What are Links in Linked Open Data? A Characterization and Evaluation of Links between Knowledge Graphs on the Web

Armin Haller, Javier D. Fernández, Maulik R. Kamdar, Axel Polleres

Arbeitspapiere zum Tätigkeitsfeld
Informationsverarbeitung, Informationswirtschaft und Prozessmanagement
Working Papers on Information Systems, Information Business and Operations

Nr./No. 02/2019
ISSN: 2518-6809
URL: http://epub.wu.ac.at/view/p_series/S1/

Herausgeber / Editor:
Department für Informationsverarbeitung und Prozessmanagement
Wirtschaftsuniversität Wien · Welthandelsplatz 1 · 1020 Wien
Department of Information Systems and Operations · Vienna University of Economics and Business · Welthandelsplatz 1 · 1020 Vienna

Abstract. Linked Open Data promises to provide guiding principles to publish interlinked knowledge graphs on the Web in the form of findable, accessible, interoperable and reusable datasets. We argue that while as such, Linked Data may be viewed as a basis for instantiating the FAIR principles, there are still a number of open issues that cause significant data quality issues even when knowledge graphs are published as Linked Data. Firstly, in order to define boundaries of single coherent knowledge graphs within Linked Data, a principled notion of what a dataset is, or, respectively, what links within and between datasets are, has been missing. Secondly, we argue that in order to enable FAIR knowledge graphs, Linked Data misses standardised findability and accessability mechanism, via a single entry link. In order to address the first issue, we (i) propose a rigorous definition of a naming authority for a Linked Data dataset (ii) define different link types for data in Linked datasets, (iii) provide an empirical analysis of linkage among the datasets of the Linked Open Data cloud, and (iv) analyse the dereferenceability of those links. We base our analyses and link computations on a scalable mechanism implemented on top of the HDT format, which allows us to analyse quantity and quality of different link types at scale.

This article is under submission for the Journal of Data and Information Quality (ISSN 19361956)

1 INTRODUCTION

While the term *knowledge graph* has been in use in information science for several decades, the inclusion of knowledge panels on the Google Search engine and the accompanying Google blog entry in 2012, has since significantly changed the game not only for Web search engines, but also for data integration within other enterprises and services. Yet, most of the prominent examples termed "knowledge graphs" are *closed* knowledge bases described, for instance, as "intelligent model[s] [...] that understand real-world entities and their relationships to one another" [1]. They have been developed and curated within single enterprises, and are not available to the public, but represent the relevant entities for a particular domain (e.g., common categories and entities relevant for Web search) very well.

In parallel to the rise of the term *knowledge graph*, the *FAIR principles* were published in 2016 [53]. In contrast to the above-mentioned current trend to keep valuable knowledge graphs closed, the FAIR principles have been imposed by the scientific research community to claim the importance of improving the Findability, Accessibility, Interoperability, and Reusability of digital assets, with an emphasis on machine-actionability (i.e., the capacity to automate the task to find, access, interoperate, and reuse data).

1.1 Linked Data Principles and Linked Open Data Cloud

Interestingly – since long before the term *knowledge graph* or *FAIR principles* became popular – these two trends have both been pre-dated by another initiative set up to publish graph-shaped data assets in an openly accessible manner using standard Web protocols, extended by four simple publishing principles for data, commonly termed under the name *"Linked Data"*, imposed by Tim Berners-Lee in 2006 (with some refinements in 2009) [7]:

- (**LDP1**) use URIs as identifiers for things;
- (**LDP2**) use HTTP URIs so those identifiers can be dereferenced;
- (**LDP3**) return useful information upon dereferencing of those URIs using a standard format (typically, RDF [42]) ; and
- (**LDP4**) include links using externally dereferenceable URIs.

Data publishers from different domains have published numerous datasets following these principles over the past 10 years. Each of these datasets represents a domain-specific knowledge asset, which can be crawled and collected from the Web. The four Linked Data principles, if followed correctly, provide: *i) accessability* through relying on the commonly implemented HTTP protocol, and *ii) interoperability* through relying on a common data format (RDF), out of the box.

While *re-usability* and *findability* are not directly addressed by the Linked Data principles alone, significant efforts have been made to catalog and curate Linked Data assets in the so-called *"Linked Open Data (LOD) Cloud"* [1]. The LOD cloud provides meta-data (e.g. concerning license information, basic descriptive statistics of datasets, and entry links for crawling domain-specific datasets). Thus, we may argue that Linked Data and its principles both capture and combine the idea of both knowledge graphs and FAIR principles: indeed, the Linked Data principles and the Linked Open Data "cloud" have enabled the growth of a network of interlinked graph-structured knowledge bases publicly accessible on the Web.

As such, the LOD cloud can be viewed as a network of open, interconnected knowledge graphs published on the Web, indeed including, for example, DBpedia [3, 31] and Wikidata[2] [17, 50] as the two most widely known and used open knowledge graphs. So, one may ask in how far Linked

[1] https://www.blog.google/products/search/introducing-knowledge-graph-things-not/

[2] Although Wikidata is not part of the LOD cloud "diagram" [1] itself as of yet!

Data has been successful in establishing a network of FAIR knowledge graphs or why — so far — has it not?

In the present paper, we critically and systematically assess the network of knowledge graphs available and accessible as Linked Data, in terms of analyzing the most critical quality aspect of a true "network" of open interconnected knowledge graphs: **links**. The last and arguably the most important of the four Linked Data principles (**LDP4**) is to *"Include links to other URIs, so that they can discover more things"*. This principle has also been the basis for the promise that filled the community with enthusiasm by the to-be-expected network effects and scale-free property to not only build domain-specific knowledge graphs in isolation, but in fact dynamically grow a virtual single knowledge graph. However, while links may be considered the greatest strength of Linked Data, they are also it's greatest vulnerability. The following are a few exemplary reasons:

- references to a large number of inaccessible URIs (i.e., broken links may render a dataset largely useless). In some cases, the information (triples) from the "external" dataset can be copied into the local dataset, which in turn creates redundancies as another downside.
- changes in the external dataset to which one links are out of the control of the data publisher.
- publishing datasets as Linked Data does not necessarily keep the dataset in one place. Thus, when crawling Linked Data it is typically hard to determine which links are actually "internal" (i.e., links between parts of one coherent dataset or "knowledge graph"), and which ones are "external" (i.e., links between different datasets).

These issues are aggravated as the sheer notions of "dataset" and "link" are not even clearly defined in RDF or in the Linked Data principles.

1.2 The Need to Define the Notion of a 'Dataset' and a 'Link' on the LOD Cloud

What is a dataset? When RDF data is published according to Linked Data principles, there is no notion about the sets of triples which form a dataset, or – in other words – a coherent knowledge graph that taken on its own provides a useful asset of information. In fact, Linked Data datasets published on the Web are often partitioned in several files (each of which forming, strictly speaking, a *separate RDF graph*) or made available through Linked Data APIs or are in *separate named graphs* behind SPARQL endpoints [37]. It is not specified further though in the Linked Data principles, how one can declare that such collections of RDF graphs form a dataset, where common practices suggest though, that single datasets and the URIs "belonging" to these datasets can be referred to by sharing a common *namespace*.

However, this notion of a namespace is typically not tied to a notion of authority, as opposed to the original intention of URIs in the Web architecture, cf. Section 3.2 of RFC3986 [45], which defines authority as an integral part of URIs as follows:

```
URI = scheme ":" [//authority] path ["?"query] ["#"fragment]
```

RFC3986 further states that typically "URI schemes include a hierarchical element for a naming authority so that governance of the namespace defined by the remainder of the URI is delegated to that authority". This notion of a namespace and thereby authority, however, is blurry in RDF: it depends on the RDF serialization, whether the prefix of an identifier determining the namespace is clearly recognizable as such or not, as opposed to XML, for instance, which rather considers identifiers as clearly separated pairs of namespace URIs and qualified names [46]. Authority in HTTP URIs (which are prevalently used for IDs in Linked Data and RDF), is typically determined by the pay-level domain, though there are arguments for finer-grained notions, or subdivisions of namespaces including parts of the path or specific sub-domains necessary to determine the authoritative namespace part of a URI.

In this sense, the lack of an explicit notion of namespace and the authority of a namespace for a particular URI makes the question to which dataset a certain URI "belongs" difficult, if not impossible to answer by automated means. A dataset may contain several namespaces and a namespace may be authoritative for several datasets.

While not being one of the Linked Data core principles, best practices have been suggested to solve this issue, by declaring certain namespace prefixes to be authoritatively owned by the dataset within metadata [36]. However, Linked Datasets do not consistently publish these authoritatively owned namespace(s) contained within the dataset. For example, according to Polleres et al. [36] and again validated in our analysis, 53.8% of all datasets in the LOD cloud did not explicitly declare their namespace(s).

The lack of notion for namespace and dataset boundaries leads to several problems. First and foremost, it means that users do not know which data and URIs are authoritatively owned by which dataset, while also not knowing what data is reused and potentially extended from other authoritative sources. We argue that without the notion of authoritative namespaces per dataset, it is impossible to determine clear boundaries between datasets and to analyze links between datasets.

What is a link? In contrast to hyperlinks on the traditional document Web which have a clear direction (from one document to another), links in Linked Data, and as such the LOD cloud, do not have a clear definition. For example, the "link" counts on the LOD cloud, rely on self-declared numbers to be entered by dataset providers in a meta-data form; rather than a principled, unambiguously (re-)computable definition of links, the lod-cloud.net Web page states as the following instruction for this form:[3]

> *"The dataset must be connected via RDF links to a dataset that is already in the diagram. This means, either your dataset must use URIs from the other dataset, or vice versa. We arbitrarily require at least 50 links."*

Here – without further clarification of ownership or authority for URIs – it is not clear what "use" of URIs from another dataset means, and also the "vice versa" leaves the direction of such a link open, i.e., which dataset A actually links to which other dataset B.

To fill this gap, we suggest to start from the notion of a triple in RDF (or an edge in the graph), which is often taken synonymously with the notion of a typed link from subject to object. For example:

- **t1:** [`dbpedia:Wolfgang_Amadeus_Mozart`, owl:sameAs, `wikidata:Q254`] establishes equality between individuals published under different URIs belonging to different datasets (i.e., Wolfgang Amadeus Mozart entities belonging in DBpedia and WikiData are the same)
- **t2:** [`dbpedia:Wolfgang_Amadeus_Mozart`, rdf:type, `dbpedia_ontology:Person`] denotes that an individual is of a certain type (i.e., Wolfgang Amadeus Mozart was a "Person" as defined by the DBpedia ontology [3])
- **t3:** [`dbpedia:Wolfgang_Amadeus_Mozart`, `foaf:name`, "Wolfgang Amadeus Mozart"@en] denotes the name of an individual (i.e., `dbpedia:Wolfgang_Amadeus_Mozart` has the name Mozart, as defined in the FOAF ontology [9])

Note, however, that – on a dataset level – the direction of the link (i.e., whether the first triple **t1** may be considered a link from DBpedia to Wikidata or vice versa) does not depend on whether the respective triple has a DBpedia or a Wikidata URI in its subject, but rather on the fact in which dataset the triple appears. Also, if we assume that the respective triples were all published within the DBpedia dataset, that we can distinguish different kinds of outlinks, **t1** denotes a link to an individual in another dataset, whereas **t3** actually links to an externally defined ontology.

[3] `https://lod-cloud.net/#about`

Although previous works (e.g., Schmachtenberg et al. [41]) have analyzed the number of links between a sample of documents in the LOD cloud and discussed their relative lack, a formal definition of interlinking and distinction between different types of links has been missing from the literature. Also, links have been considered to be directional in previous work, i.e. a link is between the entity identified by the subject and the entity identified by the object [54]. However, a dataset publisher may reuse an external resource in the subject position of a triple in their dataset. Our definitions and analysis of links herein shall capture and clarify these cases.

To address these issues, we first propose a rigorous definition of a naming authority for a Linked Dataset in this paper. We aim to distinguish internal references within the dataset from links to data defined in external datasets. Consequently, we can provide concrete definitions of links between datasets and then define different link types in Linked Datasets. We present automated methods to analyze different link types at scale, and provide an empirical analysis of linkage among the datasets of the Linked Open Data cloud.

The remainder of this paper is structured as follows. We present preliminaries, including definitions of what we mean by datasets and links in **Section 2**. Previous work conducted to analyze the availability, quality, and "linked-ness" of the Linked Open Data (LOD) cloud is discussed in **Section 3**. We then present our methodology for analysing links, including the establishment of the dataset corpus, the ontology corpus, and definitions on a dataset authority and namespace, in **Section 4**. We conclude this section by defining link types. We present results of our computation of links in practice on a corpus of datasets registered in the LOD cloud in **Section 5**, in particular also in terms of quality of links and quantifying issues related to "broken" links. We discuss our observations and insights gained from this analysis in **Section 6**. We conclude in **Section 7**.

2 PRELIMINARIES AND DEFINITIONS

The lack of a clear definition of what a "dataset" (i.e., a coherent knowledge graph) in Linked Data comprises has already been emphasised as early as 2008. Cyganiak et al. [13] propose a metadata-mechanism, in the form of Semantic Sitemaps to scope and describe the set of actually published files that form a dataset. However, as claimed in Polleres et al. [36], this schema is hardly used consistently across datasets. Therefore, we propagate a definition based rather on intuition, which we will thereafter empirically test in our evaluation below.

