CEDAL: Time-Efficient Detection of Erroneous Links in Large-Scale Link Repositories

André Valdestilhas*
AKSW, University of Leipzig
Augustusplatz 10
Leipzig, Germany 04109
valdestilhas@informatik.uni-leipzig.de

Tommaso Soru
AKSW, University of Leipzig
Augustusplatz 10
Leipzig, Germany 04109
tsoru@informatik.uni-leipzig.de

Axel-Cyrille Ngonga Ngomo
AKSW, University of Leipzig
Augustusplatz 10
Leipzig, Germany 04109
ngonga@informatik.uni-leipzig.de

ABSTRACT

More than 500 million facts on the Linked Data Web are statements across knowledge bases. These links are of crucial importance for the Linked Data Web as they make a large number of tasks possible, including cross-ontology, question answering and federated queries. However, a large number of these links are erroneous and can thus lead to these applications producing absurd results. We present a time-efficient and complete approach for the detection of erroneous links for properties that are transitive. To this end, we make use of the semantics of URIs on the Data Web and combine it with an efficient graph partitioning algorithm. We then apply our algorithm to the LinkLion repository and show that we can analyze 19,200,114 links in 4.6 minutes. Our results show that at least 13% of the `owl:sameAs` links we considered are erroneous. In addition, our analysis of the provenance of links allows discovering agents and knowledge bases that commonly display poor linking. Our algorithm can be easily executed in parallel and on a GPU. We show that these implementations are up to two orders of magnitude faster than classical reasoners and a non-parallel implementation.

ACM Reference format:
André Valdestilhas, Tommaso Soru, and Axel-Cyrille Ngonga Ngomo. 2017. CEDAL: Time-Efficient Detection of Erroneous Links in Large-Scale Link Repositories. In *Proceedings of Web Intelligence, Leipzig, Germany, August 2017 (Submitted to WI2017),* 8 pages.
DOI: 10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Links across knowledge bases play a fundamental role in Linked Data [3] as they allow users to navigate across datasets, integrate Linked Data sources [17], perform federated queries [19] across data sources and perform large-scale inference on the data. Given the importance of links, corresponding repositories such as *sameas.org*[1] and LinkLion [14] (of which *sameas.org* is a subset) have been created. In addition to facilitating the finding of links between resources and knowledge bases, these repositories also allow detecting significant errors across links. For example, according to LinkLion and by virtue of transitivity, the resources `orca:21075` and `orca:1946`[2] stand for the same entity of the real world but have different URIs within the same knowledge base. This clearly goes against the definition of URIs as used in RDF. Figure 1 shows a fictional example to help illustrate such problems, which can be classified as **contradiction problems**, according to the quality dimension of consistency [26]. In our example, we can infer that one of the links along the path that led to this inference is wrong or that the knowledge base in itself contains an error. While such errors can be potentially detected by computing the closure of equivalence links using the characteristics of equivalence relations and an inference engine, our experiments with Pellet [8] – the fastest inference engine to the best of our knowledge – suggest that inference engines do not scale to the millions of links found on the Web of Data.

The poor performance on closure computations is also known from literature (see, e.g., [5]). For instance, the computation of closures in RDF graphs has several drawbacks. Firstly, it is known that the **size** of the transitive closure of a graph G is of **quadratic order** in the worst case, making the computation and storage of the closure too expensive for web-scale applications. Secondly, once the transitive closure has been computed, all queries are evaluated over a data source which can be much larger than the original one. This can be particularly **inefficient for queries** that must scan a large part of the input data.

Our intuition is that it is actually not necessary to compute the closures; instead, we can use adjacency lists and create graph partitions based on an algorithm called *Union Find* [21]. Therewith, we obtain a solution with a time complexity that is decreased from $O(n^2)$ to $O(m \log n)$, where m is the number of operations (either *Union* or *Find*) that are applied to n elements.

In this paper, we aim to find erroneous links across knowledge bases by reusing the uniqueness of the semantics of URIs within given knowledge bases. We hence present a novel time-efficient algorithm called *Consistency Error Detection Algorithm* (CEDAL), in which the error detection consists of finding distinct resources (i.e., resources with distinct URIs) which share the same dataset, given an RDF graph representing the union of all knowledge bases in a link repository.

With our work, we address the following research questions:

(1) Is there a time-efficient algorithm to detect erroneous links in large-scale link repositories?

*All authors contributed equally.

[1] http://sameas.org/

[2] `orca` stands for the namespace `http://orca.cf.ac.uk/id/eprint/`.

