whenever possible
Hybrid Solutions	Mix of previous solutions + broken link detection
Survey Papers	Papers focused on the analysis of the literature on link evolution

Table 7

Results by Scientific Database

Database	#
ACM DL	4
IEEE Xplore	2
SpringerLink	5
Elsevier Scopus	2
Others	15

Table 8

Types of Publication.

Type	#
Book Chapter	7
Conference - Full Paper	9
Conference - Short Paper	0
Conference - Workshop	2
Journal	6
Thesis	1

Table 7 shows the articles retrieved from each of the main scientific databases. The others refer to Google Scholar and ArXiv.

Table 8 summarises the number of articles for the different types of publication. Most of them are book chapters and full papers in conferences. Only one thesis was found in the survey.

Table 9 summarises the number of papers retrieved by each query string. The line identified by the word “reference” is related to the papers that were found at the section of references of the retrieved papers.

5. Link Maintenance Approaches

In this section, we describe and analyse the studies obtained in each of the categories.

Table 9

Results by Query String.

Query	Papers Retrieved
"web semantics" + "link integrity"	4
"link maintenance" + "semantic web"	4
survey + "link integrity" + "linked data"	3
"link integrity" + rdf -hypertext	3
evolution knowledge base rdf	1
interlink problem web semantic	2
"linked data" "data fusion"	3
***reference**	8

5.1. Change detection

The act of identifying, storing and retrieving different versions of linked datasets is used in the process of defining which resources were modified. In this scope, we discuss solutions for computing and representing change detections in RDF datasets. We understand that for addressing link maintenance issues, studies concerning the detection of modifications in RDF datasets are very relevant. This is justified by the fact that changes affecting resources in the dataset can be a source for recognizing and fixing broken links. Via the comparison of different versions of the same dataset, a broken link can be potentially identified and fixed.

Change detection approaches are intrinsically related to the area of ontology evolution, resulting in researches to compare versions of the same ontology to detect the differences between different versions (releases) [10]. Our survey focused on studies reporting on techniques for addressing change detection on RDF datasets.

In this category, Powl [10] refers to a versioning framework aiming to detect and store atomic changes performed in ontologies, joining them and creating hierarchic and compound changes. This framework assists the process of ontology validation after sveral changes after a certain elapsed time. The storing of changes is performed via atomic changes. Compound changes combine distinct atomic changes to improve the readability of the changes for humans [10]. Figure 5 presents the user interface of the Powl framework for the management of RDF versioning. It presents the versioning tab with a table containing examples of changes at a given date and in the last column the possibility to rollback to a specific version.

In the context of versioning and ontology evolution, a delta expresses the difference between two versions of the same dataset. One of the limitations found in the versioning context is the size of deltas files created to map the differences between two versions of a dataset. Lee, Im and Won [12] proposed an algorithm to decrease the number and size of deltas produced by state-of-the-art similarities algorithms, using MapReduce as a framework for distributed and parallel computing. Table 10 presents a summary of the articles selected in this category.

5.2. Metadata Storage

In the context of this study, the concept of metadata is related to custom data stored with the nodes of the dataset to help detecting and fixing broken links. Some of the implementations of metadata store the current state of the link, for example “created”, “changed” or “removed”. Other metadata tag the resources with semantic descriptive tags, such as “Sherlock Holmes” > “Watson” and “IBM” > “Watson”. Table 11 presents a summary of the articles selected in this category.

Zuiderwijk, Jeffery and Janssen [25] argue about the relevance of using metadata for finding, storing, analysing, visualising and other advantages of data manipulation in LOD. The authors conducted a literature review of metadata used in LOD research. They listed eighteen directives for a concise metadata structure and validated these directives.

Kovilakath and Kumar suggested an approach to detect semantically broken links based on stamping hierarchic tags on the resource. When the resource changes, a publish/subscribe module stores and signals that some resource may be broken. The use of metadata plays a key role for their semi-automatic method, which detects semantically broken links. Although their approach enables identifying the broken links, it cannot fix the links [21].

5.3. High-Level Modifications

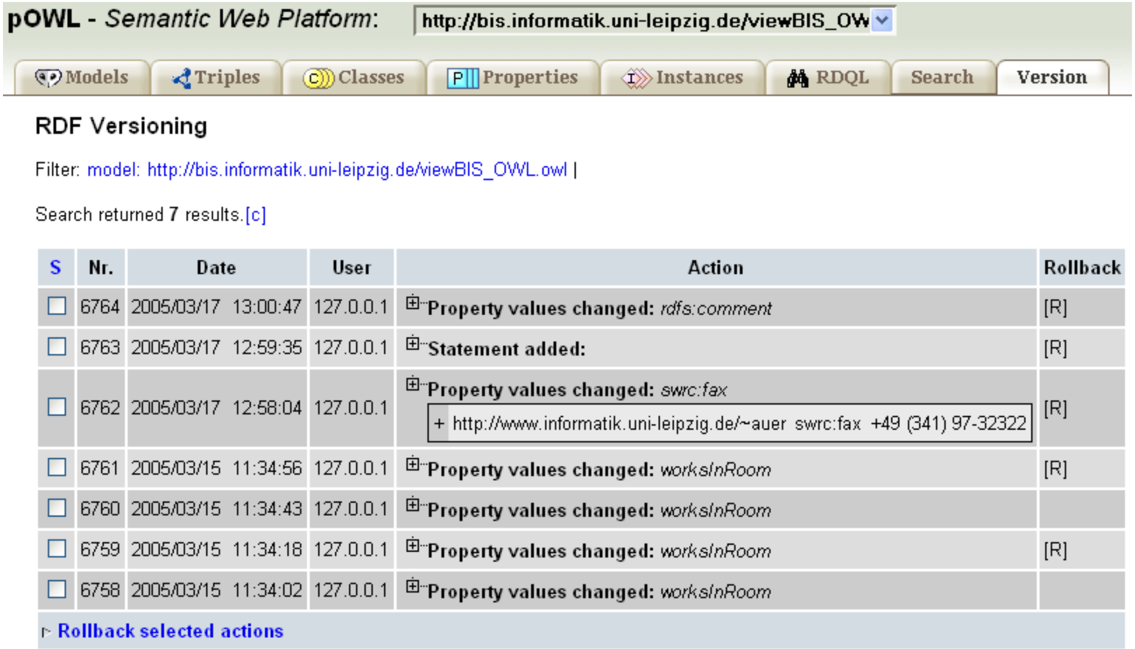
High-level changes aim to represent what has changed in the dataset in a non-atomic way. This facilitates the understanding of new, updated or deleted information over time. The so-called low-level changes are the atomic changes that add and delete nodes, links and triples in the dataset. High-level changes aggregate the atomic changes via functions, which creates a more semantic-based change definition, intuitively describing what the intention of the user who performed the action in the ontology was. For example, an addition of an employee to a given ontology can be expressed as a group of triples that sets the employee’s name, salary and skills. This group of triples is an example of a high-level and complex change, representing the set of triples in a single change of employee’s addition. Table 12 presents a summary of the articles selected in this category.

According to Galani *et al.* [26], the identification and versioning of simple events in the dataset cannot express the semantic of the change. These authors proposed a language to manage and define complex changes in RDF datasets. The concept of complex changes is analogous to the high-level change described by Roussakis *et al.* [7]: changes easy to be deduced and produced by users. The authors emphasized the relevance of complex changes, since the detection and versioning of simple changes are not suited to explain how and why the RDF data changed. In addition, simple changes cannot express with precision the semantics of the change [26]. The authors applied both proposed language and algorithm in biology datasets, which benefits from the versioning method because the evolution of these datasets is an important requirement for their consistency.

According to Papavasileiou *et al.* [27], the use of high-level changes originates an increase in the complexity to detect the changes. The increase in the amount of these types of changes and in complexity causes an increase in the level of abstraction. The authors proposed a language that creates and uses small and intuitive deltas without losing the expressiveness of the changes. The work presented by Papavasileiou *et al.* [28] is an extension of Papavasileiou *et al.* [27].