Definition 2.1. A *dataset* is a collection of one or more associated RDF graphs, published by a single controlling entity either as single or separate files, or accessible via a common SPARQL endpoint. Given a dataset ds, we denote by G_{ds} the merge of all of its graphs.

Here, when we say "published by a single controlling entity", we mean that a single controlling entity has the right or possibility to take the whole dataset offline and/or change RDF triples in the respective graphs composing the dataset. We further assume that datasets authoritatively control a subset of the mentioned URIs in the dataset, by prefixes.

Definition 2.2. We assume each dataset uses a finite set of *namespaces*,[4] (i.e., URI-prefixes), some of which it controls authoritatively. Given a dataset ds, we denote by NS_{ds} the set of its authoritative namespaces for ds. Moreover, we assume each namespace is authoritatively controlled by at most a single dataset. That is, we assume that $ds_1 \neq ds_2$ implies that $NS_{ds_1} \cap NS_{ds_2} = \emptyset$.

[4]While, datasets themselves can be infinite in principle, for instance, the dynamically generated Linked Open Numbers dataset [51].

Typical mechanisms for namespace authority is the ownership of a certain pay-level-domain. However, disjoint datasets hosted under the same pay-level domain are possible.[5] Next, with reference to common, established notions of standard use of the OWL and RDF vocabularies and under the assumption that triples with non-standard use of these vocabularies are ignored, we distinguish between different types of URIs, depending on their positions in triples.

Definition 2.3 (Non-Standard-use, extending Definition 5.5 of Hogan [22]). Let RDF, RDFS, OWL and XSD, denoted by the prefix URIs `http://www.w3.org/1999/02/22-rdf-syntax-ns#`, `http://www.w3.org/2000/01/rdf-schema#`, and `http://www.w3.org/2002/07/owl#`, respectively, denote the *reserved* namespaces. Let G_{RDF}, G_{RDFS}, and G_{OWL}, resp., denote the RDF graphs accessible at these URIs, where we write $G_{res} = G_{RDF} \cup G_{RDFS} \cup G_{OWL}$. A *non-standard triple* in any RDF graph other than G_{res} is a triple where:

- a class in G_{res} appears in a position other than as the value of `rdf:type`, or
- a property in G_{res} appears outside of the predicate position.

Assuming a triple with standard vocabulary use, we distinguish class positions, property positions, datatype positions, and instance positions of URIs outside of one of the reserved namespaces as follows:

Definition 2.4. A URI u outside of one of the reserved namespaces in an RDF triple $t = (s, p, o)$ is in a *class position* if

- $s = u \ \wedge \ p \in \{p | (p, \texttt{rdfs:domain}, \texttt{owl:Class}) \in G_{res} \vee (p, \texttt{rdfs:domain}, \texttt{rdfs:Class}) \in G_{res}\}$
- $o = u \ \wedge \ p \in \{p | (p, \texttt{rdfs:range}, \texttt{owl:Class}) \in G_{res} \vee (p, \texttt{rdfs:range}, \texttt{rdfs:Class}) \in G_{res}\}$
- $o = u \ \wedge \ p = \texttt{rdf:type}$

Definition 2.5. A URI u outside of the reserved namespaces in an RDF triple $t = (s, p, o)$ is in a *property position* if

- $s = u \wedge$
  $p \in \{p | (p, \texttt{rdfs:domain}, \texttt{owl:ObjectProperty}) \in G_{res}\} \cup \{p | (p, \texttt{rdfs:domain}, \texttt{rdf:Property}) \in G_{res}\}$
- $p = u$
- $o = u \wedge$
  $p \in \{p | (p, \texttt{rdfs:range}, \texttt{owl:ObjectProperty}) \in G_{res}\} \cup \{p | (p, \texttt{rdfs:range}, \texttt{rdf:Property}) \in G_{res}\}$

Definition 2.6. A URI u outside of the reserved namespaces in an RDF triple $t = (s, p, o)$ is in a *datatype position* if

- $s = u \ \wedge \ p \in \{p | (p, \texttt{rdfs:domain}, \texttt{rdfs:Datatype}) \in G_{res}\}$
- u occurs as the datatype of a typed literal $o = \text{"}l\text{"}\hat{}\hat{}u$
- $o = u \ \wedge \ p \in \{p | (p, \texttt{rdfs:range}, \texttt{rdfs:Datatype}) \in G_{res}\}$

Definition 2.7. A URI u outside of the reserved namespaces in an RDF triple $t = (s, p, o)$ that is neither in a class, nor property, nor datatype position, is in an *instance* position.

Based on its position we can now distinguish link types for URIs:

Definition 2.8. Let ds_1, ds_2 be datasets. Then, we call triple $t \in G_{ds_1}$ a *link* from ds_1 to ds_2, if t contains a URI u from a namespace in NS_{ds_2}. Depending on the position of u we further distinguish:

- t is called an *instance link*, if u is in an instance position in t.
- t is called an *ontology link*, otherwise, where we further distinguish TBox-Links as follows:

[5]For example, different Linked Data datasets hosted on Github using `https://github.com/USERNAME/`-prefixed URIs, where the username determined the authority instead of the pay-level-domain.

- t is called a *class link*, if u is in a class position other than the o position of an `rdf:type` triple, i.e., a link to a class from an external dataset in a TBox statement.
- t is called an *instance typing link*, if u in a the class position $o = p$ of an `rdf:type` triple, i.e., a link from an individual to a class from an external dataset in an ABox statement.
- t is called a *property link*, if u in in a property position other than p, i.e., a link to a property of an external namespace in a TBox statement.
- t is called an *instance role link*, if u is in the property position $u = p$, i.e., a link between individuals, referring to a property from an external dataset in an ABox statement.

Finally, if u does not appear in G_{ds_2}, we call t a *broken* link.

Example 2.9. For instance, let ds_1 be DBpedia with `http://dbpedia.org/resource/`, `http://dbpedia.org/ontology/` $\in NS_{ds_1}$; ds_2 be the FOAF ontology which uses a single namespace, i.e, $NS_{ds_2} = \{$`http://xmlns.com/foaf/0.1/`$\}$; finally, let ds_3 be Wikidata with the namespaces `http://www.wikidata.org/entity/`, `http://www.wikidata.org/prop/direct/` $\in NS_{ds_3}$. We shall denote these namspeaces with the prefixes `dbr:`, `dbo:`, `foaf:`, `wd:`, and `wdt:`, respectively. Let us consider the example triples from Section 1.2: the triple

$$t_1 = \texttt{dbr:Wolfgang\_Amadeus\_Mozart owl:sameAs wd:Q254 .}$$

in $ds1$ then is an instance link, from $ds1$ to $ds3$, whereas

$$t_2 = \texttt{dbr:Wolfgang\_Amadeus\_Mozart rdf:type dbo:Person.}$$

is not a link, but rather an internal refernce within ds_1. However,

$$t_2' = \texttt{dbr:Wolfgang\_Amadeus\_Mozart rdf:type foaf:Person.}$$

would be an ontology link, more specifically, an instance typing link from ds_1 to ds_2. Next,

$$t_3 = \texttt{dbr:Wolfgang\_Amadeus\_Mozart foaf:name "Wolfgang Amadeus Mozart"@en .}$$

is an example of a property link from ds_1 to ds_2. Finally,

$$t_4 = \texttt{dbo:Person rdfs:subClassOf foaf:Person}$$

is a class link from ds_1 to ds_2.

Further, assuming that the url `dbr:Wolfgang_A._Mozart` does not apear in ds_1 and `foaf:kowns` does not appear in ds_2 then,

$$t_5 = \texttt{wd:Q254 p:P2888 dbr:Wolfgang\_A.\_Mozart .}$$

appearing in ds_3 would be an example of a broken instance link, whereas

$$t_6 = \texttt{dbr:Wolfgang\_Amadeus\_Mozart foaf:kowns dbr:Antonio\_Salieri .}$$

would be an example of a broken ontology link. Whereas the last two examples of broken links are fictitious, we will provide a more thorough discussion of real broken links in practice, which constitute a significant quality problem for linked knowledge graphs, as part of our analysis in Section 5.2.6.

Analogously to the definition of links, it also makes sense to distinguish authoritative namespaces with respect to their usage (in instance, class, property, and datatype positions):

Definition 2.10 (Instance namespace/ontology namespace). Let *ns* be an authoritative namespace for dataset *ds*. We call *ns* an *instance namespace* of *ds*, if all URLs *u* within *ns* appear in G_{ds} only in instance positions. Analogously, we call *ns* an *ontology namespace* of *ds*, if all URLs *u* within *ns* appear in G_{ds} only in non-instance positions; ontology namespaces can be further subdivided into *class*, *property* and *datatype* namespaces, if they happen to be used within G_{ds} exclusively in the respective position.

Example 2.11 (cont'd). `dbr:`, i.e. `http://dbpedia.org/resource/` is an instance namespace for dbpedia, whereas `dbo:`, i.e., `http://dbpedia.org/ontology/` and `http://dbpedia.org/propeerty/`, are ontology and property namespaces, respectively, for DBpedia.

Before we further analyze how the notions introduced in this Section apply to knowledge graphs published as Linked Data datasets "in the wild" and analysing the different link types in different datasets can be implemented using SPARQL queries and HDT in Sections 4+5 below, let us review related works on "linked-ness" and link quality in the context of Linked Open Data cloud.

3 RELATED WORK

Starting from 2007 onwards, publishers have used Semantic Web technologies, such as RDF, OWL, and SPARQL querying language, to publish and link their datasets on the Web. These datasets may be available as RDF/OWL data dumps and may also be exposed through an interface that enables the users to formulate SPARQL queries (i.e., a SPARQL endpoint).

To keep track of all the sources whose datasets have been published and linked on the Web, the Semantic Web community proposed a starting point of entry for any new user who wishes to use these Linked Datasets in their research. The `LOD-cloud.net` [1] is this starting point, and different snapshots of the Linked Open Data (LOD) cloud show the growth and evolution of the cloud from 12 linked sources in 2007 (as the first prototype) to more than 1,200 linked sources, as of June 2018, with datasets being published from several different domains, such as the life sciences, geography, economics, politics, and media. Until recently, the LOD cloud diagram at `LOD-cloud.net` has been generated by looking at the Linked Dataset descriptions and metadata catalogued at the DataHub repository[6]. Several efforts have been undertaken to evaluate the availability, quality, and the "linked" nature of the LOD cloud using a myriad of approaches.

3.1 Availability and Discoverability of Linked Open Data sources

There have been numerous studies that investigate and evaluate the availability and discoverability of the LOD cloud using the list of SPARQL endpoints and RDF data dumps access URIs that are listed on the (now discontinued) DataHub repository (which has been the basis of the creation of the LOD cloud diagram on `LOD-cloud.net`). Vandenbussche et al. [48] found that many of the SPARQL endpoints in the LOD cloud had issues with availability and only 32.2% were available for more than 95% of the time over a 27 month period between 2013 and 2015. Debattista et al. [15] evaluated the 2014 version of the LOD cloud, and found that out of 569 Linked Data sources, only around 42% (i.e., 239 sources) had an available Linked Data access point (i.e., a data dump URI or a SPARQL endpoint). On conducting a preliminary analysis in 2017, Polleres et al. [36] found that while the 2017 version of the LOD cloud had 1,281 sources, only 50% (i.e., 646 sources) had a possible Linked Data access point. In this paper, we demonstrate that the availability of SPARQL endpoints in the LOD Cloud has dropped even further in 2019.

It has to be emphasized again that the LOD cloud diagram is created from the source metadata descriptions from the DataHub repository – thus, not all the metadata entries may have been

[6] `http://old.datahub.io`

updated to reflect the current resources and access points, and sources that provide a Linked Data access point may not even be listed on the DataHub repository, and hence not included in the LOD cloud diagram. To the best of our knowledge, there are no approaches that can evaluate, at scale and without seed URIs, all possible Linked Data access points available currently on the Web.

3.2 Metadata Representation and Quality

Representation of metadata of a Linked Dataset (i.e., class and property characteristics, number of instances and assertions, and also the incoming and outgoing links from a dataset) has been a widely-discussed issue within the Semantic Web community. Alexander et al. [2] proposed the Vocabulary of Interlinked Datasets (VoID) specification to achieve this goal. VoID statistics and metrics can be used for SPARQL query federation (i.e., the methodology to process and execute SPARQL queries across multiple sources on the LOD cloud), and some query federation engines, such as SPLENDID [19], support the processing of VoID-annotated metadata. However, Debattista et al. [15] found that most SPARQL endpoints and RDF data dumps, in the current state of the LOD cloud, do not provide the VoID statistics along with the Linked Dataset. While, Debattista et al. [15] extensively analyzed a small subset of LOD datasets using 27 Linked Data quality metrics (e.g., licensing, provenance, availability, metadata) that are proposed by Zaveri et al. [54], this study did not perform any analysis to detect authoritatively-owned namespaces.

Hogan et al. [25] proposed a set of fourteen guidelines (e.g., dereferenceable and short HTTP URIs, licensing, metadata) to publish good quality Linked Data on the Web. They evaluate ≈4 million RDF/XML documents constituting of over 1 billion quadruples. Certain guidelines are widely adhered to by data publishers (e.g., HTTP URIs, stable URIs) whereas certain guidelines pertaining to data licensing and human-readable metadata representation are almost always ignored.