Submitted to WI2017, Leipzig, Germany
© 2017 Copyright held by the owner/author(s). 978-x-xxxx-xxxx-x/YY/MM…$15.00
DOI: 10.1145/nnnnnnn.nnnnnnn

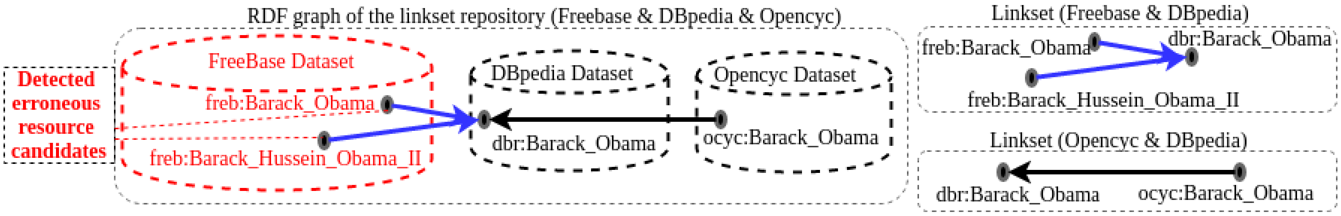

RDF graph of the linkset repository (Freebase & DBpedia & Opencyc)
Detected erroneous resource candidates
FreeBase Dataset
freb:Barack_Obama
freb:Barack_Hussein_Obama_II
DBpedia Dataset
dbr:Barack_Obama
Opencyc Dataset
ocyc:Barack_Obama
Linkset (Freebase & DBpedia)
freb:Barack_Obama
dbr:Barack_Obama
freb:Barack_Hussein_Obama_II
Linkset (Opencyc & DBpedia)
dbr:Barack_Obama
ocyc:Barack_Obama

Figure 1: Manual detection of erroneous resource candidates.

(2) Is there an approach to discover whether a linkset[3] is consistent without computing all closures required by the property axiom?

The contributions of this work are listed below:

- A time-efficient algorithm for the detection of erroneous links in large-scale link repositories without computing all closures required by the property axiom.
- An approach that brings the possibility to track the consistency problems inside link repositories.
- A scalable algorithm that works well in a parallel and non-parallel mode.
- A study case applied to a link repository called LinkLion.
- A new linkset quality measure based on the number of erroneous candidates.

The remainder of this paper is structured as follows: Section 6 presents related work; Section 2 introduces the definitions and concepts used throughout this paper; Section 3 presents our algorithm for the detection of erroneous links in large-scale link repositories; Section 4 presents the error types and a quality measure for linksets; Section 5 presents the evaluation of our approach; and finally, Section 7 presents the conclusions and future work.

2 PRELIMINARIES

RDF graph. An RDF graph is a set of RDF triples which has a set of subjects and objects of triples in the graph called nodes. Given an infinite set U of URIs, an infinite set B of blank nodes and an infinite set of literals L, a RDF triple is a triple $\langle s, p, o \rangle$ where the subject $s \in (U \cup B)$, the predicate $p \in U$ and the object $o \in (U \cup B \cup L)$. An RDF triple represents an assertion of a "piece of knowledge", so if the triple $\langle s, p, o \rangle$ exists, then, the logical assertion $p(s, o)$ holds true. An RDF graph is also represented by a collection of RDF triples and it can be seen as a set of statements describing, partially or completely, a certain knowledge.

Transitive property. Defined by: $\forall a, b, c \in X : (p(a, b) \wedge p(b, c)) \implies p(a, c)$, where p represents a relation between two elements of a set X.

Equivalence. An equivalence relation is a binary relation that is reflexive, symmetric and transitive. According to OWL semantics, `owl:sameAs` is an equivalence relation.

Linkset. According to the W3C,[4] a linkset is a collection of RDF links between two datasets. It is a set of RDF triples in which all subjects are in one dataset and all objects are in another dataset. RDF links often have the `owl:sameAs` predicate, but any other property could occur as the predicate as well. Formally, according to [4], a linkset LS is a set of *RDF* triples where for all triples $t_i = \langle s_i, p_i, o_i \rangle \in LS$, the subject is in one dataset, i.e. all s_i are described in S, and the object is in another dataset, i.e. all o_i are described in T. We use the word *linkset* for either RDF knowledge base files and dump files from RDF link repositories.

RDF graph partitioning. Given a graph $G = (V, E, lbl, L)$, a graph partitioning, C, is a division of V into k partitions $P_1, P_2, ..., P_k$ such that $\bigcup_{1 \le i \le k} P_i = V$, and $P_i \cap P_j = \emptyset$ for any $i \neq j$. The edge cut E_c is the set of edges whose vertices belong to different partitions, $lbl : E \cup V \rightarrow L$ is a labeling function, and L is a set of labels. The definition comes from a recent survey about RDF graph partitioning [22].

3 METHOD

After introducing the terminology and symbolism used in this work, in this section, we present our error detection algorithm.

3.1 Error Detection algorithm

Our algorithm targets consistency errors in large-scale link repositories. We assume a union of linksets $\mathcal{L}$ as given input. The aim is to find cases in which equivalent resources (according to the OWL semantics) in $\mathcal{L}$ share the same dataset. The basic intuition here is equivalent resources (i.e., resources that stand for the same entity from the real world) being in one knowledge base is a clear hint towards (1) an error in the knowledge base itself or (2) an error during the generation of the links that allowed generating this equivalence.

In Figure 2, we show how our algorithm works. Given datasets $D_1, ..., D_n$, resources $R = \{a, b, c, d, e, f\}$, the idea is to detect two or more resources sharing the same dataset inside the same cluster, following the steps described in algorithm 1.