5.4. Ontology-driven change representation

Ontologies can be usable to represent changes in RDF datasets. The adequate change representation is assumed in this category as essential to facilitate so ontology experts verify ontology evolution to accept changes and better understand their effects. Table 13 presents a summary of the articles selected in this category.



pOWL - *Semantic Web Platform*: http://bis.informatik.uni-leipzig.de/viewBIS_OW

RDF Versioning

Filter: model: http://bis.informatik.uni-leipzig.de/viewBIS_OWL.owl |

Search returned 7 results.[c]

S	Nr.	Date	User	Action	Rollback
□	6764	2005/03/17 13:00:47	127.0.0.1	⊞ **Property values changed:** *rdfs:comment*	[R]
□	6763	2005/03/17 12:59:35	127.0.0.1	⊞ **Statement added:**	[R]
□	6762	2005/03/17 12:58:04	127.0.0.1	⊞ **Property values changed:** *swrc:fax* + http://www.informatik.uni-leipzig.de/~auer swrc:fax +49 (341) 97-32322	[R]
□	6761	2005/03/15 11:34:56	127.0.0.1	⊞ **Property values changed:** *worksInRoom*	[R]
□	6760	2005/03/15 11:34:43	127.0.0.1	⊞ **Property values changed:** *worksInRoom*	
□	6759	2005/03/15 11:34:18	127.0.0.1	⊞ **Property values changed:** *worksInRoom*	[R]
□	6758	2005/03/15 11:34:02	127.0.0.1	⊞ **Property values changed:** *worksInRoom*	

Rollback selected actions

Fig. 5. Versions and Reviews in the Powl framework [10]

Table 10
Category: Change detection.

Paper ID	**Title**	**Author(s) and Reference**
VERSIO-1	A Versioning and Evolution Framework for RDF Knowledge Bases	Auer, Sören and Herre, Heinrich [10]
VERSIO-2	Similarity-based Change Detection for RDF in MapReduce	Lee, Taewhi and Im, Dong-Hyuk and Won, Jongho [12]

Table 11
Category: Metadata Storage.

Paper	**Title**	**Author(s) and Reference**
METADA-1	The Potential of Metadata for Linked Open Data and its Value for Users and Publishers	Zuiderwijk, Anneke and Jeffery, Keith and Janssen, Marijn [25]
METADA-2	Semantic Broken Link Detection using Structured Tagging Scheme	Kovilakath, Vishnuprakash Puthiya and Kumar, SD [21]

Table 12
Category: High-Level Modifications.

Paper	**Title**	**Author(s) and Reference**
HIGHLE-1	A Language for Defining and Detecting Interrelated Complex Changes on RDF (S) Knowledge Bases	Galani, Theodora, Papastefanatos, George and Stavrakas, Yannis [26]
HIGHLE-2	On Detecting High-level Changes in RDF/S KBs	Papavassiliou, Vicky and Flouris, Giorgos and Fundulaki, Irini and Kotzinos, Dimitris and Christophides, Vassilis [27]
HIGHLE-3	High-level Change Detection in RDF (S) KBs	Papavasileiou, Vicky and Flouris, Giorgos and Fundulaki, Irini and Kotzinos, Dimitris and Christophides, Vassilis [28]

Pernelle *et al.* [29] defined an approach that detects and semantically represents high-level changes of a given dataset. As a result, the authors proposed an ontology for representing the changes.

Kondylakis *et al.* [30] presented a framework that uses provenance queries to identify changes in the ontologies. The framework enables queries to inform when a resource is changed and which operation caused this change. The framework takes log files containing change operations and generates the corresponding instances after the change in a visual manner. This helps ontology experts visualize ontology evolution.

Pourzaferani and Nematbakhsh [24] proposed a tool that detects broken link based on the source node of the link, instead of the destination node. In addition, they defined two auxiliary ontologies: one, named superior, which presents all the subjects and objects of the main ontology; and the inferior ontology, where the subjects of the first ontology become objects and the objects become subjects. These ontologies support the finding of possible resource similarities and new resources to be connected after the discovery of broken links. This technique was proven to get more effective results in fixing links. Using datasets from the domain of "person", the proposed solution repaired more than 90 percent of the broken links. However, this tool cannot repair semantically broken links .

5.5. Hypermedia-based approaches

Link integrity in hypermedia received attention in the late 1980s and early 1990s primarily from researchers in the open hypermedia community [31]. In the traditional Web, most of the traffic runs over the HTTP protocol, where the link integrity problem is present. In our survey, the adaptation of solutions implemented in traditional Web, which might be used in the Web of Data, is discussed by a series of researches. Table 14 presents a summary of the articles selected in the hypermedia-based category.

Vesse, Hall and Carr [22] presented a web interface and service called All About That (AAT), which tracks changes in a dataset and stores it. The AAT uses a concept called URI Profiling, which is related to the storage of old data of a given URI; thus, if an URI is removed, a profile of this URI - including triples, metadata, links - can be retrieved. The linked data is stored in SQL databases. In synthesis, AAT is concerned with data preservation.

Continuing the work in Vesse, Hall and Carr [31], in 2010 an Expansion Algorithm was included into AAT. This algorithm can discover new URIs to links. Given the URI, the algorithm returns similar URIs in the form of sameAs links, acting as a crawler. A new feature - called Default Profile âĂŞ aimed to provide three big data-sources âĂŞ DBpedia, Sindice Cache[9] and sameAs.org[10] - as default sources if the user does not specify any URI Profiling.

Based on the work of Vesse, Hall and Carr, Vesse [32] designed an algorithm for retrieving linked data about the broken URI, which uses links with predicates such as "same as" and "see also". This process is based on link maintenance from the traditional hypermedia. These contributions resulted in a doctoral thesis that proposes a framework for handling broken links based on two solutions for structural broken links implemented in hypermedia.

Another inspiration from the "traditional web" used by linked data is the backlinks, which are registers in databases that points to all places that mentioned a given URL in their web pages. Stefanidakis and Papadakis [33] described and developed a framework to store links and backlinks - in a bidirectional way. The solution provided a consistent way to handle broken links in connected LOD datasets. The proposal helps a given LOD dataset to summarize which datasets are referencing it and being referenced by it.

5.6. Link Management Mechanisms

This category is related to contributions exploring similarity techniques to create links among resources from various types of datasets ranging from multipurpose domains. Some studies described in this category name the linkage of resources in different datasets as "duplicate records detection" or even "record linkage" and "instance linking". The link discovery task can be used as a potential procedure for addressing link maintenance. One of the key steps after the detection of a broken link is to choose what action is to be taken, for instance, exclusion of the link, modification of one of the involved resources or re-connection. In the last case, link discovery techniques can be used to find adequate substitutes. Table 15 summarizes the articles in the link management mechanisms category.

The MeLinDa framework [34] is based on DSNotify [23] and refers to an interlinking framework aiming to

[9] http://www.sindice.com

[10] http://www.sameas.org

Table 13

Category: Ontology-driven change representation.

Paper	Title	Author(s) and Reference
ONTOLO-1	Repairing broken RDF links in the web of data	Pourzaferani, Mohammad and Nematbakhsh, Mohammad Ali [24]
ONTOLO-2	RDF Data Evolution: Efficient Detection and Semantic Representation of Changes	Pernelle, Nathalie and Saïs, Fatiha and Mercier, Daniel and Thuraisamy, Sujeeban [29]
ONTOLO-3	EvoRDF: A Framework for Exploring Ontology Evolution	Kondylakis, Haridimos and Despoina, Melidoni and Glykokokalos, Georgios and Kalykakis, Eleftherios and Karapiperakis, Manos and Lasithiotakis, Michail-Angelos and Makridis, John and Moraitis, Panagiotis and Panteri, Aspasia and Plevraki, Maria et al. [30]

Table 14

Category: Hypermedia-based approaches.

Paper	Title	Author
HYPERM-1	All About That-A URI Profiling Tool for monitoring and preserving Linked Data	Vesse, Robert and Hall, Wendy and Carr, Les [22]
HYPERM-2	Preserving Linked Data on the Semantic Web by the application of Link Integrity techniques from Hypermedia	Vesse, Robert and Hall, Wendy and Carr, Les [31]
HYPERM-3	Link Integrity for the Semantic Web	Vesse, Robert [32]
HYPERM-4	Linking the (un)linked Data Through Backlinks	Stefanidakis, Michalis and Papadakis, Ioannis [33]

map and apply existing tools to interconnect ontological datasets, based on their URIs and underlying ontologies. A total of six link building tools - proposed in other investigations - is used to optimize the achieved results. The authors compared the level of automation, domain specificity (some of them tend to work better in certain domains) and types of similarity techniques explored in each of the used tools.

Silk [1] refers to a framework responsible for keeping alive links between two active datasets, as both evolve. Silk generates links between two datasets, evaluates them and track future links that have to be created based on the changes on these datasets.