Rietveld et al. [38] presented an automated approach to compute metadata statistics of the different datasets in the LOD Laundromat [5], a catalogue of (re)published and cleaned LOD datasets. The LOD Laundromat Meta-Dataset contains provenance annotations and uses de-facto Semantic Web vocabularies (e.g., VoID) for publishing the metadata. However, no analysis has yet been performed to detect authoritatively-owned namespaces across the datasets.

3.3 Authoritative Namespaces and Links Between Linked Datasets

Schmachtenberg et al. [41] crawled the LOD cloud in 2014 with a seed set of URIs and retrieved more than 900,000 documents describing more than 8 million resources. They found that only 56% of all datasets in their corpus link to other datasets. The analysis did not determine an authoritative namespace for a dataset to determine the link statistics, but they considered two datasets to be linked if there exists at least one RDF link between resources belonging to both datasets. As such, the number and type of links between datasets were only captured if both resources in the link existed in the corpus. It was observed that owl:sameAs is the most important linking predicate within most Linked Dataset categories, followed by rdfs:seeAlso. As shown in our analysis (**Section 4.4**), owl:sameAs and rdfs:seeAlso predicates play an insignificant role in the number of links between datasets. We consider all links, even if the resource that is linked to does not exist in our corpus, but is outside the authoritative namespace. Although their analysis did record the predicate type that was used to link, they did not distinguish between *Ontology Links*and *Instance Links*, whereas our analysis shows that the majority of links are ontology links.

Harth et al. [20] introduces the notion of a naming authority (i.e., a data source with the power to define identifiers of a certain structure). The authors use the PageRank algorithm to assign authority values to data sources based on a naming authority graph, and then propagate the authority values to identifiers referenced in the sources. In this paper, we are also interested in a naming authority, more specifically the authoritative namespace of data (i.e., classes, properties, and individuals).

Hogan et al. [23] crawled the LOD cloud in 2010 and analyzed the crawled corpus with ≈ 150 million URIs. The analysis discovered several issues pertaining to the accessibility and dereferenceability of the URIs, lack of structured data retrieved on lookup (**LDP3**), misreported content types, syntax errors, reasoning errors due to ontology hijacking (i.e., new ontologies published on the Web re-defining the semantics of existing concepts resident in other ontologies), misplaced classes or properties, misuse of established OWL and RDFS built-ins, and errors due to use of deprecated URIs. We will showcase that some of these issues are still prevalent in the LOD cloud a decade later.

Hogan et al. [24] later define authoritative sources for ontologies and discuss the problem of ontology hijacking in greater detail. Although we also consider this as bad practice, all links from ontologies to other ontologies are considered in our analysis, that is, we are also interested in links from an ontology that redefines the semantics of classes or properties defined in the authoritative source URI for these corresponding classes or properties.

Butt et al. [10] published a collection of ontologies that was retrieved by crawling a seed set of ontology URIs derived from `prefix.cc`. Several ranking algorithms were used to compute the centrality of concepts within the ontology they were defined in and within the ontology corpus. In this paper, we also use a crawl of `prefix.cc` to establish a set of classes and properties and their authoritative namespace.

3.4 Linked Data Profiling and Link Analysis Tools

Recently, there have been several tools that have reached a state of maturity for profiling Linked Data. ProLOD [8] was an early proposal for a profiling tool that assessed object values in RDF data from DBpedia, counted the number of external links and presented the value distribution of literal objects. LOUPE, as a more advanced profiling tool, uses a series of parameterized queries to unveil links between datasets and ontologies [33]. ABSTAT generates summaries of Linked Datasets using statistical methods to provide beginner users an understanding with respect to the set of assertions, ontology subscriptions, and minimal patterns used in a given dataset [35, 43]. LODVader proposes to serve as a Linked Data discovery entrypoint by maintaining a time-updated fast search index created through use of several profiling, analyses, and visualization components [4]. Hasnain et al. [21] generated a preliminary roadmap composed of profiles catalogued from more than 80 Linked Datasets pertaining to life sciences. Spahiu et al. [44] provide a framework to profile the quality of `owl:sameAs` property in the LOD cloud and automatically discover new similarity links giving a similarity score for all the instances without prior knowledge about the properties used. Debattista et al. [14] propose a conceptual methodology to profile and assess the quality of Linked Datasets and develop the Luzzu framework for evaluating the quality of several statistics-related Linked Datasets across several quality metrics. Ben Ellefi et al. [6] use dataset profiles, characterized through the set of schema concept labels present in the dataset and can be enriched using textual descriptions of classified instances, to detect overlaps and linking candidates across different Linked Datasets.

However, the Linked Data profiling algorithms have either only been implemented over some of the popular SPARQL endpoints or subsets of the LOD cloud (e.g., life sciences) and do not account for variable SPARQL versions. They also rely on fixed set of properties (e.g., `owl:sameAs`) or require the retrieval of all instances and assertions in the corpus which is often not scalable for all LOD datasets. If they compute statistics on links, they distinguish internal from external links based on the position of the entity in the triple and if the URI belongs to the dataset authority or not. However, they do not have the notion of different link types and of a namespace authority that allows us to analyse links between datasets and ontologies regardless of the position of the entity in a triple.

3.5 Domain-specific Analyses of Life Sciences Linked Open Data

There have been several domain-specific efforts to evaluate the availability, quality, and reuse across Linked Data sources. Several data and knowledge publishers in biomedical domains have published and linked their sources on the Web [11, 28, 39, 40, 52]. Indeed, several linked biomedical data and knowledge sources (i.e., biomedical ontologies) are present in the current LOD cloud diagram (available at `LOD-cloud.net`), listed under the 'Life Sciences' region. Hu et al. [26] conducted a link analysis on the datasets published by the Bio2RDF project [11] in the LOD cloud. Specifically, they evaluated the links between different Bio2RDF datasets, estimated symmetry and transitivity of links between Bio2RDF domain-specific entities (e.g., drugs and genes), and exhaustiveness of different predicates (e.g., owl:sameAs, bio2rdf:x-ref) to link similar entities. While the study offered promising results, with room for improvement, it was only focused on a small set of Linked Datasets published under the same Bio2RDF project.

Kamdar et al. [29] performed a systematic analysis over heterogeneous biomedical ontologies in the BioPortal repository to detect and estimate class reuse (i.e., when a class URI from one ontology is reused in another ontology) and class overlap (i.e., when similar classes are present in different ontologies). The study observed minimal reuse of classes (with the correct URI representation) but high levels of overlap across these biomedical ontologies (e.g., multiple ontologies use different URIs for the class Cardiac Muscle). Kamdar [27] conducted a similar analysis on vocabulary reuse and label mismatch (i.e., when different class or property URIs are used in different Linked Datasets to model similar information, such as drug–protein target interaction). Moreover, both studies document 'intent for reuse' in data and knowledge publishers. That is, publishers wish to link and reuse to classes, properties, and instances in existing sources, but end up using different and often incorrect URI representations, with faulty namespaces and deprecated versions. While these studies do not rely on the list of endpoints from the DataHub repository, and exhaustively analyze the quality, reuse, and "linked" characteristics of the Linked Datasets (or ontologies) in the corpus, they are limited in focus (i.e., only life sciences LOD) and require domain-specific knowledge.

Since the LOD cloud diagram is often represented to be the face of the Semantic Web movement, the lack of availability of resources on the Web as well as quality issues (i.e., lack of reuse, intent for reuse, semantic mismatch) have negative implications. If the LOD sources do not have available Linked Data access points with high availability and quality, then the research and development of Semantic Web-based methods (e.g., query federation) and tools is severely impacted.

4 METHODOLOGY

In the following sections, we describe a generic methodology to define and analyze link types in a corpus of Linked Datasets using a set of automated SPARQL queries.

4.1 Establish Dataset Corpus

It has been shown that although the LOD cloud is still growing, albeit at a slow pace, many datasets and SPARQL endpoints that service a dataset registered in the LOD cloud are not available anymore [48]. To establish our corpus we, therefore, first checked for all datasets registered in the LOD cloud[7] if they are still available. That is, we checked if there is either a functioning SPARQL endpoint or at least one usable download file available.

Table 1 shows the statistics of our analysis. As evident from the analysis, only about a quarter (i.e., 25.6%) of all datasets in the LOD cloud still have a functioning SPARQL endpoint or provide a downloadable file. The status of SPARQL endpoints was tested with the same queries proposed in Vandenbussche et al. [48] as shown in **Listing 1**.

[7] `https://lod-cloud.net/lod-data.json`

		% of total	Available	Available as % of total
Total # of Datasets	1359	100%	—	—
SPARQL Endpoint	459	33.5%	125	9.1%
Available Download	890	65.4%	226	16.6%

Table 1. Availability of a Linked Dataset as SPARQL Endpoint or as a Donwloadable RDF Dumps.

Listing 1. Queries used to test the status of SPARQL endpoints on the LOD cloud

```
ASK WHERE {?S ?P ?O .}
SELECT ?S WHERE {?S ?P ?O .} LIMIT 1
```

As we are using several computationally expensive SPARQL queries (i.e., queries that operate on all triples in the graph), we can not use those SPARQL endpoints directly but need to perform the queries locally. We therefore focused our attention on the downloadable datasets and checked the availability of downloaded RDF dumps. 64.38% of all datasets offer some form of downloadable file (i.e., one or many "full_download" and/or "other_download" locations) while the remaining 1.1% of datasets do not provide any data. However, of those that do, only 226 are still available, representing 16.6% of all download URLs. Although this is still more than the 9.1% availability of SPARQL endpoints, it is a first indication of the relatively poor health of the LOD cloud.

Therefore, to increase the size of our corpus we also included historical datasets from the LOD cloud that were cached in the LODLaundromat [5] and provided as a downloadable corpus in HDT by Debattista et al. [15]. The resulting corpus consists of 430 Linked Datasets (i.e. 214 more than currently still available in the LOD cloud), each encoded in HDT for a total size of 51 GB (uncompressed 204 GB), with a total number of 3,262,929,887 triples (i.e., ≈ 3.3 billion triples).

4.2 Establish ontology corpus

For our link analysis, we distinguish between *Ontology Links* and *Instance Links* as defined in **Section 2**. To distinguish between the two, we first need to establish a corpus of ontologies available and used in the LOD cloud.

Although ontologies typically only consist of terminological axioms T (TBox), they may also include a set of assertional axioms A (ABox). In the latter case, codelists or thesaurological terms can be defined as assertional axioms in an ontology. Contrarily, datasets registered in the LOD cloud typically consist of only assertional axioms. This is confirmed by our analysis of the 430 Linked Datasets, where only three datasets are, in fact, ontologies (without instance data)[8]: *i)* `opencyc.org` dataset (an upper level ontology), *ii)* `umbel.org` dataset (an upper ontology mapping and binding exchange layer that defines a large set of supertypes used to map individuals), and *iii)* `onto.beef.org.pl` (an ontology that forms the core of the OntoBeef Domain Thesaurus that is registered as a separate dataset). We excluded these three ontologies from our analysis of Linked Datasets (cf. **Section 5**), but included their axioms in our corpus of classes and properties.

Linked Datasets themselves, however, may also include terminological axioms, either, because an ontology is contained within the dataset, but using a different namespace, or because some new terminological axioms are defined within the same namespace as the assertional axioms in the dataset. Although the latter can be considered bad practise, it is possible, and as our analysis shows, also common (cf. **Section 5.2**).

To distinguish ontologies and their namespace from and within datasets we need to establish a corpus of ontology namespaces and the classes and properties contained within. While registration of an ontology on `prefix.cc` is often regarded as a common best practice in the Linked Open Data

[8]Please note that many of the other datasets include ontologies or even define an ontology namespace, but they predominantly contain assertional axioms (ABox)

community, this is voluntary. Consequently, it is difficult to establish such a corpus by just looking at those ontologies that are registered on `prefix.cc`, since many ontologies in the LOD cloud (and on the Web for that matter) may not be registered on such site. Hence, we use a two-step process to mitigate this situation and establish such a corpus:
Step 1: We crawl all ontology namespaces of `prefix.cc` and stored each unique class and property contained within those ontologies. This crawl is performed four times over the span of two months and yields a combined unique number of classes and properties as shown in **Table 2**.

# of unique Classes	204,616
# of unique Properties	1,821

Table 2. Ontology Corpus Statistics

Step 2: We also crawl each dataset in our corpus for declared classes and properties. To check for all classes that are declared within a dataset we perform the query shown in **Listing 2**[9]. We then record if all of the declared classes are contained within the `prefix.cc` corpus.

Listing 2. SPARQL query used to retrieve all classes that are declared within a dataset.

```
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT DISTINCT ?C WHERE {
   {?C a owl:Class. } UNION
   {?C a rdfs:Class. }
}
```

To retrieve all properties that are declared within a dataset, we follow a similar process and use the query shown in **Listing 3**. We compare the retrieved set of all unique properties with the properties contained within the `prefix.cc` corpus.

Listing 3. SPARQL query used to retrieve all properties that are declared within a dataset.

```
SELECT DISTINCT ?P WHERE {
   {?P a rdf:Property. } UNION
   {?P a owl:ObjectProperty. } UNION
   {?P a owl:DataTypeProperty. }
}
```

4.3 Establish dataset authority and authoritative namespace

At this point, there is an absence of a central authority and the presence of incomplete metadata. Note that, although the metadata in the old DataHub repository allows for manually defining the namespace of a given dataset, this information is rarely completed and is subject to manual error as described earlier. For instance, 53% of the 1,359 datasets registered in the LOD Cloud have an empty namespace. Thus, we are interested in identifying all used namespaces in the dataset and ontology corpus, and establishing the dataset authority (i.e., responsible) of each namespace. This information allows us to define, in an automatic way, *Ontology Links* and *Instance Links* between datasets (see **Section 4.4**).