Our algorithm works by partitioning a graph inside an adjacency list that contains: (1) An array per vertex for a total of V arrays, where we are only considering the space for the array pointer, not the contents of the array. (2) Each directed edge is contained once somewhere in the adjacency list, for a total of E edges, where a bidirectional edge is just 2 directed edges, assuming we are using bi-directional edges.

3.1.1 Problem statement. We formalize the problem of *detection of erroneous links in large-scale link repositories* as follows.

[3] We refer to the definition of linkset as defined in the W3C document at https://www.w3.org/TR/void/.

[4] https://www.w3.org/TR/void/#linkset

Algorithm 1 Consistency Error Detection Algorithm (CEDAL)

```
    Input: $L : L_i = (s, p, o) : s \in D_i^{(s)}, o \in D_i^{(t)}$
    Output: An error list with erroneous nodes.
procedure CEDAL($G(V, E)$)
    $\mathcal{P}$ = getPartitions($G(V, E)$)
    for all $p \in \mathcal{P}$ do (in parallel)
        for all $r \in p$ do push onto $clusterDataset$ r, $MapResourcesDataset.get(r)$
        end for
        for all $r \in clusterDataset$ do
            if $countDataset(r) > 1$ then push onto $ErrorList$ resource.originalFileName + resource.dataset + resource.path
            end if
        end for
    end for
    return $ErrorList$
end procedure
procedure GETPARTITIONS($G(V, E)$)
    for all $v \in V$ do
        push onto $MapResourcesDataset$ $v, extracDataset(v)t$
        push all nodes connected to $v$ onto $\mathcal{V}$          ▷ UnionFind algorithm
        push $\mathcal{V}$ onto $\mathcal{P}$
    end for
    return $\mathcal{P}$
end procedure
```

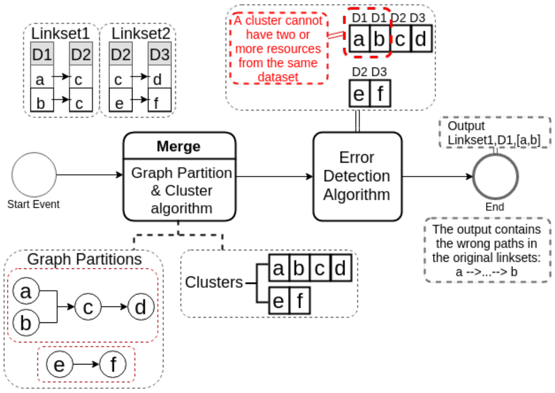

Linkset1
D1
D2
a
b
c
c
Linkset2
D2
D3
c
e
d
f
A cluster cannot have two or more resources from the same dataset
D1 D1 D2 D3
a b c d
D2 D3
e f
Output Linkset1,D1,[a,b]
Start Event
Merge
Graph Partition & Cluster algorithm
Error Detection Algorithm
End
The output contains the wrong paths in the original linksets: a -->...--> b
Graph Partitions
a
b
c
d
e
f
Clusters
a b c d
e f

Figure 2: Error detection.

From this section on, we will refer to the union of all linksets as $\mathcal{L} = \bigcup_i L_i$, the set of all datasets as $\mathcal{D}$, the clusters (or graph partitions) as C, the candidates (i.e., set of resources belonging to the same dataset) $\mathcal{P}$. A linkset $L \in \mathcal{L}$ contains triples (or links) (s, p, o) such as:

$$(s, p, o) \in L : s \in D_i, o \in D_j, i \neq j \tag{1}$$

Each candidate P abides by the following restriction:

$$\forall r_i, r_j \in P : r_i \neq r_j \Rightarrow (r_i, x, r_j) \in \mathcal{L}^* \vee (r_j, x, r_i) \in \mathcal{L}^* \tag{2}$$

where x is a property (e.g., `owl:sameAs`). In other words, each element in P is linked to at least another element in the set. Candidates are assigned one of two classes, positive (i.e., candidates with errors) or negative. The positive cases are represented as follows.

$$P \in \mathcal{P}^+ \iff (\exists r_1 \in P \cap D_1, r_2 \in P \cap D_2) \therefore D_1 = D_2 \Rightarrow r_1 \neq r_2 \tag{3}$$

The negative cases are defined in the following equation.

$$P \in \mathcal{P}^- \iff (\forall r_1 \in P \cap D_1, r_2 \in P \cap D_2) \therefore D_1 = D_2 \Rightarrow r_1 = r_2 \tag{4}$$

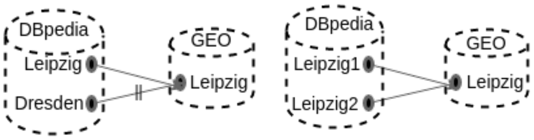
The target is thus to find the set of erroneous candidates $\mathcal{P}^+$. As shown in the next section, we cannot state that a link connecting these resources is wrong, but we can state that the error lies somewhere between the links that connect them and the organization of the dataset they belong to.