The link discovery module of Silk [1] was enhanced and new links were created by the framework using genetic programming [35]. The genetic programming applied in linkage rules chooses the candidates of links based on a fitness measure. The authors experimented over geographic datasets, interlinking DBpedia and LinkedGeoData. They also tested the algorithm in complex linkage rules over drugs' data in DBpedia and Drugbank. The complex linkage rules were tested to match two drugs based not only on their names, but on their synonyms and international identifiers. The evaluations and the algorithm required a set of reference links as an entry point [35]. The study concluded that genetic algorithms can be used as a tool for link discovery and management, since they outperform results obtained by other approaches, such as SVM (Support Vector Machine). For example, MARLIN (Multiply Adaptive Record Linkage with Induction) framework [36] uses SVM to obtain the degree of similarity between records in databases, but without focusing on LOD or Semantic Web datasets.

PARIS (Probabilistic Alignment of Relations, Instances, and Schema) [37] is a framework used to align two ontologies [38]. The difference from the other frameworks is that PARIS creates links between instances and mappings between concepts at the ontology level. Instead of using SVM or genetic algorithms, it uses a probabilistic model to map matching instances in two distinct ontologies. The framework requires no training data, differently from SVM and genetic algorithms. PARIS is not suited to deal with huge structural differences, such entities that are treated as string in one ontology and as resources in another ontology [38]. The use of probabilistic model to discover new links and the avoidance of training data can be inspirational to frameworks that deal with maintenance action, given that it is solely applied to creation new links

The last approach worth mentioning is COLIBRI [37]. It is a link discovery tool that uses unsupervised machine learning algorithms to find resources candidate to be the object of the link. To the best of our knowledge, this is the only tool that discovers links in multiple datasets at the same time. Differently from the other approaches, it does not focus only on *owl:sameAs* links [37].

Table 15

Category: Link Management Mechanisms.

Paper	Title	Author(s) and Reference
LINKMA-1	Discovering and Maintaining Links on the Web of Data	Volz, Julius [1]
LINKMA-2	MeLinDa - An Interlinking Framework for the Web of Data	Scharffe, François and Euzenat, Jérôme [34]
LINKMA-3	Learning Linkage Rules using Genetic Programming	Isele, Robert and Bizer, Christian [35]
LINKMA-4	PARIS: Probabilistic Alignment of Relations, Instances, and Schema	Suchanek, Fabian M and Abiteboul, Serge and Senellart, Pierre [38]
LINKMA-5	Unsupervised Link Discovery Through Knowledge Base Repair	Ngomo, Axel-Cyrille Ngonga and Sherif, Mohamed Ahmed and Lyko, Klaus [37]

5.7. Hybrid solutions

This category includes contributions that present more than one of the characteristics defined in this survey to address link maintenance. This concerns, for example, user notification, versioning of changes, metadata storage or the use of auxiliary ontologies. Table 16 summarizes the articles in the hybrid solutions category.

One of the first efforts to avoid broken links in the Linked Data context is the act of sending notifications to the maintainer, administrator of the ontology or the responsible for editing the current node, triple or links. These notifications are triggered by event detection mechanisms, representing what, when and why the resource has changed.

DSNotify [23] refers to an event detection framework, which aims to preserve link integrity with the aid of notifications. Based on detected events, this framework notifies the maintainer of the dataset about changes in its structure. Its core is organised by three modules: the first is responsible for storing the content of the items and their metadata reached by a certain URI in a vector structure, named Feature Vector; the second is responsible for storing the vectors in three distinct types of index: the item index (when a new resource is created in the dataset), the removed item index (when *DSNotify* computes that a certain resource is unreachable) and the archived index (when there is a new location to the resource); and, the third is responsible for notifying the application that something in the dataset has changed [23].

DSNotify can fix structurally broken links with the need of human review and analysis. The authors discussed strategies that could be employed to maintain link integrity, such as deleting all the statements that contain the target resource that was deleted. One of the limitations regarding the methodology used in *DSNotify* is the periodicity to check if there are changes in the dataset, which can be an obstacle to the adoption of the framework by real-time applications that cannot wait and require knowing about the changes when they occur. In addition, the approach may have scalability issues, if the number of notifications sent increases [15].

The Delta LD framework [20] aims to classify the changes detected between versions of the same dataset. The framework organizes the changes based on structural and semantic approaches for storing/representing them in a triple-centric way. The proposed framework can fix broken links caused by the updating of the resources, properties and triples of a dataset. Firstly, it detects and classifies changes in “removed”, “moved” and “renewed” groups of changes. Then, based on these groups, a repairing action is performed using SPARQL to deal with structurally broken links found using SPARQL templates. If the change is categorized as a “removed”, then the broken link is deleted. If there was a “moved” or “renewed”, the framework deletes the old link and adds a new link using the URI of the updated resource. By evaluating the framework with the use of Delta LD the authors showed evidence that the precision and recall obtained better results on detecting and classifying both types of changes, compared with other solutions in the literature [20].

The study conducted by Liu and Li [39] was based on the DSNotify framework [23]. Their proposal was the use of metadata to detect and notify real-time changes in the dataset, without need to scan the entire dataset periodically. Their investigation defined an automatic method for synchronisation and propagation of changes, without the need to periodically sweep the dataset looking for changes. Another difference is related to the way that notifications are communicated: if something changes in the dataset, the user receives the event only in the next time (s)he views the resources that have undergone the change. This decision was taken to avoid an overwhelming number of changes notified to the user. In this sense, the authors implemented a tool using this approach, which avoids monitoring all linked data in the dataset and update the consumers in a more controlled way.

An alternative approach was proposed by Meehan *et al.* [40], named as the SUMMR methodology. Their work based on metadata explores SPARQL query templates to store and select mappings (inserted using metadata) that may have become invalid. In such work, the term "mapping" refers to external links, such as the 27 million links connecting DBpedia to 36 external datasets, in one of the DBpedia releases. SUMMR is concerned about both reuse and repair of mappings between datasets.

Roussakis *et al.* [7] proposed a framework for joining links and detecting complex changes by storing detection queries of change events in order to retrieve events occurred in the past. The work reports "high-level changes" as the aggregation of simple changes (additions or exclusion of triples) resulting in human-understandable changes. The authors proposed providing a way to easily navigate between two different dataset releases seeking for changes, known as cross-snapshot query. The authors argue this solution facilitates the access to the changes in a queryable way.

5.8. Surveys

This category comprises the presentation of retrieved surveys concerning research issues relevant for link maintenance. Table 17 summarizes the articles in the survey category.

Nentwig *et al.* [41] conducted a review of ten studies including tools and frameworks for discovering candidates for possible links between datasets. This work summarized relevant aspects of having an interconnected dataset in LOD. It highlighted that 44% of LOD datasets do not create links to other datasets, which does not follow the LOD best practices. The authors proceed on the evaluation of existing tools by comparing their effectiveness (assessment regarding the generation of high-quality links), their degree of automation, diversity of links, and computation efficiency other aspects.

The survey carried out by Nentwig *et al.* [41] concluded that most of the techniques used for link discovery rely basically on the analysis of the resource itself, not on the neighbourhood or the ontology context [42]; the adoption of genetic programming proved to be an important strategy; and the use of existing links and background knowledge to create new links is not widely adopted.

Dos Reis *et al.* [43] conducted a literature survey on mapping maintenance techniques for KOS (Knowledge Organization Systems). KOS comprise ontologies, thesauri and other structures to represent knowledge. Ontology mappings connect concepts that are semantically related between distinct ontologies. The semantic update of a given concept can invalidate an existing mapping. The authors proposed a definition of the mapping maintenance task and explained the importance of maintaining mappings up to date. Then, they categorized existing studies into four categories: mapping revision, calculation, adaptation, and representation.

The mapping revision category concerned identification and fixing of invalid mappings. The mapping calculation category addressed a fully or partial re-connection of mappings. The mapping adaptation category described a collection of strategies to re-organize affected mappings such as: composition of mappings, merging various mappings and transforming them into a single mapping; rewriting, to store the mappings in databases schemas; and synchronisation, to generate and maintain mappings between different types of KOS. The last category included studies concerning the construction of user interfaces for handling mapping maintenance.

Dos Reis *et al.* [43] described challenging aspects related to the mapping maintenance task. Most of the existing techniques (at the time of the survey) relied on logical inferences, which only benefit KOS possessing "high level of formalization", such as ontologies [43]. The authors stated that the proposal of using partial re-calculation of mappings tends to reduce time cost. However, this is preventive for huge KOS.

Groß *et al.* [44] surveyed approaches to ontology and mapping evolution in the biomedical area. This area contains several huge ontologies (e.g, SNOMEDCT [45]), that are updated constantly and with overlapping information. The non-static behaviour of biomedical ontologies result in the creation of new mappings, as long as existing ones can be invalidated. The survey identified key aspects of mapping maintenance and how the existing literature addresses them.