Towards this aim, we first convert all datasets to HDT [18], an RDF compressed format with retrieval capabilities that is successfully deployed in client-side query processors, such as Triple Pattern Fragments (TPF) [49] and SAGE [34], indexing/reasoning systems like HDT-FoQ [32] or WaterFowl [12] or Question Answering systems [16] among others. HDT splits the RDF graph into

[9]Please note that we will define prefixes for SPARQL queries only once in the paper

Algorithm 1 Computing the namespaces and their relative percentage of a dataset

Input: `HDT(G)`, the HDT version of an RDF dataset G, and MINOCCS, the minimum number of terms in a namespace

Output: `namespaces`, a map with the relative percentage of each namespace occurring in G, namespaces:*string*→ {0..1}

```
1:  namespaces = {}, tempCounter = {}, numSubjectURIs = 0
2:  for subject ∈ HDT(G).getSubjects() do
3:     if subject ∉ BlankNodes then
4:        namespace_subj = getNamespace(subject)
5:        tempCounter[namespace_subj]++
6:        numSubjectURIs++
7:     end if
8:  end for
9:  for (namespace, count) ∈ tempCounter do
10:    if (count >= MINOCCS) then
11:       namespaces[namespace] = (count/numSubjectURIs)
12:    end if
13: end for
14: return namespaces
```

three main components: *(i)* the Header, providing general metadata of the RDF datasets (publisher and other provenance information, number of triples, etc.), *(ii)* the Dictionary, that assigns and provides a mapping between each term in the RDF graph (URIs, literals and blank nodes) and a numeric identifier, and *(iii)* the Triples, that makes use of the HDT Dictionary[10] to replace and index the original graph of terms with a graph of ids. HDT provides built-in indexes for the Dictionary and Triples components [32] that allow for efficient term and id retrieval in the dictionary, and triple pattern resolution at triple level. In particular, the HDT Dictionary splits terms by roles and lexico-graphically indexes four different subdictionaries:

SO: Shared subject-objects (i.e., all subject terms that also appear in the graph as objects).
S: Unique subjects (i.e., all terms occurring in the subject position that are not objects).
O: Unique objects (i.e., all terms occurring in the object position that are not subjects).
P: Predicates (i.e., all predicates, irrespective whether they also appear as subjects or objects).

Thus, we make use of the HDT Dictionary functionality to efficiently iterate through all different roles (subject, object and predicate) in each RDF dataset and extract all different namespaces in each RDF dataset. This method is shown in **Algorithm 1**. Given that subdictionaries are sorted lexico-graphically, the process is just limited to a series of simple steps such as namespace finding (line 4) and counting (line 5). We then compute the 'relative occurrence' of each namespace in the dataset as the percentage of each namespace over the total terms in the subdictionary (line 9), discounting blank nodes (line 3) if present.

Note that we also disregard those namespaces with a small number of occurrences. In our experiments this threshold was practically set to 50 occurrences.

The authoritativeness of each namespace is then assigned to the dataset(s) with a maximum (relative, compared to all other datasets in our corpus) occurrence, above the aforementioned threshold.

[10]The HDT-based code is available at `https://github.com/AxelPolleres/hdt-cpp/tree/develop/libhdt/tools`.

Finally, a namespace that is extensively used in a dataset may be classified as its authoritative namespace. However, we need to consider special cases, where the namespace is in fact an external link to a dataset that might not be present or available in the LOD corpus. For example, an automatic inspection on DBpedia can incorrectly determine that it is the authoritative dataset of the `wikimedia.org` namespace. To minimize this effect in our analysis, we restrict to defining only one authoritative namespace for each Linked Dataset. That is, the namespace that (i) has been assigned as an authoritative namespace of the dataset and (ii) it has the maximum relative occurrence of all authoritative namespaces in the dataset. In order to consider a wider range of URIs, for our further analysis, we only consider the Pay Level Domains (PLD) of the authoritative namespace.

Table 3 shows statistics of the process. In general, our process finds an authoritative namespace for 92% of the datasets (395 out of 430 datasets in our corpus). The missing 8% corresponds to datasets with few triples (less than our minimum threshold) and/or namespaces that are further represented in a different dataset. Note that only 65% of the datasets with authoritative namespace (i.e. 257) had an assigned namespace in the LOD cloud metadata and, from them, only 63% (i.e. 162) exactly correspond with our assigned namespaces[11]. A manual inspection of the remaining 37% reveals different errors in the metadata declaration in the LOD Cloud metadata. For example, the dataset `bbc-music` defines *http://www.bbc.co.uk/music/artist/* as the namespace, while the data actually contains only the namespace *http://purl.org/ontology/mo/*. In other cases, such as `didactalia`, the dataset includes the VOiD descriptive metadata with a different namespace (e.g. *http://didactalia.net* vs. *http://didactalia.com/*). A similar problem can be found with SPARQL endpoints, which we currently do not crawl, such as in `dbpedia-es`, and can be subject of future work.

# of Datasets in our corpus	430
# of D. with Auth. namespace	395
# of D. with namespace in LOD Cloud metadata	257
# of D. matching Auth. namespace and LOD Cloud metadata	162

Table 3. Authoritative namespace statistics

.

4.4 Link Type Analysis

As of our definitions in **Section 2** we distinguish two general types of links, Ontology (TBox) Links and Instance (ABox) Links. In the following sections, we provide more details on the SPARQL queries that correspond to the different links defined above.[12]

4.4.1 Ontology (TBox) Links. With the query shown in **Listing 4** that instantiates the definitions from **Section 2**, we retrieve all external classes (i.e., classes using a namespace other than the authoritative namespace) that are not explicitly declared as a class, but are used to *i)* define an instance (i.e., they are used in an assertional axiom), *ii)* define a terminological axiom that either extends a class through a *subclass* or *superclass* relationship, *iii)* define a class' equivalence, disjointedness, unionOf, disjointUnionOf, intersectionOf, complementOf, or "enumeration" kind, *iv)* define the domain or key of a property or range of a property, or *v)* describe a universal or existential object property expression.

Listing 4. SPARQL query used to retrieve all external classes.

```
SELECT DISTINCT ?C WHERE {
```

[11]We compare the PLDs of both our authoritative namespace and the LOD cloud metadata.
[12]The detailed statistics per authoritative namespace are published at: `https://github.com/arminhaller/LinksInLOD`

```
    {[] a ?C. } UNION
    {[] rdfs:SubClassOf ?C. } UNION {?C rdfs:SubClassOf []. } UNION
    {?C owl:disjointWith [].} UNION {[] owl:disjointWith ?C.} UNION
    {?C owl:disjointUnionOf [].} UNION
    {?C owl:equivalentClass [].} UNION {[] owl:equivalentClass ?C.} UNION
    {?C owl:intersectionOf [].} UNION
    {?C owl:unionOf [].} UNION
    {[] rdfs:complementOf ?C. } UNION {?C rdfs:complementOf []. }
    {?C owl:oneOf [].} UNION
    {[] rdfs:domain ?C. } UNION
    {[] rdfs:range ?C. } UNION
    {[] owl:onClass ?C. } UNION
    {[] owl:allValuesFrom ?C. } UNION
    {[] owl:someValuesFrom ?C. }
    FILTER (!regex(?C, "AUTHORITATIVENAMESPACEURI","i")) .
}
```

For each class URI retrieved through this query, we check its occurrence in either the subject or object position in any triple in the dataset through the query shown in **Listing 5**.

Listing 5. SPARQL query used to determine subject/object position in any triple in a given dataset.

```
SELECT ?C WHERE {
    {[] [] ?C . } UNION
    {?C [] []}
    FILTER (regex(?C, "CLASSURI","i")) .
}
```

The number of resulting triples constitutes the number of *Class Links* in the dataset.

For *Property Links* we follow a similar process. With the query shown in **Listing 6**, we retrieve all external properties (i.e. properties using a namespace other than the authoritative namespace) that are not explicitly declared as a property but are used: *i)* within a subproperty relation, *ii)* within a property chain, *iii)* in a property restriction, or negative property assertion *iv)* to define a properties' equivalence, disjointedness or inverseness with/to another property, or *v)* to define the domain or range of a class.

Listing 6. SPARQL query used to retrieve external properties.

```
SELECT DISTINCT ?P WHERE {
    {?P rdfs:SubPropertyOf []. } UNION {[] rdfs:SubPropertyOf ?P. } UNION
    {?P owl:propertyChainAxiom []. } UNION
    {[] owl:onProperty ?P. } UNION
    {[] owl:assertionProperty ?P. } UNION
    {?P owl:equivalentProperty []. } UNION {[] owl:equivalentProperty ?P. } UNION
    {?P owl:propertyDisjointWith []. } UNION {[] owl:propertyDisjointWith ?P. } UNION
    {?P owl:inverseOf []. } UNION {[] owl:inverseOf ?P. } UNION
    {?P rdfs:domain []. } UNION
    {?P rdfs:range []. }
    FILTER (!regex(?P, "AUTHORITATIVENAMESPACEURI","i")) .
}
```

For each property URI retrieved through this query, we check its occurrence in the predicate position in any triple in the dataset through the query below.

Listing 7. SPARQL query to check position for each property URI in any triple in a given dataset.

```
SELECT ?P
WHERE {
```

Listing 9. SPARQL query to determine the semantics for *Instance Links*

```
SELECT ?S ?O WHERE {
?S ?P ?O .
FILTER ((?P = owl:sameAs || ?P = owl:differentFrom || ?P = owl:AllDifferent) &&
     (!regex(?S, "AUTHORITATIVENAMESPACEURI","i") || (!regex(?O,
     "AUTHORITATIVENAMESPACEURI","i"))
       }
```

```
    [] ?P [] .
    FILTER (regex(?P, "PROPERTYURI","i")) .
}
```

The number of resulting triples constitutes the number of *Property Links* in the dataset.

4.4.2 Instance Links (ABox Links). Before we can compute the number of *Instance Links* from an individual in the authoritative namespace to any individual in an external namespace, we first need to find all unique individuals in a dataset.

(1) We find all individuals of classes/properties that are declared (i.e., individual that are defined as a type of a class/property).

Listing 8. SPARQL query to retrieve all individuals defined as a type of a class/property.

```
SELECT DISTINCT ?S WHERE { ?S a ?O. }
```

For each retrieved individual, we check if they are defined in the authoritative namespace. If not, they are counted as an *Instance Typing Link*.

(2) We then find all individuals that are reused from a non-authoritative namespace URI in the subject position without being explicitly declared as a type of a class or property. To retrieve those, we first query all triples in the dataset and then check for each unique subject URI that is not in the authoritative namespace, if it is already in the set of declared instances (as of the previous step), or if it is in the set of classes and properties (cf. **Section 4.2**). If it is neither, we count it as an *Instance Link*.

(3) We then follow a similar process for each individual reused from a non-authoritative namespace URI in the object position. For each unique object URI, we check the following conditions: *i)* if the subject is not a blank node, *ii)* the subject URI does not contain the authoritative namespace URI, *iii)* the predicate is not an RDF type relation, and *iv)* the object URI is not already contained within the set of declared instances. If none of these conditions are satisfied, we record it as an *Instance Link*.

For each of these *Instance Links*, we also check if they are explicitly using an owl:sameAs, owl:differentFrom, or owl:AllDifferent relation for the link.

5 COMPUTATION OF LINKS IN PRACTICE

In the following sections, we discuss the results of the analysis of the LOD cloud corpus.

5.1 General characteristics of the LOD corpus

In the first step, we computed general statistics of the datasets in the LOD cloud (cf. **Table 4**). The first observation we can make is that the majority of the datasets are rather small in size, that is, 50% of all datasets have less than 4,478 triples. Although the mean (17,860,436 triples) is much larger, it is skewed by some few much larger datasets (e.g., DBpedia, the Zeitschriftendatenbank dataset, the WebIsA dataset, and the catalogue of the German National Library).

On average, the number of subjects is about an order of magnitude smaller than the number of triples, implying that there are on average 10 statements made about each subject. The mean number of unique predicates is interestingly very small — only 31 unique predicates (including RDF(S) and OWL predicates) are used in each dataset. Again, the mean is larger, but is, in fact, largely skewed by just one dataset, namely DBpedia with 68,687 unique predicates. The dataset defining the second most predicates, the B3Kat dataset, has only 3,259 unique predicates. The large and unusual number of predicates in DBpedia can be explained by the automated generation of its triples and a lack of reconciliation of similar properties with slightly different names (labels). Not surprisingly, the average number of unique objects in Linked Datasets is larger than the number of unique subjects. This is an indication of the existence of links between datasets (i.e., the reuse of objects). However, again the mean is much larger — more than three orders of magnitude larger than the median. This is again due to some few large datasets, in particular, the Zeitschriftendatenbank, DBpedia, the WebIsA database and the catalogue of the German National Library.