4 ERROR TYPES AND QUALITY MEASURE FOR LINKSET REPOSITORIES

The application of the two measures requires the output from CEDAL, allowing to identify two types of errors among the erroneous candidates from the output.

4.1 Error Types

We identified two types of errors, in which can be defined in quality dimensions by [26]. (1) **Semantic accuracy** errors, in which we detect if data values correctly represent the real world facts. (2) **Consistency** and **Conciseness** errors where a knowledge base is free of logical or formal contradictions concerning particular knowledge representation and inference mechanisms and the minimization of redundancy of entities at the schema and the data level. Figure 3 shows a fictional example of both error types, in which we represent links between GeoNames[5] and DBpedia.[6]



(a) Error type (1).

(b) Error type (2).

Figure 3: Detected error types.

In this example, Figure 3(a) shows an error of type (1), in which an erroneous `owl:sameAs` link between the city of Dresden and the city of Leipzig was detected. The fig. 3(b) shows an error of type (2) where the resource about the city of Leipzig is duplicated within the DBpedia dataset.

[5] http://www.geonames.org/

[6] http://dbpedia.org/

We manually analyzed a random sample of the errors. Among 100 occurrences, 90% are of type (2). It was not feasible in practice to perform this evaluation in an automatic way, due to the fact that it involves semantic accuracy and thus needs human feedback. Moreover, some URIs are unreachable, resulting many times in *timeout errors*, such as HTTP 404, 500 and 503 errors.[7] In summary, it was not practicable to automatically distinguish these type of errors among the erroneous candidates detected by CEDAL.

4.2 Quality Measure

Based on the error types from CEDAL, we present three linkset quality measures, evaluating the information accessed by crosswalking the linksets of LinkLion.

The **Semantic Accuracy of linksets** indicates whether the data values from the RDF links represent real world facts. **Example:** Let us assume that we have a linkset from DBpedia and Geonames. A link `<dbr:dresden owl:sameAs geo:leipzig>` would clearly be inaccurate, since Dresden and Leipzig are two different cities.

The **consistency and conciseness of links** inform whether a linkset is free of logical or formal contradictions with respect to particular knowledge representation and inference mechanisms and the minimization of redundancy of resources that belongs to the same dataset inside a linkset repository. **Example:** With a linkset from DBpedia and Geonames, let us assume we found two links represented by `<dbr:leipzig1 owl:sameAs geo:leipzig>` and `<dbr:leipzig2 owl:sameAs geo:leipzig>`. Since `dbr:leipzig1` and `dbr:leipzig2` belong to the same dataset, this characterizes a redundancy and it contradicts the assumption that two URIs in a dataset cannot stand for the same thing from the real world.

In order to evaluate data quality in linksets, on the lines of the works summarized in the Data Quality survey [26], we propose three new metrics:

M1: Rate of consistent resources inside linkset repositories. Let us consider a candidate $P \in \mathcal{P}$ containing only resources which belong to the same dataset. The rate of consistent candidates is defined as follows:

$$M1 = \frac{\sum_{P \in \mathcal{P}^-} |P|}{\sum_{P \in \mathcal{P}} |P|} \tag{5}$$

where $\mathcal{P}^-$ is the set of consistent (i.e., non-erroneous) candidates. We call $M1$ the **consistency index**.

M2: Rate of candidates in $\mathcal{P}$ containing resources whose internal links are real world facts. Let us introduce a function $f(s, p, o)$ which expresses the verification of a triple (s, p, o) in the real world, assuming value 1 if the statements holds true and 0 otherwise. This metric addresses errors of type (1).

$$M2 = \frac{|\{P \in \mathcal{P}^+ : \forall r_i, r_j \in P \; r_i \neq r_j \Rightarrow f(r_i, p, r_j) = 1\}| + |\mathcal{P}^-|}{|\mathcal{P}|} \tag{6}$$

M3: Rate of candidates in $\mathcal{P}$ which are free of redundant resources. This metric addresses errors of type (2).

$$M3 = \frac{|\{P \in \mathcal{P}^+ : \exists r_i, r_j \in P \; r_i \neq r_j \Rightarrow f(r_i, p, r_j) = 0\}| + |\mathcal{P}^-|}{|\mathcal{P}|} \tag{7}$$

As can be seen, M2 and M3 are dependent on each other.

In this paper, we focus on the computation of M1. Although M2 and M3 are left for future research, we included them to encourage their evaluation and use.

5 EVALUATION

To verify our hypothesis, in this section we show that CEDAL brings an efficient way to track the erroneous candidates inside large-scale linkset repositories.

5.1 Experimental setup

As our study case, we use a linkset repository called LinkLion [14] due to some advantages such as provenance, linksets from the most used datasets, i.e. DBpedia, Yago and Opencyc, where the users are empowered to upload links and specify how these were created. Moreover, users and applications can select and download sets of links via dumps or SPARQL queries.

The table 1 shows that 99.9% of links from LinkLion are `owl:sameAs` links, amounting to $19,200,114$ triples. Thus, in our experiments we are using only `owl:sameAs` links.