Groß *et al.* [44] described techniques that help in the task of mapping adaptation, including: detection, visualisation and the prediction of changes at the concept level (ontologies). In order to compare the adaptation of ontology-based mappings, the authors enumerated a series of requirements to compare five novel algorithms and frameworks. For example, evaluation of the mappings' quality, ontology size, user interaction

Table 16
Category: Hybrid solutions.

Paper ID	Title	Author(s) and Reference
HYBRID-1	DSNotify - Detecting and Fixing Broken Links in Linked Data Sets	Haslhofer, Bernhard and Popitsch, Niko [23]
HYBRID-2	DSNotify - A Solution for Event Detection and Link Maintenance in dynamic datasets	Popitsch, Niko and Haslhofer, Bernhard [15]
HYBRID-3	Delta LD - A Change Detection Approach for Linked Datasets	Singh, Anuj and Brennan, Rob and o'Sullivan, Declan [20]
HYBRID-4	Using Metadata to Maintain Link Integrity for Linked Data	Liu, Fangfang and Li, Xiaojing [39]
HYBRID-5	Validating Interlinks between Linked Data Datasets with the SUMMR Methodology	Meehan, Alan and Kontokostas, Dimitris and Freudenberg, Markus and Brennan, Rob and o'Sullivan, Declan [40]
HYBRID-6	A Flexible Framework for Understanding the Dynamics of Evolving RDF Datasets	Roussakis, Yannis, Chrysakis, Ioannis, Stefanidis, Kostas, Flouris, Giorgos Stavrakas, Yannis [7]

and calculus of semantic mappings. Two studied cases use more complex diff operations to identify changes in mappings, such as merge operations. Non-equality mappings, such as "part-of" and "is involved in", were also compared. Most of the studied cases only use equality mappings. None of the cases provides visualisation to aid ontology maintainers to check the life cycle of mappings. Scalability issues were not compared among the studies investigated [44].

The survey reported by Groß *et al.* [44] indicated future directions in mapping evolution research. Focusing on semantic mappings is challenging according to them. Each domain-specific mapping has its own semantics and, maintaining them at different ontology versions demands further investigations. The use of machine learning to predict and recommend mapping candidates is also a challenge. The maintenance of mappings in multilingual ontology environments requires research to fulfil the requirements of developing a robust mapping evolution algorithm [44].

6. Discussion and Research Challenges

The explicit connection between resources belonging to distinct datasets plays a central role in the interconnection between repositories in the LOD. Linked Data sources are subject to changes, since data regarding new entities are added as well as outdated data are modified or removed. The update of data repositories leads to the link maintenance problem. The maintenance of links among the linked data cloud is hard and expensive.

This systematic literature review is relevant to understand the frontiers of the link integrity and link maintenance problem. Our investigation aimed to gather and organise the existing literature on the problem to provide a complete survey to pave the way for further research and guide future researches to overcome the drawbacks and limitations of recent studies. Our literature review indicated that there is no evidence of a survey related to the state of the art in broken link diagnosis or even a method that can accurately and automatically maintain the links in a unbroken state or, given the broken state, fix them.

Our literature analysis indicated that state-of-the-art techniques do not fully address the complexity of the chain of events involved between the period of the occurrence of a change and the automatic fixing. Some approaches explore the notification solution, which might have their usability reduced due to the huge amount of triples stored. The manual checking of links to certify real broken links - *i.e.,* whether it is a false positive or not - becomes impossible with the existing amount of data.

The significant number of changes affecting datasets might considered in link maintenance solutions. We understand that although the use of notifications to solve the broken link problem presents its limitations, the combination with other methods considering the way the datasets were changed can be relevant to advance the problem resolution and decrease the human burden.

Copying or being inspired by existing techniques from the traditional Web based on documents can benefit the handling of link maintenance in the Semantic Web data. However, the context and impact of a dead link in linked data is more problematic. In traditional Web, when the final user hits a dead link, (s)he can go back to the last page and search for another link that points to the same resource or to a similar one. In Semantic Web, applications seeking for information need to deal with dead links differently. Existing investigations point out the use of backlinks stored in a database

Table 17

Category: Survey papers.

Paper ID	Title	Author(s) and Reference
SURVEY-1	A Survey of Current Link Discovery Frameworks	Nentwig, Markus and Hartung, Michael and Ngonga, Axel-Cyrille and Rahm, Erhard [41]
SURVEY-2	State-of-the-art on Mapping Maintenance and Challenges Towards a Fully Automatic Approach	dos Reis, Julio Cesar and Pruski, Cedric and Reynaud-Delaitre, Chantal [43]
SURVEY-3	Evolution of Biomedical Ontologies and Mappings: Overview of Recent Approaches	Groß, Anika and Pruski, Cédric and Rahm, Erhard [44]

to know where the dead link comes from, and where it is heading. In our understanding, it is not enough to provide dataset link-error proof.

Our literature review emphasized the use of versioning systems, which create logs and show to the user when something changed in the dataset. The datasets that obtain most benefits with versioning are those with evolutionary characteristics, such as biologic datasets. The detection and understanding of changes in datasets, such as modification operations in a property and/or a statement, is valuable to the link maintenance problem. We advocate that detected changes must be explicitly used to help updating links. Thus, links in an invalid state can be modified or rollback the modifications to a valid state.

Our results showed that most of the existing approaches are suited to avoid creating broken links as well as to detect broken links. However, they still lack techniques to adequately fix them. The use of dataset changes could play a more prominent role in this task. Indeed, RDF versioning considers both atomic and compound changes. We argue that both types of changes should be further explored to support the discovery of link integrity issues and to provide information to support their update according to changes underwent in internal triples. Nevertheless, versioning with atomic and compound changes in both TBox (specification of concepts in a domain) and ABox (specification of individuals of concepts) are often used solely to document the ontology evolution, not to help fixing links. In this sense, approaches combining versioning and backlink techniques can be a great information source for link maintenance.

6.1. *Comparative Analysis*

This subsection presents a comparative analysis among the retrieved studies in each category. Our comparison is based on characteristics summarised in Table 18 as well as on the analysis of implemented tools.

Change detection category: In this category, we analysed 2 studies with different approaches for creating snapshots of a given dataset. In the case of broken link detection, the maintainer is able to rollback to an older version and to fix the issue. The VERSIO-1 [10] (Powl) study was one of the first efforts to adapt change versioning strategies to the Semantic Web context. The strategy used for versioning was similar to the one used in the investigations classified in the high-level modifications category, such as HYBRID-6 [7], HIGHLE-1 [26] and HIGHLE-2 [27]. The versions are stored in atomic/simple changes, which can be versioned and investigated as compound/complex changes. The change tracking operations performed in the dataset can be used to undo applied actions, such as to fix a link that became broken after a change. VERSIO-1 [10] (Powl) presented a change conflict detection feature, in case two conflicting operations are performed in the same resource.

VERSIO-2 [12] (MRSimDiff) concerned performance issues to produce and store dataset versions. It uses the MapReduce framework [46] for distributed computing. In their approach, RDF triples are stored in an unordered way while not affecting their semantics to enable reducing the size of the stored versions.

Metadata storage category: In the two studies analysed, the first one METADA-1 [25] focused on the importance of using metadata for transparency in open data. This work analysed why poorly-created metadata is still present in LOD architectures. The study discussed the quality of the generated metadata, which is strictly related to the quality of links and maintenance tasks. The METADA-2 [21] approach explored metadata to discover semantically broken links.

High-level modifications category: This category focused on investigations dealing with high-level modifications in ontology evolution. According to the approach developed in HIGHLE-1 [26], change computation refers to the process of identifying changes given two releases of a dataset. This study proposed a language to handle how to avoid losing semantics and dependencies in versioning.

Both studies HIGHLE-2 [27] and HIGHLE-3 [28] served as a basis to HIGHLE-1 [26] because they shared a similar hypothesis, *i.e.*, develop a framework containing a specific language to produce concise deltas between dataset versions in a manner that changes are easily readable by humans.

Ontology-driven change representation category: This category is composed of various techniques to handle links using ontology-based solutions. ONTOLO-1 [24] focused on understating what happened to changed links by using two temporary ontologies. The first ontology contains all triples that have the observed entity as object; and the second ontology contains all triples presenting the observed link as subject. The authors argued that a modification at a resource in most cases preserves its structure, so the temporary ontologies can be used to find possible candidates to the broken structure. This was the only study in this category that claimed to be capable of detecting and fixing structurally broken links. ONTOLO-2 [29] explored an ontology to store the addition and exclusion of triples to represent changes in a SPARQL queryable way. ONTOLO-3 [30] (EvoRDF) explored ontologies to represent changes and store relevant data regarding the evolution of datasets.