	Median	**Mean**
Number of Triples	4,478	17,860,436
Number of Unique Subjects	613	1,774,578
Number of Unique Predicates	31	455
Number of Unique Objects	2,245	5,296,390

Table 4. General statistics of the corpus

5.2 Ontology Links

Our analysis of *Ontology Links* in the corpus revealed some interesting usage patterns of ontologies. However, before we discuss the number of *Ontology Links* we present some general statistics on the use of classes and properties in the LOD cloud, which are shown in **Table 5**:

	Median	**Mean**
Number of Declared Classes	0	52
Number of Undeclared Classes:	7	54
Number of Declared Properties:	0	550
Number of Undeclared Properties:	24	226

Table 5. General statistics on the use of classes and properties in the LOD cloud

Not surprisingly, the median number of declared classes and properties for Linked Datasets is 0. In fact, 67% of all datasets do not declare any classes or properties. In terms of undeclared classes, we can see that 50% of all datasets reuse at least 7 classes, while the average number of reused classes is 54. All, but three datasets, include at least one reused class (which for some datasets is just an owl:Class or rdfs:Class).

We also compared the resulting class URIs for each dataset to the class URIs retrieved from `prefix.cc` to check how many classes in our corpus are not registered on `prefix.cc`. Out of 36,970 unique classes used, in total, in our corpus, 20,217 classes are not registered. The low number of registered classes on `prefix.cc` is quite surprising, given that its corpus includes 204,616 classes. Although we can not determine the individual ontology namespace from a class URI, if we group those class URIs by their Pay Level Domains (PLDs), we end up with only 135 “ontology” PLDs

that are not registered with prefix.cc. These PLDs account for the total number of unregistered classes. The top ten of these PLDs are listed in **Table 6**.

PLD	# of class URIs
dbpedia.org	10,579
sli.uvigo.gal	1,818
semanticscience.org	1,427
purl.org	1020
www.productontology.org	990
purl.obolibrary.org	855
semanticweb.org	809
minsky.gsi.dit.upm.es	714
wikidata.dbpedia.org	455
www.wikidata.org	219

Table 6. PLDs with the highest number of unregistered class URIs

PLD	# of property URIs
sw.opencyc.org	54,916
umbel.org	27,919
dbpedia.org	10,591
www.orpha.net	6,198
purl.obolibrary.org	5,609
www.ebi.ac.uk	4,712
onto.beef.org.pl	2,369
sli.uvigo.gal	1,818
semanticscience.org	1,429
purl.org	1,239

Table 7. PLDs with the highest number of unregistered property URIs

Some of these point to the use of deprecated or wrong URIs in Linked Datasets. For example, the complete set of class URIs for dbpedia.org is registered with prefix.cc and therefore the class URIs used in Linked Datasets not registered are either wrong or deprecated which is confirmed by our analysis of the existence of these URIs in Section 5.2.6. On the other hand, there are no ontologies registered in prefix.cc for the sli.uvigo.gal, the semanticscience.org, and surprisingly, for the www.productontology.org namespace.

Repeating the process for property URIs shows that the ratio of unregistered unique properties in prefix.cc is much bigger even than for class URIs (cf. Table 7). Namely, out of 142,694 unique properties used, 141,943 are not registered with prefix.cc. Again, a large part of these unregistered property URIs are, in fact, broken URIs (cf. Section 5.2.6). However, analysing again the PLDs of those URIs that are not registered with prefix.cc we end up with 160 PLDs, few examples of which are listed in **Table 7**. The largest number are from sw.opencyc.org and umbel.org, both of which are not registered in their entirety with prefix.cc.

Table 8 and **Table 9** show the most commonly used class and property URIs (other than RDF-S/OWL URIs) in datasets in our corpus, respectively.

Class URI	Number of datasets
http://rdfs.org/ns/void#Dataset	118
http://rdfs.org/ns/void#Linkset	90
http://xmlns.com/foaf/0.1/Person	74
http://persistence.uni-leipzig.org/nlp2rdf/ontologies/nif-core#Word	65
http://persistence.uni-leipzig.org/nlp2rdf/ontologies/nif-core#Sentence	64
http://www.w3.org/2004/02/skos/core#Concept	56
http://xmlns.com/foaf/0.1/Organization	51
http://vivoweb.org/ontology/core#CoreLaboratory	30
http://vivoweb.org/ontology/core#Center	28
http://xmlns.com/foaf/0.1/Agent	24

Table 8. Number of datasets that use a specific class URI

Property URI	Number of datasets
http://purl.org/dc/terms/title	163
http://purl.org/dc/terms/creator	140
http://purl.org/dc/terms/description	134
http://xmlns.com/foaf/0.1/homepage	125
http://purl.org/dc/terms/publisher	112
http://purl.org/dc/terms/subject	105
http://rdfs.org/ns/void#vocabulary	103
http://purl.org/dc/terms/modified	98
http://rdfs.org/ns/void#exampleResource	96
http://rdfs.org/ns/void#subset	88

Table 9. Number of datasets that use a specific property URI

Median:	0
Mean:	1,299
Proportion above 0:	44%

Table 10. Class Links Statistics

5.2.1 Class Links. Only a few datasets include *Class Links*, which is not particularly surprising, considering the low number of declared classes in datasets in the corpus. However, 44% of all datasets link to classes outside of the authoritative namespace. This is ≈ 10% points more than datasets declaring classes, which points to the reuse of external classes in terminological axioms. The mean number of *Class Links* with 1,299 triples is largely influenced by the top ranked authoritative namespaces `http://vivo.iu.edu` with 119,358 links and `http://vivo.scripps.edu` with 63,128, both more than two orders of magnitude larger than the 10^{th} ranked namespace, `http://vivoweb.org` with 847 *Class Links*. The distribution of *Class Links* per the size of the dataset is shown in **Figure 1**. The distribution shows that a large proportion of authoritative namespaces include between 10 and 1,000 *Class Links*, regardless of the size of the dataset.

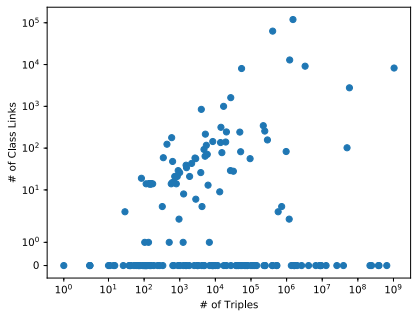

Fig. 1. Class Links per # of Triples

http://vivo.iu.edu	119,538
http://vivo.scripps.edu	63,128
http://www.imagesnippets.com	12,874
http://core.kmi.open.ac.uk	9,143
http://commons.wikimedia.org	8,258
http://vivo.psm.edu	8,036
http://datos.bne.es	2,778
http://dbpedia.org	1,614
http://www.productontology.org	1,000
http://vivoweb.org	847

Table 11. Authoritative namespaces with most Class Links

5.2.2 Property Links. Very few authoritative namespaces use *Property Links* (only 18%). The maximum with 4,995 such links in `http://commons.wikimedia.org` is more than two orders of magnitudes larger than the 10^{th} ranked namespace, `http://tkm.kiom.re.kr` with 60. Although slightly linearly correlated to the size of the datasets, the majority of authoritative namespaces that include

Property Links use between 10 and 1,000 of those (**Table 2**). As with *Class Links*, one would expect *Property Links* mostly in ontologies, and therefore the low number of such links in our corpus is, in fact, a positive sign of the reuse of ontologies, rather than the redefinition/extension of properties in the local dataset namespace.

Median:	0
Mean:	47
Proportion above 0:	18%

Table 12. *Property Links* Statistics

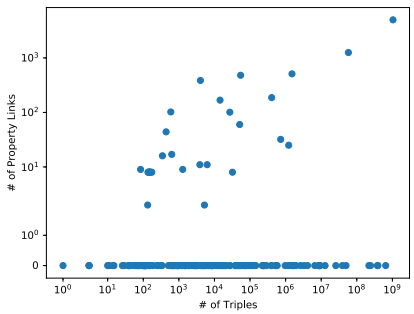

Fig. 2. *Property Links* per # of Triples

`http://commons.wikimedia.org`	4,995
`http://datos.bne.es`	1,255
`http://vivo.iu.edu`	510
`http://vivo.psm.edu`	481
`http://vivoweb.org`	386
`http://vivo.scripps.edu`	187
`http://semanticscience.org`	168
`http://www.iupac.org`	102
`http://dbpedia.org`	101
`http://tkm.kiom.re.kr`	60

Table 13. Authoritative Namespaces with most *Property Links*

5.2.3 Instance Typing Links. With a median of 206 and a mean of 1,967,570 *Instance Typing Links* per authoritative namespace are the most common link type in the datasets in our corpus and for most datasets represent a large portion of the overall number of links **cf. Figure 4**). Consequently, we can observe a strong linear correlation between the number of triples and the number of *Instance Typing Links* (cf. **Figure 3**). All, except eight datasets, use external classes to type individuals in the authoritative namespace. Unsurprisingly, the authoritative namespace with the most such links is `http://webisa.webdatacommons.org` (cf. **Table 16**), as its purpose is to define IsA relations for hypernymy relations extracted from the Common Crawl.

Median:	206
Mean:	1,967,570
Proportion above 0:	97%

Table 14. *Instance Typing Links* Count

Median:	11
Mean:	108
Proportion above 0:	97%

Table 15. Distinct class URIs in *Instance Typing Links*

We also analysed the distinct external class URIs used in such links (cf. Table 17). The median is a relatively high 11 external classes used and the mean is 108. `http://commons.wikimedia.org` is the authoritative namespace with the most distinct external classes (mostly from the DBpedia ontology namespace) used for typing individuals with 3,197 in total.

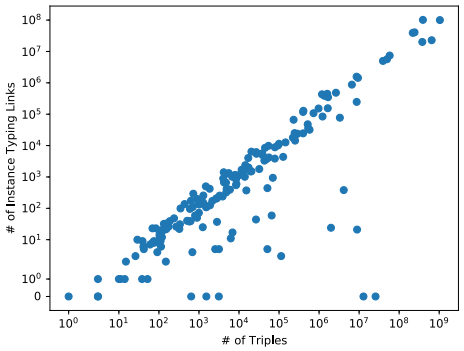

Fig. 3. *Instance Typing Links* per # of Triples

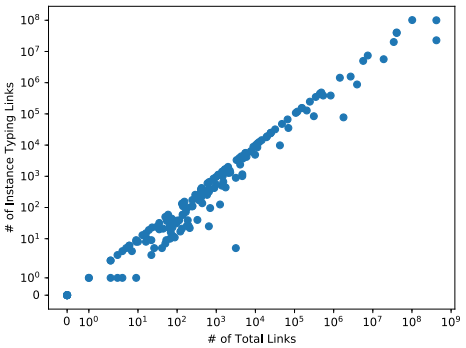

Fig. 4. *Instance Typing Links* per # of Links

http://webisa.webdatacommons.org	101,491,507	http://commons.wikimedia.org	3,197
http://commons.wikimedia.org	100,022,186	http://sli.uvigo.gal	1,830
http://lod.b3kat.de	40,674,519	http://semanticscience.org	1,595
http://lod.hebis.de	39,160,423	http://data.tharawat-magazine.com	1115
http://d-nb.info	20,096,228	http://www.productontology.org	1005
http://datos.bne.es	7,419,630	http://dbpedia.org	756
http://data.ordnancesurvey.co.uk	5,653,997	http://minsky.gsi.dit.upm.es	740
http://data.europeana.eu	4,987,332	http://semanticweb.org	724
http://id.loc.gov	1,570,877	http://www.imagesnippets.com	720
http://data.bibsys.no	1,440,011	http://data.wordlift.it	649

Table 16. Authoritative namespaces with most `Instance Typing Links`

Table 17. Authoritative Namespaces with most distinct class URIs used in *Instance Typing Links*

5.2.4 Instance Links. Our analysis of the LOD cloud shows that there are relatively few *Instance Links* defined in Linked Datasets. In fact, 28% of all datasets do not include any link from any individual in the authoritative namespace to any other individual in an external namespace, either in the subject or object position. The mean number of links with 1,984,955 is highly skewed by the top two ranked authoritative namespaces which are listed in Table 19, with the median being a mere 24 *Instance Links*, while the 90^{th} percentile is still only 3,863.

Median:	206
Mean:	4,240,890
Proportion above 0:	72%
90^{th} percentile:	3,863%

Table 18. Instance Links Count

The authoritative namespaces with the most *Instance Links* are `http://ld.zdb-services.de` and `http://commons.wikimedia.org`, while the 12^{th} ranked `http://data.coi.cz` already uses four orders of magnitude fewer links. Although there is a slight linear correlation between the size of the dataset and the number of *Instance Links* (cf. **Figure 5**), there is a large cluster of authoritative namespaces that only uses between 10 and 10,000 *Instance Links*.