Table 1: Link types

Property	Triples
sameAs	19,606,657 (with duplicates)
country	1,309
author	766
spokenIn	624
locatedIn	250
exactMatch	167
near	30
spatial#P	28
seeAlso	14
organism	14
made	4

The experiments were performed using two configurations: (1) a laptop with Intel Core i7, 8 GB RAM, a video card NVIDIA NVS4200, Operational System MS Windows 10 and Java SE Development Kit 8. (2) An Intel Xeon Core i7 processor with 40 cores, 128 GB RAM on an Ubuntu 14.04.5 LTS with Java SE Development Kit 8. The results including the output file for LinkLion are available online. The total number of $19.6 million$ links was processed by our algorithm in 4.6 minutes with the configuration (2). The total amount of errors were $1,352,366$ of candidates, where the total amount of domains were 254 and the number of linkset files was 553, where 48.3% of these knowledge base files has less than 10 resources detected as erroneous candidates.

5.2 Ranking the erroneous candidates

To evaluate how effective CEDAL is, we create a score in order to rank the erroneous candidates based on the number of detected resources with errors, in which the table 2, show two fictional examples of tuples in the same pattern of the output from CEDAL.

[7] https://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html

Table 2: Fictional example results.

Knowledge-base	Data-set domain	$\mathbb{C}$	μ
Linkset1.nt	Data-set1	URI1,URI2	1
Linkset2.nt	Data-set2	URI1,URI2,URI3,URI4	6

The μ score is calculated by $\mu = \frac{|\mathbb{C}|(|\mathbb{C}|-1)}{2}$, in which we use the cardinality of $\mathbb{C}$ representing the detected erroneous candidates. The figs. 4(a) and 4(b) shows the top 5 erroneous candidates according to the rank score.

Table 3: Legend for the figs. 4(a) and 4(b)

Label	Knowledge Base
K1	dotac.rkbexplorer.com—eprints.rkbexplorer.com.nt
K2	d-nb.info—viaf.org.nt
K3	dblp.rkbexplorer.com—dblp.l3s.de.nt
K4	linkedgeodata.org—sws.geonames.org.nt
K5	citeseer.rkbexplorer.com—kisti.rkbexplorer.com.nt
K6	wiki.rkbexplorer.com—oai.rkbexplorer.com.nt
K7	www4.wiwiss.fu-berlin.de—dbpedia.org.nt
K8	southampton.rkbexplorer.com—nsf.rkbexplorer.com.nt
K9	rae2001.rkbexplorer.com—newcastle.rkbexplorer.com.nt
K10	lod.geospecies.org—bio2rdf.org.nt

Considering only linksets between different datasets, the knowledge-base with more errors comes from the links in

`dotac.rkbexplorer.com - eprints.rkbexplorer.com`

with 458, 324 links per mapping[8], in which we found 53, 074 erroneous resource candidates resulting in a score of 1, 408, 398, 201. The knowledge base with fewer errors comes from the links in

`lod.geospecies.org - bio2rdf.org.nt`

with 9, 723 links per mapping, in which we found 6 erroneous resource candidates and a score of 15, also 193 datasets with no errors at all.

5.3 Runtime experiments

The experiments were performed with the input size varying between 10^3 and 10^6 RDF triples using the configuration (1), as shown in Figure 5. Our algorithm processed all 19, 200, 114 links from LinkLion in 4.6 minutes with the configuration (2). The results indicate that our algorithm scales well to large links repositories and can also be adapted to the hardware on which it is executed. For example, it can be easily implemented to make use of benefits of CPUs and GPUs.

5.3.1 Scalability Evaluation. Our algorithm performs well in parallel and non-parallel environments. The performance of our algorithm improved in accordance to the number of CPUs, showing that our algorithm is scalable, performing well with large linksets with size more than 10^6 as shown in figs. 5(a) and 5(b).

5.3.2 Parallel Implementation. Our algorithm implementation contains parallel code snippets in which we perform a load-and-balance of the data among CPU/GPU cores when available. This specific characteristic offers the possibility for utilization when hardware for parallel computing is available, such as CPU/GPU processors.

To illustrate this part of our idea, we can state: Given a graph $G(V, E)$, that contains all linksets from the repository $G(V, E) \subseteq \mathcal{L}$, this graph has partitions $P \subset G(V, E)$. Thus, errors are calculated for each partition, the processes are separated in threads and these threads are spread among CPU/GPU cores. Thus, we process the graph partitions in parallel, as shown in Figure 6.

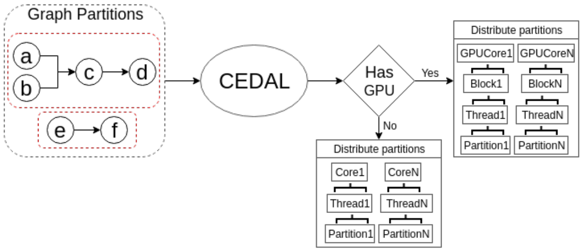


Figure 6: CEDAL CPU/GPU processing

5.4 Consistency by provenance of links

Thanks to the information found in LinkLion, we are able to check the provenance of links. This allowed us to analyze them more in details by finding which link discovery framework has created the link. The links were generated by four types of frameworks: *LIMES* [16], *SILK* [24], *DBpedia Extraction Framework* [12] and *sameas.org*[9]. The type of links provided by *sameas.org* were generated into human-curated knowledge bases, such as DBpedia (which is based on Wikipedia), Freebase, and OpenCyc.