Hypermedia-based solutions category: This category includes investigations inspired by methods in the traditional Web. These findings in this category indicate that there is no evidence of a global solution able to handle inconsistent links in the Web, only solutions for small scale networks. HYPERM-1 [22] (All About That) developed a framework that creates profiles of the user's URIs of interest. A graph with the associations and links of this URI is created. The graph was used to track changes in key URIs components, such as new resources or removed links.

HYPERM-2 [31] followed a similar proposal as that of HYPERM-1 [22] (All About That): data recovery instead of data monitoring, *i.e.*, preventing errors in case of some failure, (*e.g.*, a broken link). The study in HYPERM-2 [31] argued that monitoring links does not perform and scale well in the Web. HYPERM-2 [31] was an expanded version of HYPERM-1 [22].

HYPERM-4 [33] investigated a different approach to track links: the backlinks. This was inspired by traditional Web and proposed to store links pointing to and being pointed by a given resource, which might facilitate the task of broken link tracking.

Link management category: This category presents frameworks that somehow deal with the creation of links among datasets. They track and discover datasets and resources within the datasets that can be linked. The first tool LINKMA-1 [1] (Silk) proposed the creation of links based on user defined characteristics that the resources must have. The user must define which semantic measures should be used, which resources should be concatenated, as well as if codes or labels should be used in the comparison procedure. The quality of the generated links heavily depends on the user-defined settings. LINKMA-2 [34] (MeLinda) represented by the framework MeLinda follows a similar research path and extended the work of LINKMA-1 [1] (Silk): user-defined rules for semantic equivalence search among resources in distinct datasets.

Differently from LINKMA-1 [1] (Silk) and LINKMA-2 [34] (MeLinda), the other studies in this category do not rely on user-defined parameters. LINKMA-3 [35] used genetic programming and relies on training data to perform link discovery tasks. LINKMA-4 [38] (PARIS), on the other hand, does not require training data and uses probabilistic models to link resources. LINKMA-5 [37] (COLIBRI) is an approach using machine learning techniques to link resources in n datasets, where $n >= 2$.

Hybrid solutions category: This category includes investigations combining distinct strategies towards solving link maintenance issues. HYBRID-1 [23] (DSNotify), notifies users about operations in the dataset, such as creation and removal of triples. It can, for example, notify the maintainer of a dataset when the target of a link becomes unreachable. It affords finding possible candidates to be a damaged link. To this end, it seeks for target resources possessing parts of the URIs similar to the old broken resource. In summary, HYBRID-1 [23] (DSNotify) and HYBRID-2 [15] notify users about changes; and, in case of broken link detection, they fix the structurally broken ones automatically. The fixing action is performed after a resource that has parts of the old and broken resource is located.

HYBRID-3 [20] (Delta-LD) uses the broken link versioning and detection strategy to fix structurally broken links automatically. HYBRID-4 [39] argues that the solutions using notification, such as the one described in HYBRID-1 [23] and HYBRID-2 [15], are not efficient because it is unfeasible to report all cases and periodic monitoring may miss some of them. HYBRID-4 [39] advocates the use of notifications and metadata together for broken link detection task without fixing them.

HYBRID-5 [40] (SUMMR) investigated how enriched metadata can be useful for link maintenance. Differently from HYBRID-4 [39], HYBRID-5 uses SPARQL queries. HYBRID-5 [40] (SUMMR) stores metadata about changes and detects and fixes structurally broken links in an assisted way.

HYBRID-6 [7] describes a framework for both detection and analysis of changes in LOD datasets. It explored advanced versioning to detect broken links. To this end, HYBRID-6 combines querying versioned databases and change ontology to deal with the manipulation, organization and information on triple changes.

We defined a set of tasks related to link maintenance, which appeared in the analysed literature. The following tasks were used to analyse existing proposals in the literature surveyed in this work:

i **Change detection:** It refers to the identification of changes in RDF datasets in a synchronous or asynchronous way.

ii **User notification:** Task explored mainly to notify the responsible of the repository via messages regarding events related to link integrity;

iii **Versioning:** Deals with techniques for the versioning of LOD datasets;

iv **Metadata storage:** Attachment of metadata for the description of dataset resources with the aim of supporting link maintenance;

v **Broken link detection:** Methods for the recognition of invalid links in interconnected datasets;

vi **Fix of structurally broken links**: Proposal of algorithms to suggest the correction of structurally broken links supported by human interventions;

vii **Automatic fix of structurally broken links**: Fully automatic techniques to fix broken links without the need of human intervention;

viii **Fix of semantically broken links**: Proposal of algorithms to suggest the correction of semantically broken links supported by human interventions;

ix **Automatic fix of semantically broken links**: The algorithm does not need any human intervention to fix broken links considering its semantic aspects.

Tasks (i) to (v) represent the phases initially carried out to handle link maintenance, such as detecting a broken link. Tasks (vi) to (ix) emphasise how to fix the links.

Table 18 presents the analysed papers studied in this survey and shows which tasks related to link maintenance are supported/solved by the explored studies. The three survey articles summarized in Table 17 were not mentioned in Table 18, since we only compared characteristics of individual articles. Each column of the table refers to a task.

Results in Table 18 indicate that existing studies still do not support a fully automatic framework for handling link maintenance tasks. The human intervention is still relevant for the existing proposals. Whereas change detection has been widely studied the way of fixing semantically broken links is uncovered by the investigations:

- 20 of 25 studies present the ability to know that something has changed and they deal with this in their own way: 3 send messages to the user; 9 store the changes in versions; 3 attach metadata to it; and 8 are able to indicate that a link is correlated to the change;
- Most investigations deal with the first tasks (i to iv), which indicates a trend in the literature of detecting instead of fixing;
- 5 studies were able to detect broken links and correct them manually, or suggest corrections to the user; 3 investigations described techniques to enable fixing links automatically. However, they only apply to structurally broken links;
- The lack of solutions that automatically fix semantically broken links demonstrates that the link maintenance topic in linked data must be further investigated.

Most of the found solutions focus on handling the first part of the process to prevent broken links: *i.e.*, the discovery. This part, which consists of data preservation and monitoring actions, such as change versioning and notification of changes, is only halfway of a full link maintenance process. To fulfil the process of maintaining a link up to date with no semantic or structural errors we should further advance in how to address the fixing part of the process.

6.2. Answers to the Research Questions

The conduct of our literature survey enables us to answer the defined research questions presented in Table 2. The posed research questions were answered based on the knowledge acquired from the 28 scientific articles selected. We considered successful our goal to uncover the unsolved open research questions and potential solutions presented in the literature on link integrity in LOD.

RQ-01: What are the benefits of having an RDF dataset with no or very few broken links?

The benefits of having an RDF dataset with no or very few broken links include increased trust in dataset consistency. Unbroken links, or links that are in an in-

Table 18
Analysis of surveyed papers on link maintenance tasks. The lines refer to each study analysed in this survey. Columns represent the distinct defined tasks related to link maintenance. The tasks are: (i) change detection; (ii) user notification; (iii) versioning; (iv) metadata storage; (v) broken link detection; (vi) fix of structurally broken links; (vii) automatic fix of structurally broken links; (viii) fix of semantically broken links; (ix) automatic fix of semantically broken links. The X sign indicates that such study (work) addressed the defined task somehow.

	i	**ii**	**iii**	**iv**	**v**	**vi**	**vii**	**viii**	**ix**
VERSIO-1 [10]	X		X						
VERSIO-2 [12]	X		X						
METADA-1 [25]				X					
METADA-2 [21]	X				X				
HIGHLE-1 [26]	X		X						
HIGHLE-2 [27]	X		X						
HIGHLE-3 [28]	X		X						
ONTOLO-1 [24]	X				X	X			
ONTOLO-2 [29]	X								
ONTOLO-3 [30]	X								
HYPERM-1 [22]	X								
HYPERM-2 [31]									
HYPERM-3 [32]	X		X						
HYPERM-4 [33]	X		X						
LINKMA-1 [1]	X								
LINKMA-2 [34]	X								
LINKMA-3 [35]									
LINKMA-4 [38]									
LINKMA-5 [37]									
HYBRID-1 [23]	X	X			X	X	X		
HYBRID-2 [15]	X	X			X	X	X		
HYBRID-3 [20]	X		X		X	X	X		
HYBRID-4 [39]	X	X		X	X				
HYBRID-5 [40]	X			X	X	X			
HYBRID-6 [7]	X		X		X				
TOTAL: 25	20	3	9	3	8	5	3	0	0

tegrity state, deliver the data that was intended by the link creator. A highly connected dataset cloud is beneficial for both maintainers and final users. Maintainers can keep focus on the core purpose of the dataset without worrying about lacking important information about related domains.