Looking at some specific predicate types that are used in those links we can see that the often considered popular owl:sameAs link is not particularly widely used. In fact, it is only used in

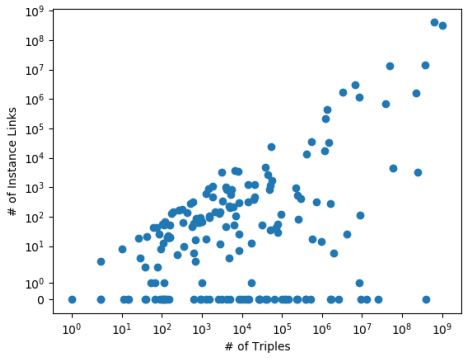

Fig. 5. Instance Links per # of Triples

http://ld.zdb-services.de	398,381,851
http://commons.wikimedia.org	319,988,690
http://d-nb.info	14,160,649
http://data.ordnancesurvey.co.uk	13,277,718
https://data.gov.cz	3,081,559
http://core.kmi.open.ac.uk	1,696,618
http://lod.hebis.de	1,624,579
http://id.loc.gov	1,143,545
http://data.europeana.eu	687,735
http://spraakbanken.gu.se	451,081
http://www.imagesnippets.com	214,362
http://data.coi.cz	34,277

Table 19. Authoritative namespaces with most Instance Links

53% of all datasets, while some few authoritative namespaces, in particular, http://commons.wikimedia.org (linking mostly to http://dbpedia.org/resource) and some of the authoritative namespaces of the German library community (i.e., http://ld.zdb-services.de, http://d-nb.info, http://lod.b3kat.de and http://lod.hebis.de) account for a large part of the mean number of owl:sameAs links of 503,859. The owl:differentFrom predicate is only used by one authoritative namespace, again http://commons.wikimedia.org, while owl:allDifferent is not used in any dataset to link an individual in the authoritative namespace to an external individual. The rdfs:seeAlso relation is used slightly more often, but it is again http://commons.wikimedia.org that uses it extensively (to link to http://dbpedia.org/resource), whereas the third ranked http://data.nobelprize.org includes only 5,827 *Instance Links* using the rdfs:seeAlso predicate.

	owl:sameAs	owl:DifferentFrom	rdfs:seeAlso	owl:AllDifferent
Median	0	0	0	0
Mean	503,859	581	2,735	0
Proportion > 0	53%	<1%	14%	0
90th Percentile	1,460	0	1	0
1st	http://commons.wikimedia.org			N/A
1st #	40,636,493	103,439	324,659	
2nd	http://ld.zdb-services.de	N/A	http://stitch.cs.vu.nl	N/A
2nd #	18,049,155	N/A	153,699	N/A
3rd	http://d-nb.info	N/A	http://data.nobelprize.org	N/A
3rd #	17,410,586	N/A	5,827	N/A

Table 20. Selected usage of predicates for linking

5.2.5 Total Number of Links. In **Table 21** some statistics on the total number of links per authoritative namespace are presented. There is a strong linear correlation between the number of triples and the total number of links in the authoritative namespace. However, since the number of *Instance Typing Links* per authoritative namespace is by far the largest, while also showing a strong linear correlation, this result is not surprising. Surprisingly, though, 4% of all authoritative namespaces do not use any link type to an external namespace. The namespaces with the most number of total links are http://ld.zdb-services.de and http://commons.wikimedia.org.

5.2.6 Detailed analysis of Link Quality – Broken Links.

Median:	416
Mean:	6,209,808
Proportion above 0:	96%

Table 21. Total Links Count

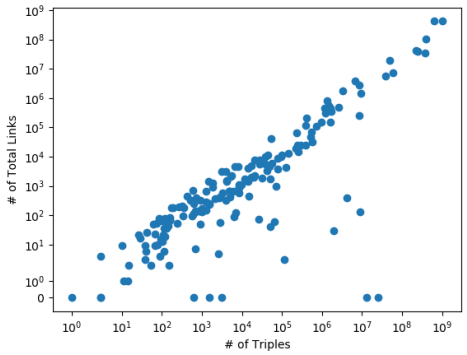

Fig. 6. Total Links per # of Triples

http://ld.zdb-services.de	421,206,061
http://commons.wikimedia.org	420,024,129
http://webisa.webdatacommons.org	101,491,507
http://lod.hebis.de	40,785,002
http://lod.b3kat.de	40,677,795
http://d-nb.info	34,256,877
http://data.ordnancesurvey.co.uk	18,931,817
http://datos.bne.es	7,428,111
http://data.europeana.eu	5,675,067
https://data.gov.cz	3,958,043

Table 22. Authoritative namespaces with highest number of links

Broken Ontology Links. Our *Class Link* and *Property Link* analysis was performed in two steps. First, we checked all retrieved class and property URIs from our prefix.cc crawl, i.e. 204,616 class URIs and 1,821 property URIs. Then we performed an analysis of all unregistered class and property URIs, i.e. 20,217 class URIs and 141,943 property URIs respectively, from our class link and propery link analysis of our corpus of 430 dataset. For checking the status of the URIs, we performed an HTTP HEAD method call on the class or property URI using the Python requests library with the timeout set at one second. For all URIs that timed out without an HTTP 408 response code, we repeated the call 10 times over the period of two weeks. If it was still not responding after those calls we deemed the resource unavailable.

Table 23 shows the results of the class URI analysis. Of the total 204,616 class URIs retrieved from prefix.cc 146,145 actually belonged to one namespace, i.e. http://ncicb.nci.nih.gov/xml/owl/EVS/Thesaurus.owl#, none of which resolved. Therefore, we removed those class URIs from our analysis and used the remaining number of class URIs (i.e. 58,471) as the base for our ratio calculations in Table 23. Despite removing this large number of class URIs from one namespace, we end up with a concerningly small number (i.e. 12.3%) of available URIs, i.e. URIs with a 200 HTTP response code. There was a larger number of 303 response codes (i.e. 21.9%) which, as a standard way of implementing Cool URIs to redirect from a resource identifier to a URL of a document that represents the resource, also indicate the availability of the class URI. However for the large number of 301 and 302 codes it is unclear if the resource is actually available. Although 301 and 302 codes point to an alternate location which may still indicate the existence of the resource, but with a different identifier, the new resource that is redirected to may not be the same as the old one. Although it is infeasible to check all these URIs, a manual check of a sample showed that a large part of these redirects are still pointing to an RDF resource. However, a not insignificant number of 301 and 302 redirects are just to a generic webpage. We can therefore conclude that 34.2% of all class URIs (the sum of HTTP code 200 and 303) are in most likelihood available, while a further 39.2% are partially available which leaves us with more than a quarter of all class URIs that are

either not available (i.e. return a 40x response code), i.e. 20.6%, or where no response was received after repeated attempts, i.e. 5.9%.

Analysing the class URIs retrieved from the *Class Links* in our corpus, the picture looks even more bleak. While with 12.8% of the total a similar number of URIs return an HTTP 200 response code, and a further 19.3% of all URIs respond with a 303 code, indicating a Cool URI implementation, more than half of all class URIs, and as such *Class Links* in our corpus, are actually broken, i.e. either return a 40x or 50x response code or timed out repeatedly.

Looking at the PLDs of the URIs that are not working, there is only one that is responsible for more than a couple of dozen failed links, i.e. namely 795, and that is `http://semanticweb.org/`. Sadly, our own community website and its RDF content has not been available for several years already. The majority of the remaining broken class URIs are links to DBpedia classes that are either not available or that have been incorrectly spelled. The larger part of these errors seem to stem from automated linking tools or hard-coded mapping rules. For example, there is a large part of class URIs that seem to have been generated from Wikipedia category pages, such as `http://dbpedia.org/class/yago/2010DisastersInTheUnitedStates` or `http://dbpedia.org/class/yago/1960sAutomobiles`, both of which have category pages in Wikipedia (i.e. `https://en.wikipedia.org/wiki/Category:2010_natural_disasters_in_the_United_States` and `https://en.wikipedia.org/wiki/Category:1960s_automobiles`, respectively), but clearly no class URI, as they would be represented as lists of entitities. Errors in URIs come in many shapes and forms, ranging from commas, brackets and blank spaces in URIs to encoding issues with special characters in languages other than English. However, these errors account for less than 3% of all broken class URIs.

Broken class URIs				
	prefix.cc crawl		LOD cloud corpus	
HTTP Response Code	#	% of Total	#	% of Total
200	7,175	12.3%	2,579	12.8%
301	18,598	31.8%	2,610	12.9%
302	4,331	7.4%	925	0.5%
303	12,805	21.9%	3,903	19.3%
40x	12,054	20.6%	8,664	42.9%
50x	66	<0.1%	111	<0.1%
No response	3,442 (146,145)	5.9%	1,425	7%
Total	58,471 (204,616)	100%	20,217	100%

Table 23. Type and # of response codes to HTTP Header requests for class URIs

Table 24 shows the results of checking dereferenceability of property URIs. While the results for property URIs retrieved from `prefix.cc` shows slightly better results than for class URIs, there is still a large number of 301 and 302 response codes (i.e. 35%). And although marginally less than for class URIs, still 14.3% of property URIs are not available anymore, either because of a 40x or 50x response code or because of repeated time-outs.

While there is a much larger number of unregistered property URIs in our corpus that respond with a 200 HTTP response code (i.e. 40.9%) than for class URIs, the number of property URIs that are broken or not accessible (i.e. 53.6%) is even larger. While there are no significant PLDs that are responsible for more than a few broken property URIs, the majority of broken property URIs originate from links to DBpedia properties. Some of those look legitimate like `http://dbpedia.`

org/property/typeOfPlace. While the property "typeOfPlace" does not exists in DBpedia, we do not know if this property has existed before.

Similar to *Class Links* and the resulting class URIs, also a large portion of *Property Links* seem to have been created through link generation frameworks, resulting in broken links such as http://dbpedia.org/property/trusteePresident or http://dbpedia.org/property/authorMask1. There are, however, also a significant number of DBpedia URIs that work, but which are clearly errors introduced through some internal mappings within DBpedia, e.g. http://dbpedia.org/property/pushpinMapBruhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhh or http://dbpedia.org/property/purposeRepresentationOfFloristsToGovernmentAndThePublic, neither of which is registered in prefix.cc, but which are nonetheless accessible and are counted within the number of HTTP 200 responses.

Broken property URIs				
	prefix.cc crawl		LOD cloud corpus	
HTTP Response Code	#	% of Total	#	% of Total
200	814	44.7%	58,108	40.9%
301	442	24.3%	1,137	0.8%
302	194	10.7%	1,391	1.0%
303	108	5.9%	5,247	3.7%
40x	130	7.1%	73,366	51.7%
50x	4	<0.1%	362	0.3%
No response	129	7.1%	2,332	1.6%
Total	1,821	100%	141,943	100%

Table 24. Type and # of response codes to HTTP Header requests for property URIs

Summarizing, about half of all class and property URIs introduced through *Class Links* and *Property Links* in our corpus are broken, while even a quarter of the registered class and property URIs on prefix.cc are broken too. While many broken links seem to stem from automated link generation not considering (i) special characters, (ii) translating categories into class URIs that should rather be named lists or (iii) links to URIs that have existed previously, this large number of broken TBox links raises the question of the practicality of the use of distributed ontologies in Linked Data.

Broken Instance Links. A full account of checking broken *Instance Links* in terms of checking dereferenceability of all mentioned *Instance Links* within each dataset is beyond scope, as it would require millions of HTTP lookups. However, we have conducted different sub-analyses with different strategies to investigate broken *Instance Links*, focusing on the analysis of instance namespaces. To this end, we have chosen the in-links of DBpedia on an instance level as a proxy, which we analyse over the whole corpus, by checking its instance namespace (cf. Def. 2.10), http://dbpedia.org/resource/, (dbr:). We note that such an experiment could be conducted analogously, for each authoritative *instance namespace*.

The analysis involves extracting, for each of the datasets in our corpus, the set of URLs in the dbr: namespace, and compare them with those also mentioned in $G_{dbpedia}$ – any url u in the dbr: namespace not occurring in $G_{dbpedia}$ hints to a broken instance link. Note that, strictly speaking, we cannot exclude misuse of instance URLs, i.e. such instance URLs being used by other datasets in non-instance positions, but we leave such an in-depth investigation to future work, and for the moment work under the simplifying assumption that all references to an instance namespace of another dataset are indeed *Instance Links*.

Again, HDT helps us to perform this analysis in a scalable manner, by allowing to extract a namespace-filtered version of its dictionary per (HDT dump of the) dataset. Overall, 145 out of 430 datasets in total contain dbr: URLs. 11,696 of these URLs do not occur within the roughly 26M dbr: URLs in $G_{DBpedia}$, hinting to a lower bound (disregarding duplicate usage of erroneous URLs), of at least 11k broken *Instance Links* to DBpedia. By analysing these per dataset, we can see that the number of such broken links can become a potentially significant issue: as Table 25 shows, several datasets use thousands of broken dbr: URLs.

A closer look into the most common broken dbr: URLs (Table 26) suggests, that many of these stem from URL encoding of special characters: indeed, DBpedia itself, while not exposing those URLs explicitly in their dump, redirect most of the URLs using URL-encoding for special characters correctly, for example http://dbpedia.org/resource/C%C3%B4te_d%27Ivoire is redirected correctly to http://dbpedia.org/page/Ivory_Coast, by the triple

```
dbr:Côte_d'Ivoire dbo:wikiPageRedirects dbr:Ivory_Coast .
```

in DBpedia. However, this automated resolution of URL-encodings of special characters according to RFC3986 [45] is not explicit in the export, i.e., there are no triples like

```
dbr:C%C3%B4te_d%27Ivoire dbo:wikiPageRedirects dbr:Côte-d'Ivoire .
```

in DBpedia. However, since providing explicit (owl:sameAs or dbo:wikiPageRedirects) links between all possible combinations is obviously infeasible (e.g., it is unclear where to stop, i.e. should variations like dbr:C%C3%B4te_d'Ivoire dbr:Côte_d%27Ivoire also be considered?), it seems to be advisable for dataset providers to not use special (non-ASCII) characters for minting URIs, or, likewise, for consumers to check whether the dataset they link to provides Unicode-URIs directly or uses ASCII-encoded escaped special characters in URLs. In our experiment, fixing synonyms with single quote characters seems to have a big effect on the ieee.rkbexplorer.com dataset, plus fixing synonyms in URLs for escaped quotes, for instance, seem to have a big effect on the kasabi.com_dataset_discogs dataset, where it drastically reduces the number of errors.