We found a 13.5% error rate from *sameas.org* versus a 4.1% error rate of the algorithms such as *LIMES* [16]. They are all below 5% as table 4 shows; column *Errors* represents the rate of all resources belonging to error candidates and column $M1$ represents the respective quality measure.

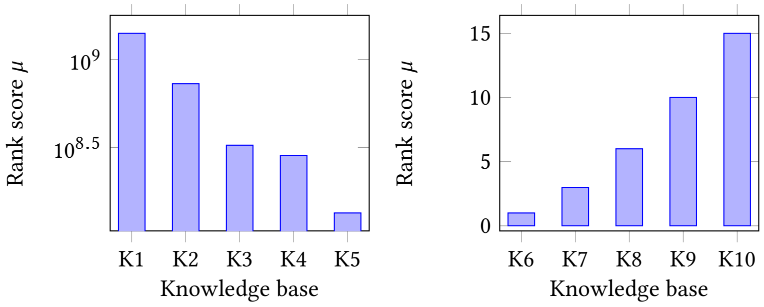
According to this data, we can say that algorithms such as LIMES, SILK, and the DBpedia Extraction Framework have a higher consistency index than *sameas.org*. This might be explained by the fact that no mechanism of link validation is present on *sameas.org*.

5.5 Comparison with other works

To the best of our knowledge, CEDAL is the first approach that aims to detect the consistency of RDF link repositories. However, the problem that CEDAL addresses can be solved in other ways. One alternative to solving our problem is to use reasoning. However, this approach requires the computation of all closures required by the property axiom. To check whether our approach performs better than a closure-based approach, we compared CEDAL with an algorithm for computing closures – dubbed Closure Generator – without using a reasoner and with Pellet, which is considered the

[8] Links per mapping from http://www.linklion.org/

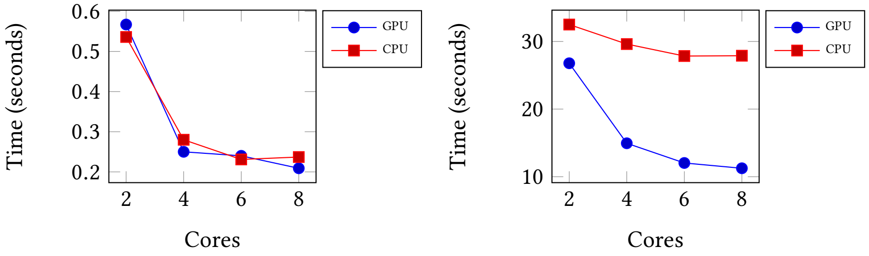
[9] http://sameas.org



(a) Top 5 Knowledge-base pairs with more candidates.

(b) Top 5 Knowledge-base pairs with fewer candidates.

Figure 4: Error rank (legends: see table 3).



(a) input size=10^3.

(b) input size=10^6.

Figure 5: Runtime results according to the input size, CPU and GPU.

Table 4: Comparison of results with respect to the provenance of the links.

Framework	Errors	Resources	Errors (%)	M1
sameas.org	3,792,326	28,130,994	13.5	0.865
LIMES	1,130	27,819	4.1	0.951
Silk	5,933	208,300	2.8	0.972
DBpedia Extraction Framework	12,615	914,180	1.4	0.986
All frameworks	**3,812,004**	**29,281,293**	**13.0**	**0.870**

state-of-the-art reasoner [8]. Figure 7 shows that CEDAL is significantly faster than Pellet, reaching up to three orders of magnitude of speedup when faced with 10^6 triples. This result can be partly explained by Pellet also checks the knowledge base for every single coherence and consistency axiom. However, we did not need such an in-depth analysis.

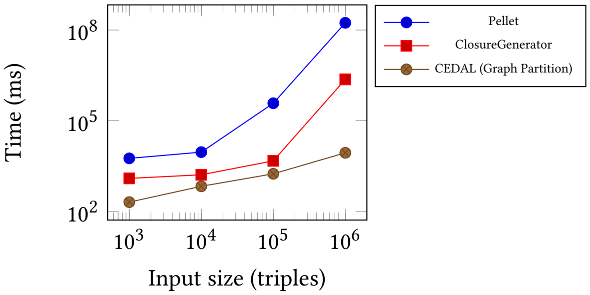


Figure 7: Pellet vs. ClosureGenerator vs. CEDAL

6 RELATED WORK

Our work has the aim of detecting erroneous links in large-scale link repositories. Related works include the following:

- Albertoni et al. [3] focus on the quality dimension of completeness using scoring functions and also introduce a notion of linkset quality, considering only `owl:sameAs`. The work proposes three quality indicators to assess completeness. The extension of this work [1] focuses on `skos:exactMatch` linksets and a multilingual gain.
- The LINK-QA tool [11] uses two network measures designed for Linked Data (i.e., open `sameAs` chains, and description richness) and three classic network measures (i.e., degree, centrality, clustering coefficient) in order to determine whether a set of links can improve the quality of linked data.
- DBpediaSameAs [23] is a work in which Transitive Redirects Links are redundancies at DBpedia that supposes a link to the same place, in other words, they use the `owl:sameAs` property. These links will redirect other links, to provide a transition between the links, hence the name transitive. In this case, instead of using the transitive links that point to the same final destination URI, the final URI is used directly.
- The work proposed in [25] is a metric-driven approach for interlinking assessment of a single dataset. It introduces the concept of *link-ability*, which shows the potential of a dataset being linked from other datasets, and in general, assesses whether a dataset is good to be interlinked with another dataset using three groups of metrics.
- The approach at [9] proposes strategies in order to reduce the cognitive overhead of creating materialized `owl:sameAs` linksets and to correctly maintain them using two types of components that triple stores should include, which would improve the support for materialized `owl:sameAs` linksets in the creation and maintenance stages.
- [15] evaluates the quantity of links between distributions, datasets, and ontologies categorizing and defining different types of links using probabilistic data structures. The results show valid links, dead links, and a number of namespaces described by distributions and datasets. The analysis was conducted using LODVader [6]. An important point here and in works such as [7, 13, 20] is that they do not mention or use any quality dimension as defined in some important works such as [1, 26]; moreover, they do not consider the axioms related to the properties. The quality is given solely by the number of links between datasets.
- The work described in [18] covers an unsupervised approach for finding erroneous links, in which each link is represented as a feature vector in a higher dimensional vector space, and finds wrong links by means of different multi-dimensional outliers.
- The W3C has a vocabulary to express the quality of data, including a linkset[10], that is based on the survey by Zaveri et al. [26].
- The work described in [2] discuss results of quality evaluation on linksets created using a framework called LusTRE with two quality measures, the *average linkset reachability* and the *average linkset importing*. Similar to CEDAL, this work realizes that the research on Linked Data quality has been mainly focused on datasets, not on linksets. However, they focus on the `SKOS`[11] vocabulary, more precisely `skos:exactMatch` linksets, and the experiments from CEDAL were processed in large scale link repositories with more than 10^7 triples, not only $31,298$ triples.
- The work described in [10] provides an algorithm with the intention to mitigate the problem of constraints violations in sameAs links automatically. Our approach has some different characteristics, such as CEDAL provides a classification of errors and show that some of them cannot be dealt with automatically in an accurate way, CEDAL use graph partitions in which scales better than this existing approach. This is also shown by the evaluation of CEDAL on larger datasets (19 million vs 3 million), CEDAL preserves the provenance of the links, CEDAL does not remove automatically constraints violations due to the fact that the output results involves semantic accuracy and thus needs human feedback. The similarities include the fact that we also focus on owl:sameAs and we reveal a significant amount of sameAs links that do not adhere to the strict semantics of the OWL standard and hence do not reflect genuine identity.

Common points among these existing works are the improvement and maintenance of link repositories. Aspects include the use of scoring functions, transitive and redirect links, metrics for interlinking datasets, probabilistic data structures, vector space models, and the creation of standards. Our approach provides a high-performance way to dealing with heterogeneous knowledge bases showing the provenance and detecting inconsistent links in large-scale link repositories. Although our work bears some resemblance to existing work on detecting erroneous link candidates in large link repositories, none of the above has considered evaluating the consistency of equivalence relations using a data quality measure.

The novelty of CEDAL with respect to the state-of-the-art can be enumerated in five points. Our approach (1) uses graph partitions and hence scales better than existing approaches using closures, (2) can be applied to larger linksets, with more than 19 million triples, (3) preserves the provenance of the links, (4) shows that some of the error classifications cannot be dealt with automatically in an accurate way and (5) does not remove automatically constraints violations due to the fact that the output results involves semantic accuracy and thus needs human feedback.

7 CONCLUSION AND FUTURE WORKS

In this paper, we present CEDAL, a new algorithm that allows tracking consistency problems inside linkset repositories. Our approach allows detecting potential causes of errors, for example the linkset, the underlying dataset(s) and the graph path where the problem subsists. We showed that our approach scales well. In particular, we reduced the complexity of obtaining clusters by relying on

[10] https://w3c.github.io/dwbp/vocab-dqg.html#ExpressQualLinkset

[11] https://www.w3.org/2004/02/skos/

graph partitions, thus decreasing the time complexity from $O(n^2)$ to $O(m \log n)$. Our results showed that at least 13% of `owl:sameAs` links we considered are erroneous, and algorithms LIMES, SILK and DBpedia Extraction Framework have a better consistency index than repositories such as *sameas.org*.

In future work, we will carry out a survey on Linkset quality. To the best of our knowledge, the survey [26] is the most complete collection of data quality measures, which, however, misses specific measures for linkset quality, such as ways a linkset can improve dataset quality [3, 9, 25]. Moreover, we will investigate how our graph partition algorithms can improve the performance of SPARQL endpoints by distributing resources among computer clusters, core processors, and GPUs. The CEDAL repository[12] contains all necessary resources to run CEDAL, verify and reproduce our results.