For example, the Geonames dataset has information about coordinates of a given country. DBpedia presents information about government hierarchy in a country. The Geonames maintainers do not have to worry about government information in their dataset, as long as it has a link to the same country in DBpedia, which contains information about government hierarchy. The absence of this kind of link would separate the datasets, turning them into several islands of knowledge. Without adequate links or with a large number of broken links, final users would not be able to reach a myriad of interconnected and rich information.

RQ-02: What are the types of broken links?
Surveyed studies consider broken links as structurally and semantically broken links. We found existing studies suited to address structurally broken links as they are easier to detect and be fixed by algorithms. For example, HYBRID-3 [20] and HYBRID-5 [40] (cf. Table 18) presented how they deal with the issue of fixing structurally broken links. On the other hand, semantically broken links are harder to detect and fix. To the best of our knowledge, our survey did not detect solutions able to automatically fix these links.

RQ-03: What are the existing proposals for solving broken links outside the Linked Open Data community?
Our literature analysis found that the reported solutions for broken links outside the Semantic Web field are mainly based on hypermedia-based techniques [32], such as the use of backlinks [33]. Nevertheless,

backlinks are only used to detect the presence of broken links, *i.e.*, they are not used to fix them.

Another strategy used outside the LOD community is to profile all possible changes and store them in a database, as stated by HYPERM-1 [22] (cf. Table 18). Although this work concerned preservation of data and avoidance of broken links, it did not deal with their correction.

RQ-04: What are the existing solutions for the link integrity and link maintenance problem in the Linked Open Data community?

The most common solution used in LOD datasets is to ignore broken links and make them the responsibility of the application [23], which is not the best alternative.We organized the analyzed existing solutions into two groups: the first one includes solutions related to data preservation techniques, which is concerned with detection of possible broken links; and the second one includes solutions related to maintenance techniques, which is concerned with the fixing of broken links.

In the detection of broken links (first group), we found the following techniques:

- Adopting a sub ontology (ONTOLO-1) [24] for change representation;
- Visualising high-level changes (HIGHLE-2) [27];
- Notifying Changes (HYBRID-2) [23].
- Versioning deltas of computed changes (HYBRID-3) [20].

In the maintenance of broken links (second group), we found the following techniques:

- Relying on Not Found HTTP response errors via active monitoring and changes notifications (HYBRID-1) [23] (HYBRID-2) [15];
- Using SPARQL templates [20] (HYBRID-3), [40] (HYBRID-5).

RQ-05: Are there fully automated approaches to maintain links up-to-date?

Our results indicate that there is no evidence of a fully automated framework that can theoretically or in practice maintain all kinds of links up-to-date. All kinds of links refer to RQ-02, which categorises broken links as semantic and structural. We found approaches to fixing structural broken links using human intervention (cf. column vi of Table 18) as well as approaches without need of human intervention (cf. column vii of Table 18). However, none of them is able to fix semantically broken links, with or without human intervention (cf. columns viii and ix of Table 18).

We found approaches suited to discovering links. The frameworks that fall in this category present valuable ideas on how to seek for candidate links among datasets, which includes (un)supervised learning algorithms, genetic programming and probabilistic methods. These strategies can be used, for example, to identify broken links in future work, instead of just creating reliable new links automatically.

6.3. *Open Research Challenges*

Our thorough literature review and experience in the link maintenance problem helped to detect and categorise major research challenges that deserve further investigations. In the following section, we describe open problems closely related to link integrity in linked datasets:

Empirical analyses to understand link maintenance. Our literature review showed a lack of empirical investigations regarding the evolution of LOD datasets. Further studies could be conducted to provide findings regarding how links evolve in the context of LOD. Experiments should be conducted considering several versions of real-world RDF repositories taking their links into account to analyse the way change operations in RDF repositories correlate to modifications observed in links. This could show key factors of link evolution according to RDF change operations. This should enable understanding and predicting the consequences of changes performed in RDF datasets over links. Initial results were obtained in our previous work on this research challenge [17].

Detection of outdated links based on detected changes. The literature on broken links and link integrity is not concerned with the impact of changes in the coherence of links. In this sense, techniques must be studied to detect which types of RDF changes in internal triples turn the links invalid. This must consider more than atomic changes and rely on composed changes. Domain-related changes and the modification of the meaning in the resources should be considered in the technique. In this sense, we should be able to detect how triple changes unveil semantically broken links.

Spam and overflow of notifications. One of the challenges is to deal with the overflow of spam and error messages to the administrators due to broken links. Due to the high number of notifications and size of datasets, it is essential to devise strategies related to periodical notification. Few frameworks were developed to address this challenge. For instance, DSNotify [23] sends notification periodically whereas other frameworks handle overflow differently, such as the work proposed by Liu and Li [39], in which the solution uses metadata to notify the user about the changes in the resource (s)he is viewing instead of the entire dataset changes. We understand that there is still room for improvements in the issues about notifications. This involves the management and demand of messages about changes on real-time applications, which requires fast response of the dataset and includes the detection and fixing of semantically broken links.

Lack and Fault Tolerance of Notifications. In contrast to sending too many notifications, there is the problem of not detecting all the changes in the dataset leading to a problem of low coverage. This results in an agglomeration of broken links that was not detected and informed to the administrator.

Size of Deltas. Usually, the created deltas are huge from one dataset to another, when several changes are detected. The challenge is how to reduce the size of the delta [12]. The production of small deltas is particularly difficult in Linked Data mainly because of RDF triples characteristics. It is important to mention that the order of the triples does not matter: so, if a triple is moved to the end of the file that stores the triples, this triple cannot be computed.

Broken/Missing Links Fixing. In our view, handling broken links and link integrity is part of the link maintenance problem as the key focus of our survey. In our understanding, in order to address the way of fixing links, it is necessary to identify changes in the RDF repositories, detect outdated links, to then apply techniques to correctly fix them. The detection of structurally and semantically broken links is an important and challenging task. Structurally broken links are widely covered by the literature. Semantic ones are harder to detect and correct based on the fact that semantics are not always consensual and highly domain dependent.

Link prediction in the context of maintenance. In our understanding, link discovery techniques can be useful for the link maintenance problem. For example, it can help in computing possible candidate links to handle the broken links in linked RDF datasets. However, finding how to combine change detection with link prediction to handle link integrity constitutes a challenge. Link maintenance could benefit from existing approaches to link prediction as part of its process. Nevertheless, it requires further research to understand how these approaches can be combined to provide more automatic results for link maintenance.

7. Conclusion

The growing number of semantically-enhanced data published and consumed in RDF repositories in the Web confirms the real potentiality of the global data space proposed by the Semantic Web vision. Links between distinct resources in different datasets play a key role to interconnect RDF repositories. RDF statements defining real-world resources tends to change. These operations can affect established links turning them broken. This hampers data linkage integrity over time.

This article contributed with a systematic literature survey concerning link maintenance in LOD. We presented, discussed and compared existing approaches for tasks related to the maintenance of links. Our results indicated the need for improvements in this research field. Our results using a controlled process and a formal bibliographic research should benefit the research community, listing topics of interest and challenges demanding more investigation and knowledge deepening towards a fully automatic approach for link maintenance. We found that most of the existing investigations focus on the broken link detection phase whereas the fixing phase still involves several open research challenges. The findings obtained in this survey can be valuable for inspiring ideas and design solutions for novel software tools suited to deal with the full link maintenance process, including discovery and fixing of both structurally and semantically broken links.

Future work involves addressing the key research challenges. A new study could experimentally compare existing software tools shown in this survey by considering effectiveness measures such as precision, recall and accuracy, as this was not the focus of our survey. In addition, we plan to conduct extensive experiments to understand the evolution of links in the LOD to correlate changes in the semantic definition of data resources with modifications observed in predefined links. This must pave the way to the definition of an automatic and precise solution for link maintenance mechanism suited to deal with LOD dynamics and link integrity.