Another problem relates to the different language versions of DBpedia; for instance data.persee.fr uses a lot of French Wikipedia names that are neither present in nor redirected to the English Wikipedia (nor the DBpedia export): e.g. https://fr.wikipedia.org/wiki/G%C3%A9rard_Maarek has no corresponding URL dbr:G%C3%A9rard_Maarek, nor dbr:Gérard_Maarek (after resolving escaped characters).

Lastly, the dataset linked.opendata.cz apparently tries to import the whole coding system of https://en.wikipedia.org/wiki/Anatomical_Therapeutic_Chemical_Classification_System into DBpedia URLs, which, however is not represented in this depth in the DBpedia dataset: e.g., dbr:ATC_code_M01AE52 referenced in this dataset is not present in DBpedia itself, while this sub-code is mentioned on the respective DBpedia page corresponding to dbr:ATC_code_M01, a redirects links only exist for the next level of this hierarchy, i.e.,

```
dbr:ATC_code_M01AE dbo:wikiPageRedirects dbr:ATC_code_M01.
```

Summarizing, many issues about broken *Instance Links* into DBpedia seem to stem from automated link generation not considering (i) special characters, (ii) missing cross-language links between multi-lingual Wikipedias, or (iii) translating hierarchical coding schemes into DBpedia URLs that are not completely covered.

sw.opencyc.org	4511
ieee.rkbexplorer.com	2084
data.persee.fr	2083
linked.opendata.cz	1176
kasabi.com_dataset_discogs	759

Table 25. Broken *Instance Links*: datasets in terms of number of broken dbr: outlinks – top 5

dbr:C%C3%B4te_d%27Ivoire	3
dbr:People%27s_Republic_of_China	3
dbr:Location_%28geography%29	3
dbr:Washington%2C_D.C.	3
dbr:Ge'ez_language	3
dbr:Eugene_O'Neill	3
dbr:L'OrÃľal	3
dbr:McDonald's	3
dbr:Farmers'_market	3
dbr:Course_%28education%29	3

Table 26. A list of broken DBpedia instance URLs (ordered by in how many datasets they appear) – top 10

6 DISCUSSION

There are several observations from our analysis of the Linked Open Data cloud corpus that are worth discussing.

Ontologies are reused widely: With 36,970 classes and 142,694 properties reused in authoritative namespaces in our corpus, the popularity of ontologies can not be denied. Also, while there is a relative lack of Instance (ABox) links, external classes are used extensively to type individuals in the authoritative namespace of datasets in our corpus. Only a few datasets define their own ontology or extend/narrow the semantics of classes and properties of external ontologies. This is a sign that: 1) dataset publishers follow best practices and separate the ontology namespace from the authoritative namespace of the dataset, and 2) it is also a sign that there exists a large number of ontologies that cover already many domains that can be readily reused.

Need for ontology publishing best practices: As our analysis showed, many ontology namespaces, and as such, their classes and properties are not registered on prefix.cc. Even if they are registered, their historical namespace and/or deprecated class and property URIs are often not available anymore. Although there are attempts to establish domain-specific ontology repositories (e.g. BioPortal [52]) and general domain ontology repositories (i.e. the LOV portal [47]), an authoritative ontology register and a persistence mechanism beyond prefix.cc is missing. Such a mechanism should assign a DOI to an ontology and persist the document itself in perpetuity (attributes offered by portals such as zenodo.org), but also register its authoritative namespace(s), preferred authoritative prefix and resolve its class URIs and property URIs in perpetuity. While the latter are partly covered by using prefix.cc in combination with https://w3id.org or http://purl.org, a repository and mechanism offering all these features is lacking.

Ubiquity of broken Class and Property links: While our analysis shows that the datasets in our corpus include a good number of ontology links, in particular, *Instance Typing Links*, our dereferenceability analysis of the URIs used in those links shows an alarming number of broken links, i.e. more than half of all class URIs and property URIs were broken. Some of these broken links can be explained by the inaccuracy of link generation tools that generated those links, but

there is also an issue with the long-term availability of some of the ontologies that are linked from datasets. While ontologies and Linked Open datasets are built in a truly decentralised manner, companies and organisations still need to trust the publisher when reusing a digital asset on the Web. As the analysis shows, many of these publishers of ontologies seem to be unable to guarantee availability of the resource in perpetuity. Therefore data publishers relying on these resources need to consider to replicate the ontology and the URIs contained within in a storage location that they have control over or that can be guaranteed in the long term.

Lack of ABox Links: Many (28% of all) datasets do not use any *Instance Links*. Although this number is significantly higher than the results reported in earlier work on samples of the Linked Open Data Cloud (i.e., 56%) [41], together with the median number of *Instance Links* (i.e. 206) this is still disappointingly low. Furthermore, the authoritative namespaces that actually do use *Instance Links* use mostly other predicates than owl:sameAs relations, that were thought of as the most popular relations for linking [41], while also being the relation that is most useful to reconcile similar individuals in different datasets. The lack of *Instance Links* can be explained by several factors: 1) these links are expensive to establish manually 2) expensive to maintain, and 3) even if they exist, there is no incentive to publish them openly. Evidence that these factors play a large part in explaining the relative lack of *Instance Links* is the fact that datasets (other than the community-built DBpedia) that do include *Instance Links* are largely from the GLAM sector (i.e. Galleries, Libraries, Archives, and Museums) where there is a strong community that follows standardised publishing principles and where data is largely historic and static, i.e. once a link is established, it does not have to be updated, ever again.

Lack of and incorrect namespace declarations: Only 59% of all datasets in our corpus publish their namespace in the LOD cloud metadata, and of those 257 that do, only 162 match the namespace that we obtain through an analysis of the triples in the graph (cf. **Section 4.3**). Although based on a rigorous analysis of the triples in a dataset, our algorithm may not always choose the correct authoritative namespace. However, as discussed in **Section 4.3**, if available, the namespace in the metadata is incorrect in many cases, as there are no guidelines or best-practices what actually constitutes a namespace in a Linked Dataset. With the definitions in this paper, we hope to provide both the necessary rigour but also a tool for future data publishers to be able to publish the authoritative namespace of a Linked Dataset.

Plethora of data and metadata formats: Running analyses like the ones we did in our paper might seem tedious, but we argue that one of the main reasons for this is the heterogeneity of publication formats, used in Linked Data. Downloading and converting files from different RDF serialisations into HDT, potentially involving parse errors, constituted a major part of the effort used for our experiments. Once each dataset node/dump had been converted to HDT though, the analysis was easy: as we have shown, link computations can be done at scale on even large datasets in HDT, and due to the extensible header format of HDT, the respective metadata about links and authoritative namespaces per dataset can be easily published and computed in place at HDT generation time. As one of our insights, we therefore recommend:

- to make a published dataset available as one file in the HDT format,
- along with the respective meta-data, directly in the HDT header.

Having an HDT dump generated this way with up-to-date link statistics and namespace metadata in place, dereferenceable at the namespace URL, could potentially solve issues with other publication methods:

(a) as shown in prior analysis [36, 48] and again confirmed in this paper, for instance SPARQL endpoints are an unreliable access point for Linked Data. Also, for most larger datasets, many typical queries (such as the exploratory queries used in our analysis) time out and as such do

not provide a result. HDT [18] as a scalable mechanism to reuse and analyze Linked Datasets published on the Web, can circumvent many of these issues: firstly, HDT requires far less resources than running a SPARQL endpoint for maintenance on the publisher side; secondly, Triple Pattern Fragments [49] servers are readily available as an interface for HDT, and gaining more attention supporting lightweight querying that balances query processing between clients and servers.

(b) further, for datasets, as is common best practice for ontologies, the authoritative namespace of the data contained within should be published in its metadata. The `void:uriSpace` property offered by the VoID vocabulary is a suitable property to do so. Link statistics that so far could have been provided manually using `void:Linksets` can be computed directly using the HDT link analysis script developed herein and readily available at `https://github.com/arminhaller/LinksInLOD`.

Summarizing, HDT and the namespace authority and link analysis annotations published/linked from the namespace URI provide a simple and effective publishing principle that potentially enables an easier findable/accessible, interoperable and directly reusable way of publishing FAIR, interlinked knowledge graphs.

7 CONCLUSION & FUTURE WORK

In this paper, we critically and systematically assessed the network of knowledge graphs available and accessible as Linked Data, in terms of analyzing the most critical quality aspect of a true "network" of open interconnected knowledge graphs: **links**. We first proposed a rigorous definition of a naming authority for a Linked Dataset. This definition of an authoritative namespace allows us to distinguish internal references within a dataset from links to data defined in an external namespace. Consequently, we provided concrete definitions of links between datasets, distinguishing between *Ontology (TBox) Links* and *Instance (ABox) Links*.

We presented automated methods to analyze different link types at scale, and provided an empirical analysis of linkage and the quality of those links among the datasets of the Linked Open Data (LOD) cloud. For this analysis we established a corpus of classes and properties defined within our corpus and within ontologies registered on `prefix.cc`. This ontology corpus allowed us to distinguish TBox links from ABox links.

In our current implementation we consider only the Pay Level Domain (PLD) of the authoritative namespace. This assumption excludes, for example, links from a data repository stored in a hosting service such as Github to another data repository hosted in the same repository, since its PLD is the same. In future work we could compute the links for a path structure after the PLD and compare it to the links we computed through our simplified method here. However, we do not expect a significant difference between the two.

For our analysis of the LOD cloud we disregard any literal objects in a triple. We do not analyse the datatype of literals and therefore miss custom-typed literals (i.e., literals using another type than the XML Schema datatypes). According to our definition, a custom-typed literal using a URI external to the authoritative namespace is considered a link. However, custom datatypes in RDF have only recently been given attention and are not supported yet by most reasoners [30].

Our definition of *Class Links* (and analogously *Property Links*) does not require the URI in the subject or object to be in class position. Strictly speaking, that definition includes triples that implement punning on a class or role level and ontology hijacking Hogan et al. [23]. Our analysis of the LOD cloud includes those links, but we are not checking the correctness of those triples, i.e. if these are indeed intended puns (pun intended) or if these are, in fact, errors. We also do not distinguish between authoritative and non-authoritative TBox links [22], i.e. we count them as links, regardless. In future work we intend to check non-authoritative URIs in class positions and analyse if they are 1) classes, 2) hijacked classes, or 3) correctly punned individuals.

The analysis of a corpus of 430 datasets from the LOD cloud showed that almost all datasets use external ontologies for the typing of individuals, i.e. *Instance Typing Links*, while links on the data level, i.e. *Instance Links*, are relatively sparse, with a median number of such links of 206 per authoritative namespace. Also, only 72% of all authoritative namespaces include links to other individuals at all, either in the subject or object position of a triple. The previously thought to be popular of owl:sameAs relations are, in fact, only used in 53% of all datasets. Although this low number and quality of links between datasets on the ABox level is concerning and somehow undermines the idea of Linked Data, the number and quality of links on the TBox level is promising. It shows a strong propensity of reuse of classes and properties defined in ontologies on the Web.

However, our analysis on the dereferenceability of *all Ontology Links* showed that about half of all class URIs and property URIs introduced through *Class Links* and *Property Links* in our corpus are broken. While a full account of checking dereferenceability of all *Instance Links* within each dataset in our corpus was beyond scope, as it would require millions of HTTP lookups, the issues that we encountered when analysing a sample, i.e. links into DBpedia, were similar for broken *Instance Links* and broken ontology links, namely (i) mistreatment of special characters, (iii) translation of lists and hierarchical coding schemes into URIs that either do not exist or are wrong, and (iii) the use of links to URIs that have existed previously, but that have since become unavailable.

To better enable reusability and findability of data and to ease linking to existing resources, one way to address the problem of broken links would be to make datasets available in dump formats such as HDT (rather than purely rely on Cool URIs) which should – in our opinion – enable consumers to make informed decisions to reuse data more effectively for the following reasons: (1) HDT allows users to locate and download the dump in one file and process the data in an efficient manner without the need to decompress it (2) HDT enables the provision of dataset namespace authority metadate and computation of link statistics metadata, published along with and in sync with the dump in one file.

ACKNOWLEDGEMENTS

The authors would like to acknowledge Jeremy Debattista for his support with the LOD cloud corpus. This work is supported in part by the European Union's Horizon 2020 research and innovation programme under grant 731601 (SPECIAL) and by the Austrian Research Promotion Agency (FFG): grant no. 861213 (CitySPIN).

REFERENCES

[1] Andrejs Abele, John P McCrae, Paul Buitelaar, Anja Jentzsch, and Richard Cyganiak. 2017. *Linking open data cloud diagram 2017.* URL: http://lod-cloud.net (Accessed: 31.12.2018). Insight-Centre.

[2] Keith Alexander, Richard Cyganiak, Michael Hausenblas, and Jun Zhao. 2011. *Describing Linked Datasets with the VoID Vocabulary.* W3C Interest Group Note 03 March 2011. W3C.

[3] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. DBpedia: A nucleus for a web of open data. In *Proceedings of the International Semantic Web Conference (ISWC).* LNCS, Busan, South Korea, 722–735.

[4] Ciro Baron Neto, Kay Müller, Martin Brümmer, Dimitris Kontokostas, and Sebastian Hellmann. 2016. LODVader: An Interface to LOD Visualization, Analytics and DiscovERy in Real-time. In *Proceedings of the 25th International Conference Companion on World Wide Web.* ACM, Montreal, Quebec, Canada, 163–166.