REFERENCES

[1] R. Albertoni, M. De Martino, and P. Podesta. A linkset quality metric measuring multilingual gain in skos thesauri. In *Proceedings of the 2nd Workshop on Linked Data Quality (LDQ2015), Portorož, Slovenia*, 2015.

[2] R. Albertoni, M. De Martino, and P. Podestà. Linkset quality assessment for the thesaurus framework lustre. In *International Conference, MTSR 2016, Göttingen, Germany, November 22-25, 2016, Proceedings*. Springer, 2016.

[3] R. Albertoni and A. G. Pérez. Assessing linkset quality for complementing third-party datasets. In *Proceedings of the Joint EDBT/ICDT 2013 Workshops*, EDBT '13, New York, NY, USA, 2013. ACM.

[4] K. Alexander, R. Cyganiak, M. Hausenblas, and J. Zhao. Describing linked datasets. In *LDOW*, 2009.

[5] M. Arenas, C. Gutierrez, and J. Pérez. An extension of sparql for rdfs. In *Semantic Web, Ontologies and Databases*. Springer, 2008.

[6] C. Baron Neto, K. Müller, M. Brümmer, D. Kontokostas, and S. Hellmann. Lodvader: An interface to lod visualization, analyticsand discovery in real-time. In *WWW Companion volume*, 2016.

[7] W. Beek, L. Rietveld, H. R. Bazoobandi, J. Wielemaker, and S. Schlobach. Lod laundromat: a uniform way of publishing other people's dirty data. In *International Semantic Web Conference*. Springer, 2014.

[8] J. Bock, P. Haase, Q. Ji, and R. Volz. Benchmarking owl reasoners.

[9] M. A. Casanova, V. M. Vidal, G. R. Lopes, L. A. P. P. Leme, and L. Ruback. On materialized sameas linksets. In *International Conference on Database and Expert Systems Applications*. Springer, 2014.

[10] G. de Melo. Not quite the same: Identity constraints for the web of linked data. In *AAAI*, 2013.

[11] C. Guéret, P. T. Groth, C. Stadler, and J. Lehmann. Assessing linked data mappings using network measures. In *ESWS*, 2012.

[12] J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P. N. Mendes, S. Hellmann, M. Morsey, P. van Kleef, S. Auer, et al. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*, 6(2), 2015.

[13] E. Mäkelä. Aether–generating and viewing extended void statistical descriptions of rdf datasets. In *European Semantic Web Conference*. Springer, 2014.

[14] M. Nentwig, T. Soru, A.-C. N. Ngomo, and E. Rahm. Linklion: A link repository for the web of data. In *The Semantic Web: ESWC 2014 Satellite Events*, pages 439–443. Springer, 2014.

[15] C. B. Neto, D. Kontokostas, S. Hellmann, K. Müller, and M. Brümmer. Assessing quantity and quality of links between linked data datasets.

[16] A.-C. N. Ngomo and S. Auer. Limes-a time-efficient approach for large-scale link discovery on the web of data. *integration*, 15:3, 2011.

[17] A. N. Ngomo, M. A. Sherif, and K. Lyko. Unsupervised link discovery through knowledge base repair. In *ESWC*, pages 380–394, 2014.

[18] H. Paulheim. Identifying wrong links between datasets by multi-dimensional outlier detection. In *WoDOOM*, 2014.

[19] M. Saleem, A.-C. N. Ngomo, J. X. Parreira, H. F. Deus, and M. Hauswirth. Daw: Duplicate-aware federated query processing over the web of data. In *ISWC*. Springer, 2013.

[20] F. Scharffe, Y. Liu, and C. Zhou. Rdf-ai: an architecture for rdf datasets matching, fusion and interlink. In *Proc. IJCAI 2009 workshop on Identity, reference, and knowledge representation (IR-KR), Pasadena (CA US)*, 2009.

[21] R. E. Tarjan. Efficiency of a good but not linear set union algorithm. *J. ACM*, 22(2), Apr. 1975.

[22] D. Tomaszuk, Ł. Skonieczny, and D. Wood. Rdf graph partitions: A brief survey. In *BDAS*. Springer, 2015.

[23] A. Valdestilhas, N. Arndt, and D. Kontokostas. Dbpediasameas: An approach to tackle heterogeneity in dbpedia identifiers.

[24] J. Volz, C. Bizer, M. Gaedke, and G. Kobilarov. Silk-a link discovery framework for the web of data. *LDOW*, 538, 2009.

[25] N. Yaghouti, M. Kahani, and B. Behkamal. A metric-driven approach for interlinking assessment of rdf graphs. In *Computer Science and Software Engineering (CSSE), 2015 International Symposium on*. IEEE, 2015.

[26] A. Zaveri, A. Rula, A. Maurino, R. Pietrobon, J. Lehmann, and S. Auer. Quality assessment for linked data: A survey. *Semantic Web*, 7(1), 2015.

[12] https://github.com/firmao/CEDAL