Appendix

Table 19 presents the number of papers retrieved by the queries run in each of the scientific databases. The

Table 19

Retrieved and selected articles by queries and databases

Query	**ACM**	**IEEE**	**Elsevier**	**Springer**	**Others**
Q-01	0(1)	0	1(1)	0	3(18)
Q-02	0(1)	1(1)	1(1)	0	0
Q-03	0	0(1)	1(10)	1(540)	8(100)
Q-04	0	0	0(5)	0	0
Q-05	0	0	0	0	0
Q-06	1(2)	0	1(17)	1(700)	4(100)
Q-07	0	0	0	0	0
Q-08	0	0	1(7)	2(117)	8(100)
Q-09	0	0	0	0	1(100)
Q-10	0	0	0	0	0
Q-11	0(1)	0	0(2)	0	1(100)
Q-12	0	0	0	0	0
Q-13	0(5)	0	0(106)	0	0
Q-14	0(35)	0	0(538)	0	0
Q-15	0	0	0(21)	0	0(36)
Q-16	0	0	0	0	0(5)
Q-17	0(1191)	0	0(3)	0	1(30)
Q-18	0	0	0(3)	0	5(34)
Q-19	0	0	0	0	5(15)
Q-20	0	0	0	0	1(34)
Q-21	0(4)	0	0(2)	0	0(51)
Q-22	0	0	1(14)	0	0
Q-23	1(4)	0(1)	1(55)	0	0
Q-24	0(3)	0(6)	1(47)	0	0
Q-25	0	0(5)	0(192)	0	0
TOTAL	**2**	**1**	**8**	**4**	**37**

first column shows the id of the query (*cf.* Table 3). The first number in each row represents the number of used papers from the query used in this survey, whereas the second number (between parentheses) represents the number of articles returned by the database search engine. For example, *Q-06* identifies the query *"broken links"* + *"web of data"*, which retrieved seventeen results in Elsevier Science Direct database; one paper from it was used in this survey. The queries run resulted in the retrieval of fifty-two papers. Some of them were repeated among the databases. The sum of papers retrieved by the queries and considered relevant in this study totals twenty-two. They cover topics related to link maintenance tasks.

Acknowledgements

This work was financially supported by the São Paulo Research Foundation (FAPESP) (list of grants: #2013/08293-7, #2017/02325-5, #2018/08082-0 and #2018/14199-7)[11].

References

[1] J. Volz, C. Bizer, M. Gaedke and G. Kobilarov, Discovering and Maintaining Links on the Web of Data, in: *The Semantic Web - ISWC 2009*, A. Bernstein, D.R. Karger, T. Heath, L. Feigenbaum, D. Maynard, E. Motta and K. Thirunarayan, eds, Springer Berlin Heidelberg, Berlin, Heidelberg, 2009, pp. 650–665, https://doi.org/10.1007/978-3-642-04930-9_41. ISBN 978-3-642-04930-9.

[2] T.R. Gruber, Toward principles for the design of ontologies used for knowledge sharing, *International Journal of Human-Computer Studies* **43**(5) (1995), 907–928, ISSN 1071-5819. doi:https://doi.org/10.1006/ijhc.1995.1081.

[3] C. Bizer, T. Heath and T. Berners-Lee, Linked Data - The Story So Far, *International Journal on Semantic Web and*

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

Information Systems **5**(3) (2009), 1–22, ISSN 1552-6283. doi:10.4018/jswis.2009081901.

[4] C. Bizer, T. Heath, D. Ayers and Y. Raimond, Interlinking open data on the web, in: *Demonstrations track of the 4th European Semantic Web Conference, Innsbruck, Austria*, ESWC 2007, 2007, pp. 802–815.

[5] T. Galani, Y. Stavrakas, G. Papastefanatos and G. Flouris, Supporting Complex Changes in RDF(S) Knowledge Bases, in: *Proceedings of the First Diachron Workshop on Managing the Evolution and Preservation of the Data Web co-located with 12th European Semantic Web Conference (ESWC 2015), Portorož, Slovenia, May 31, 2015.*, J. Debattista, M. d'Aquin and C. Lange, eds, CEUR-WS.org, 2015, pp. 28–33.

[6] R. Djedido and M.-A. Aufaure, Ontology evolution: state of the art and future directions., *Ontology Theory, Management and Design: Advanced Tools and Models* (2010), 179–207. doi:http://doi:10.4018/978-1-61520-859-3.ch008.

[7] Y. Roussakis, I. Chrysakis, K. Stefanidis, G. Flouris and Y. Stavrakas, A flexible framework for understanding the dynamics of evolving RDF datasets, in: *The Semantic Web - ISWC 2015 - 14th International Semantic Web Conference, Bethlehem, PA, USA, October 11-15, 2015, Proceedings, Part I*, M. Arenas, Ó. Corcho, E. Simperl, M. Strohmaier, M. d'Aquin, K. Srinivas, P. Groth, M. Dumontier, J. Heflin, K. Thirunarayan and S. Staab, eds, Lecture Notes in Computer Science, Vol. 9366, Springer, 2015, pp. 495–512. doi:https://doi.org/10.1007/978-3-319-25007-6_29.

[8] A. Morishima, A. Nakamizo, T. Iida, S. Sugimoto and H. Kitagawa, Bringing Your Dead Links Back to Life: A Comprehensive Approach and Lessons Learned, in: *Proceedings of the 20th ACM Conference on Hypertext and Hypermedia*, HT '09, Association for Computing Machinery, New York, NY, USA, 2009, pp. 15–24. ISBN 9781605584867. doi:10.1145/1557914.1557921.

[9] N.P. Popitsch and B. Haslhofer, DSNotify: Handling Broken Links in the Web of Data, in: *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, Association for Computing Machinery, New York, NY, USA, 2010, pp. 761–770. ISBN 9781605587998. doi:10.1145/1772690.1772768.

[10] S. Auer and H. Herre, A Versioning and Evolution Framework for RDF Knowledge Bases, in: *Proceedings of the 6th International Andrei Ershov Memorial Conference on Perspectives of Systems Informatics*, PSI'06, Springer-Verlag, Berlin, Heidelberg, 2006, pp. 55–69. ISBN 9783540708803. doi:10.1007/978-3-540-70881-0_8.

[11] M. Meimaris, G. Papastefanatos, C. Pateritsas, T. Galani and Y. Stavrakas, A Framework for Managing Evolving Information Resources on the Data Web, 2015. http://arxiv.org/abs/1504.06451.

[12] T. Lee, D.-H. Im and J. Won, Similarity-based Change Detection for RDF in MapReduce, *Procedia Computer Science* **91** (2016), 789–797, ISSN 1877-0509. doi:https://doi.org/10.1016/j.procs.2016.07.081.

[13] A. Jaffri, H. Glaser and I. Millard, Managing URI Synonymity to Enable Consistent Reference on the Semantic Web, in: *Proceedings of the 1st IRSW2008 International Workshop on Identity and Reference on the Semantic Web, Tenerife, Spain, June 2, 2008*, P. Bouquet, H. Halpin, H. Stoermer and G. Tummarello, eds, CEUR Workshop Proceedings, Vol. 422, CEUR-WS.org, 2008.

[14] A. Zaveri, A. Rula, A. Maurino, R. Pietrobon, J. Lehmann and S. Auer, Quality assessment for linked data: A survey, *Semantic Web* **7**(1) (2016), 63–93.

[15] N. Popitsch and B. Haslhofer, DSNotify - A Solution for Event Detection and Link Maintenance in Dynamic Datasets, *Web Semantics* **9**(3) (2011), 266–283, ISSN 1570-8268. doi:10.1016/j.websem.2011.05.002.

[16] J.C. Dos Reis, C. Pruski, M. Da Silveira and C. Reynaud-Delaître, Understanding Semantic Mapping Evolution by Observing Changes in Biomedical Ontologies, *Journal of Biomedical Informatics* **47** (2014), 71–82.

[17] A.G. Regino, J.K.R. Matsoui, J.C. dos Reis, R. Bonacin, A. Morshed and T. Sellis, 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*, A. Hasnain, V. Novácek, M. Dumontier and D. Rebholz-Schuhmann, eds, CEUR Workshop Proceedings, Vol. 2477, CEUR-WS.org, 2019, pp. 40–54.

[18] D. Budgen and P. Brereton, Performing Systematic Literature Reviews in Software Engineering, in *ICSE '06*, Association for Computing Machinery, New York, NY, USA, 2006, pp. 1051–1052. ISBN 1595933751. doi:10.1145/1134285.1134500.

[19] S. Faisal, K.M. Endris, S. Shekarpour, S. Auer and M.-E. Vidal, Co-evolution of RDF Datasets, *CoRR* **abs/1601.05270** (2016), 225–243, Springer.

[20] A. Singh, R. Brennan and D. O' Sullivan, DELTA-LD: A Change Detection Approach for Linked Datasets, in: *4th Workshop on Managing the Evolution and Preservation of the Data Web (MEPDaW) colocated with 15th European Semantic Web Conference (ESWC 2018)*, CEUR.ws, Crete, Greece, 2018, pp. 1–15.