[5] Wouter Beek, Laurens Rietveld, Hamid R Bazoobandi, Jan Wielemaker, and Stefan Schlobach. 2014. LOD laundromat: a uniform way of publishing other people's dirty data. In *Proceedings of the International Semantic Web Conference (ISWC).* LNCS, Riva del Garda, Italy, 213–228.

[6] Mohamed Ben Ellefi, Zohra Bellahsene, Stefan Dietze, and Konstantin Todorov. 2016. Dataset Recommendation for Data Linking: An Intensional Approach. In *Proceedings of the Extended Semantic Web Conference.* LNCS, Crete, Greece, 36–51.

[7] Tim Berners-Lee. 2006. Linked Data. W3C Design Issues. From `http://www.w3.org/DesignIssues/LinkedData.html`; (Accessed: 27.10.2010.

[8] Christoph Böhm, Felix Naumann, Ziawasch Abedjan, Dandy Fenz, Toni Grütze, Daniel Hefenbrock, Matthias Pohl, and David Sonnabend. 2010. Profiling linked open data with ProLOD. In *Proceedings of the 26th IEEE International Conference on Data Engineering Workshops (ICDEW 2010)*. IEEE, Long Beach, CA, USA, 17–178.

[9] Dan Brickley and Libby Miller. 2007. FOAF vocabulary specification 0.91.

[10] Anila Sahar Butt, Armin Haller, and Lexing Xie. 2014. Ontology Search: An Empirical Evaluation. In *Proceedings of the International Semantic Web Conference (ISWC)*. LNCS, Riva del Garda, Italy, 130–147.

[11] Alison Callahan et al. 2013. Bio2RDF release 2: Improved coverage, interoperability and provenance of life science linked data. In *Proceedings of the Extended Semantic Web Conference (ESWC)*. LNCS, Montpellier, France, 200–212.

[12] O. Curé, G. Blin, D. Revuz, and D.C. Faye. 2014. WaterFowl: A Compact, Self-indexed and Inference-Enabled Immutable RDF Store. In *Proceedings of the Extended Semantic Web Conference (ESWC)*. 302–316.

[13] Richard Cyganiak, Holger Stenzhorn, Renaud Delbru, Stefan Decker, and Giovanni Tummarello. 2008. Semantic Sitemaps: Efficient and Flexible Access to Datasets on the Semantic Web. In *Proceedings of the European Semantic Web Conference (ESWC)*. LNCS, Tenerife, Canary Islands, Spain, 690–704. https://doi.org/10.1007/978-3-540-68234-9_50

[14] Jeremy Debattista, Sören Auer, and Christoph Lange. 2016. Luzzu - A Methodology and Framework for Linked Data Quality Assessment. *Journal of Data and Information Quality* 8, 1 (Oct. 2016), 4:1–4:32.

[15] Jeremy Debattista, Christoph Lange, Sören Auer, and Dominic Cortis. 2017. Evaluating the Quality of the LOD Cloud: An Empirical Investigation. *Semantic Web* Preprint (2017), 1–43.

[16] Dennis Diefenbach, Andreas Both, Kamal Singh, and Pierre Maret. 2018. Towards a question answering system over the Semantic Web. *Semantic Web* Preprint (2018), 1–19.

[17] Fredo Erxleben, Michael Günther, Markus Krötzsch, Julian Mendez, and Denny Vrandečić. 2014. Introducing Wikidata to the linked data web. In *Proceedings of the International Semantic Web Conference (ISWC)*. LNCS, Riva del Garda, Italy, 50–65.

[18] Javier D Fernández, Miguel A Martínez-Prieto, Claudio Gutiérrez, Axel Polleres, and Mario Arias. 2013. Binary RDF representation for publication and exchange (HDT). *Web Semantics: Science, Services and Agents on the World Wide Web* 19 (2013), 22–41.

[19] Olaf Görlitz and Steffen Staab. 2011. Splendid: Sparql endpoint federation exploiting void descriptions. In *Proceedings of the International Workshop on Consuming Linked Data, in conjunction with International Semantic Web Conference (ISWC)*. CEUR-WS.org, Bonn, Germany, 13–24.

[20] Andreas Harth, Sheila Kinsella, and Stefan Decker. 2009. Using Naming Authority to Rank Data and Ontologies for Web Search. In *Proceedings of the International Semantic Web Conference (ISWC)*. LNCS, Washington, DC., USA, 277–292.

[21] Ali Hasnain, Syeda Sana e Zainab, Maulik R Kamdar, Qaiser Mehmood, Claude N Warren, Qurratal Ain Fatimah, Helena F Deus, Muntazir Mehdi, and Stefan Decker. 2014. A roadmap for navigating the life sciences linked open data cloud. In *Semantic Technology*. Springer, 97–112.

[22] Aidan Hogan. 2011. *Exploiting RDFS and OWL for Integrating Heterogeneous, Large-Scale, Linked Data Corpora.* Ph.D. Dissertation. Digital Enterprise Research Institute.

[23] Aidan Hogan, Andreas Harth, Alexandre Passant, Stefan Decker, and Axel Polleres. 2010. Weaving the Pedantic Web. In *Proceedings of the WWW2010 Workshop on Linked Data on the Web, LDOW (CEUR Workshop Proceedings)*, Vol. 628. CEUR-WS.org, Raleigh, USA, 1–10. http://ceur-ws.org/Vol-628/ldow2010_paper04.pdf

[24] Aidan Hogan, Andreas Harth, and Axel Polleres. 2008. SAOR: Authoritative Reasoning for the Web. In *Proceedings of the International Semantic Web Conference (ISWC)*. LNCS, Karlsruhe, Germany, 76–90.

[25] Aidan Hogan, Jürgen Umbrich, Andreas Harth, Richard Cyganiak, Axel Polleres, and Stefan Decker. 2012. An empirical survey of Linked Data conformance. *Journal of Web Semantics* 14 (2012), 14 – 44. Special Issue on Dealing with the Messiness of the Web of Data.

[26] Wei Hu, Honglei Qiu, and Michel Dumontier. 2015. Link analysis of life science linked data. In *Proceedings of the International Semantic Web Conference (ISWC)*. LNCS, Bethlehem, PA, USA, 446–462.

[27] Maulik R. Kamdar. 2019. *A web-based integration framework over heterogeneous biomedical data and knowledge sources.* Ph.D. Dissertation. Stanford University. https://purl.stanford.edu/jr863br2478

[28] Maulik R. Kamdar et al. 2017. PhLeGrA: Graph Analytics in Pharmacology over the Web of Life Sciences Linked Open Data. In *Proceedings of the World Wide Web Conference (WWW)*. ACM, Perth, Australia, 321–329.

[29] Maulik R Kamdar, Tania Tudorache, and Mark A Musen. 2017. A systematic analysis of term reuse and term overlap across biomedical ontologies. *Semantic Web* 8, 6 (2017), 853–871.

[30] Maxime Lefrançois and Antoine Zimmermann. 2016. Supporting Arbitrary Custom Datatypes in RDF and SPARQL. In *Proceedings of the Extended Semantic Web Conference (ESWC)*. LNCS, Heraklion, Crete, Greece, 371–386.

[31] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N. Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sören Auer, and Christian Bizer. 2015. DBpedia - A large-scale, multilingual

knowledge base extracted from Wikipedia. *Semantic Web* 6, 2 (2015), 167–195. https://doi.org/10.3233/SW-140134

[32] Miguel A Martínez-Prieto, Mario Arias Gallego, and Javier D Fernández. 2012. Exchange and consumption of huge RDF data. In *Proceedings of the Extended Semantic Web Conference (ESWC)*. LNCS, Heraklion, Crete, Greece, 437–452.

[33] Nandana Mihindukulasooriya, María Poveda-Villalón, Raúl García-Castro, and Asunción Gómez-Pérez. 2015. Loupe-An Online Tool for Inspecting Datasets in the Linked Data Cloud. In *International Semantic Web Conference (Posters & Demos)*.

[34] Thomas Minier, Hala Skaf-Molli, and Pascal Molli. 2019. SaGe: Web Preemption for Public SPARQL Query Services. In *Proceedings of The Web Conference*. https://callidon.github.io/pdf/paper_www19.pdf.

[35] Matteo Palmonari, Anisa Rula, Riccardo Porrini, Andrea Maurino, Blerina Spahiu, and Vincenzo Ferme. 2015. ABSTAT: linked data summaries with abstraction and statistics. In *International Semantic Web Conference*. Springer, 128–132.

[36] Axel Polleres, Maulik R. Kamdar, Javier D. Fernández, Tania Tudorache, and Mark A. Musen. 2018. A More Decentralized Vision for Linked Data. In *Proceedings of the 2nd Workshop on Decentralizing the Semantic Web, co-located with the International Semantic Web Conference (ISWC), DeSemWebISWC*, Vol. 2165. CEUR-WS.org, Monterey, CA, USA, 8.

[37] Bastian Quilitz and Ulf Leser. 2008. Querying distributed RDF data sources with SPARQL. In *Proceedings of the European Semantic Web Conference (ESWC)*. LNCS, Tenerife, Canary Islands, Spain, 524–538.

[38] Laurens Rietveld, Wouter Beek, Rinke Hoekstra, and Stefan Schlobach. 2017. Meta-data for a lot of LOD. *Semantic Web* 8, 6 (2017), 1067–1080.

[39] Alan Ruttenberg, Tim Clark, William Bug, Matthias Samwald, Olivier Bodenreider, Helen Chen, Donald Doherty, Kerstin Forsberg, Yong Gao, Vipul Kashyap, et al. 2007. Advancing translational research with the Semantic Web. *BMC bioinformatics* 8, 3 (2007), S2.

[40] Matthias Samwald, Anja Jentzsch, Christopher Bouton, Claus Stie Kallesøe, Egon Willighagen, Janos Hajagos, M Scott Marshall, Eric Prud'hommeaux, Oktie Hassanzadeh, Elgar Pichler, et al. 2011. Linked open drug data for pharmaceutical research and development. *Journal of cheminformatics* 3, 1 (2011), 19.

[41] Max Schmachtenberg, Christian Bizer, and Heiko Paulheim. 2014. Adoption of the Linked Data Best Practices in Different Topical Domains. In *Proceedings of the International Semantic Web Conference (ISWC)*. LNCS, Riva del Garda, Italy, 245–260.

[42] Guus Schreiber and Yves Raimond. 2014. RDF 1.1 Primer. W3C Note.

[43] Blerina Spahiu, Riccardo Porrini, Matteo Palmonari, Anisa Rula, and Andrea Maurino. 2016. ABSTAT: ontology-driven linked data summaries with pattern minimalization. In *International Semantic Web Conference*. Springer, 381–395.

[44] Blerina Spahiu, Cheng Xie, Anisa Rula, Andrea Maurino, and Hongming Cai. 2016. Profiling similarity links in Linked Open Data. In *Proceedings of the 32nd IEEE International Conference on Data Engineering Workshops (ICDEW)*. IEEE, Helsinki, Finland, 103–108.

[45] Larry Masinter Tim Berners-Lee, Roy Fielding. 2005. Uniform Resource Identifier (URI): Generic Syntax. IETF Network Working Group Request for Comments: 3986 (RFC3986). Available at https://tools.ietf.org/html/rfc3986.

[46] Andrew Layman Richard Tobin Henry S. Thompson Tim Bray, Dave Hollander. 2009. Namespaces in XML 1.0 (Third Edition). Available at https://www.w3.org/TR/xml-names/.

[47] Pierre-Yves Vandenbussche, Ghislain Atemezing, María Poveda-Villalón, and Bernard Vatant. 2017. Linked Open Vocabularies (LOV): A gateway to reusable semantic vocabularies on the Web. *Semantic Web* 8, 3 (2017), 437–452.

[48] Pierre-Yves Vandenbussche, Jürgen Umbrich, Luca Matteis, Aidan Hogan, and Carlos Buil Aranda. 2017. SPARQLES: Monitoring public SPARQL endpoints. *Semantic Web* 8, 6 (2017), 1049–1065. https://doi.org/10.3233/SW-170254

[49] Ruben Verborgh, Miel Vander Sande, Olaf Hartig, Joachim Van Herwegen, Laurens De Vocht, Ben De Meester, Gerald Haesendonck, and Pieter Colpaert. 2016. Triple Pattern Fragments: A low-cost knowledge graph interface for the Web. *Journal of Web Semantics* 37-38 (2016), 184–206.

[50] Denny Vrandecic and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Communication of the ACM* 57, 10 (2014), 78–85.

[51] Denny Vrandecíc, Markus Krötzsch, Sebastian Rudolph, and Uta Lösch. 2010. Leveraging non-lexical knowledge for the linked open data web. *Review of April Fool's day Transactions (RAFT'2010)* 5 (2010).

[52] Patricia L Whetzel et al. 2011. BioPortal: enhanced functionality via new Web services from the National Center for Biomedical Ontology to access and use ontologies in software applications. *Nucleic acids research* 39, suppl 2 (2011), W541–W545.

[53] Mark D Wilkinson, Michel Dumontier, IJsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, Jan-Willem Boiten, Luiz Bonino da Silva Santos, Philip E Bourne, et al. 2016. The FAIR Guiding Principles for scientific data management and stewardship. *Scientific data* 3 (2016).

[54] Amrapali Zaveri, Anisa Rula, Andrea Maurino, Ricardo Pietrobon, Jens Lehmann, and Sören Auer. 2016. Quality assessment for Linked Data: A Survey. *Semantic Web* 7, 1 (2016), 63–93.