[21] V.P. Kovilakath and S.D.M. Kumar, Semantic Broken Link Detection Using Structured Tagging Scheme, in *ICACCI '12*, J.A. Sabu M. Thampi El-Sayed El-Afry, ed., Association for Computing Machinery, New York, NY, USA, 2012, pp. 16–20. ISBN 9781450311960. doi:10.1145/2345396.2345400.

[22] R. Vesse, W. Hall and L. Carr, All about that-a uri profiling tool for monitoring and preserving linked data, in: *8th International Semantic Web Conference (ISWC), volume 5823 of Lecture Notes in Computer Science, Washington DC, USA*, 2009.

[23] B. Haslhofer and N. Popitsch, DSNotify - Detecting and Fixing Broken Links in Linked Data Sets, in: *20th International Workshop on Database and Expert Systems Application*, IEEE, Linz, Austria, 2009, pp. 89–93. doi:10.1109/DEXA.2009.13.

[24] M. Pourzaferani and M.A. Nematbakhsh, Repairing broken RDF links in the web of data, *International Journal of Web Engineering and Technology* **8**(4) (2013), 395–411. doi:10.1504/IJWET.2013.059106.

[25] A. Zuiderwijk, K. Jeffery and M. Janssen, The potential of metadata for linked open data and its value for users and publishers, *Journal of eDemocracy and Open Government* **4**(2) (2012), 222–244. doi:https://doi.org/10.29379/jedem.v4i2.138.

[26] T. Galani, G. Papastefanatos and Y. Stavrakas, A Language for Defining and Detecting Interrelated Complex Changes on RDF(S) Knowledge Bases, in: *Proceedings of the 18th International Conference on Enterprise Information Systems*, S. Hammoudi, L.A. Maciaszek, M. Missikoff, O. Camp and J. Cordeiro, eds, ICEIS 2016, SCITEPRESS - Science and

Technology Publications, Lda, Setubal, PRT, 2016, pp. 472–481. ISBN 9789897581878. doi:10.5220/0005833804720481.

[27] V. Papavassiliou, G. Flouris, I. Fundulaki, D. Kotzinos and V. Christophides, On detecting high-level changes in RDF/S KBs, in: *The Semantic Web - ISWC 2009. ISWC 2009*, A. Bernstein, D.R. Karger, T. Heath, L. Feigenbaum, D. Maynard, E. Motta and K. Thirunarayan, eds, Springer, Berlin, Heidelberg, 2009, pp. 473–488. doi:https://doi.org/10.1007/978-3-642-04930-9_30.

[28] V. Papavasileiou, G. Flouris, I. Fundulaki, D. Kotzinos and V. Christophides, High-Level Change Detection in RDF(S) KBs, *ACM Transactions on Database Systems* **38**(1) (2013), ISSN 0362-5915. doi:10.1145/2445583.2445584.

[29] N. Pernelle, F. Saïs, D. Mercier and S. Thuraisamy, RDF data evolution: efficient detection and semantic representation of changes, in: *12th International Conference on Semantic Systems (SEMANTiCS 2016)*, Leipzig, Germany, 2016, p. 4. doi:10.1145/1235.

[30] H. Kondylakis, M. Despoina, G. Glykokokalos, E. Kalykakis, M. Karapiperakis, M.-A. Lasithiotakis, J. Makridis, P. Moraitis, A. Panteri, M. Plevraki et al., EvoRDF: A framework for exploring ontology evolution, in: *European Semantic Web Conference*, E. Blomqvist, K. Hose, H. Paulheim, A. Lawrynowicz, F. Ciravegna and O. Hartig, eds, Springer, 2017, pp. 104–108. doi:10.1007/978-3-319-70407-4_20.

[31] R. Vesse, W. Hall and L. Carr, Preserving Linked Data on the Semantic Web by the application of Link Integrity techniques from Hypermedia, in: *Proceedings of the WWW2010 Workshop on Linked Data on the Web, LDOW 2010, Raleigh, USA, April 27*, C. Bizer, T. Heath, T. Berners-Lee and M. Hausenblas, eds, CEUR Workshop Proceedings, Vol. 628, CEUR-WS.org, 2010.

[32] R. Vesse, Link integrity for the Semantic Web, PhD thesis, Faculty of Physical and Applied Science Electronics and Computer Science, University of Southampton, 2012.

[33] M. Stefanidakis and I. Papadakis, Linking the (Un)Linked Data Through Backlinks, in: *Proceedings of the International Conference on Web Intelligence, Mining and Semantics*, WIMS '11, ACM, New York, NY, USA, 2011, pp. 61–1615. ISBN 978-1-4503-0148-0. doi:10.1145/1988688.1988759.

[34] F. Scharffe and J. Euzenat, MeLinDa: an interlinking framework for the web of data, *CoRR* **abs/1107.4502** (2011).

[35] R. Isele and C. Bizer, Learning linkage rules using genetic programming, in: *Proceedings of the 6th International Conference on Ontology Matching-Volume 814*, CEUR-WS.org, 2011, pp. 13–24.

[36] M. Bilenko and R.J. Mooney, Adaptive Duplicate Detection Using Learnable String Similarity Measures, in *KDD '03*, Association for Computing Machinery, New York, NY, USA, 2003, pp. 39–48. ISBN 1581137370. doi:10.1145/956750.956759.

[37] A.N. Ngomo, M.A. Sherif and K. Lyko, Unsupervised Link Discovery through Knowledge Base Repair, in: *The Semantic Web: Trends and Challenges - 11th International Conference, ESWC 2014, Anissaras, Crete, Greece, May 25-29, 2014. Proceedings*, V. Presutti, C. d'Amato, F. Gandon, M. d'Aquin, S. Staab and A. Tordai, eds, Lecture Notes in Computer Science, Vol. 8465, Springer, 2014, pp. 380–394. doi:10.1007/978-3-319-07443-6_26.

[38] F.M. Suchanek, S. Abiteboul and P. Senellart, PARIS: Probabilistic Alignment of Relations, Instances, and Schema, *Proceedings of the VLDB Endowment* **5**(3) (2011), 157–168, ISSN 2150-8097. doi:10.14778/2078331.2078332.

[39] F. Liu and X. Li, Using metadata to maintain link integrity for linked data, in: *Internet of Things (iThings/CPSCom), 4th International Conference on Cyber, Physical and Social Computing*, IEEE, Dalian, China, 2011, pp. 432–437. doi:10.1109/iThings/CPSCom.2011.58.

[40] A. Meehan, D. Kontokostas, M. Freudenberg, R. Brennan and D. O' Sullivan, Validating Interlinks Between Linked Data Datasets with the SUMMR Methodology, in: *International Conferences On the Move to Meaningful Internet Systems*, C. Debruyne, H. Panetto, R. Meersman, T.D. Kuhn, D. O'Sullivan and C.A. Ardagna, eds, Springer, 2016, pp. 654–672. doi:https://doi.org/10.1007/978-3-319-48472-3_39.

[41] M. Nentwig, M. Hartung, A.-C. Ngonga Ngomo and E. Rahm, A survey of current link discovery frameworks, *Semantic Web* **8**(3) (2017), 419–436.

[42] M. Schmachtenberg, C. Bizer and H. Paulheim, Adoption of the linked data best practices in different topical domains, in: *International Semantic Web Conference*, P. Mika, T. Tudorache, A. Bernstein, C. Welty, C. Knoblock, D. Vrandecic, P. Groth, N. Noy, K. Janowicz and C. Goble, eds, Springer, 2014, pp. 245–260. doi:https://doi.org/10.1007/978-3-319-11964-9_16.

[43] J.C.D. Reis, C. Pruski and C. Reynaud-Delaître, State-of-the-art on mapping maintenance and challenges towards a fully automatic approach, *Expert Systems with Applications* **42**(3) (2015), 1465–1478, ISSN 0957-4174. doi:https://doi.org/10.1016/j.eswa.2014.08.047.

[44] A. Groß, C. Pruski and E. Rahm, Evolution of biomedical ontologies and mappings: overview of recent approaches, *Computational and structural biotechnology journal* **14** (2016), 333–340.

[45] R. Cornet and N. de Keizer, Forty years of SNOMED: a literature review, in: *BMC medical informatics and decision making*, Vol. 8, BioMed Central, 2008, p. 2. doi:10.1186/1472-6947-8-S1-S2.

[46] J. Dean and S. Ghemawat, MapReduce: Simplified Data Processing on Large Clusters, *Communications ACM* **51**(1) (2008), 107–113, ISSN 0001-0782. doi:10.1145/1327452.1327492.