Constructing and Cleaning Identity Graphs in the LOD Cloud: Postprint

Authors: Raad, Joe, Beek, Wouter, van Harmelen, Frank, Wielemaker, Jan, Pernelle, Nathalie, Sais, Fatiha, Raad, Joe

Date: 2022-11-18T00:00:00+00:00

Abstract

The absence of a central naming authority on the Semantic Web frequently results in different datasets referring to identical entities through disparate identifiers. The utilization of multiple identifiers for equivalent entities necessitates owl:sameAs statements to facilitate data linking and promote reusability. Research dating to 2009 has documented instances of incorrect usage of the owl:sameAs property. In prior work, we introduced an identity graph comprising over 500 million explicit and 35 billion implied owl:sameAs statements, along with a scalable methodology for the automated computation of an error degree for each identity statement. This paper presents the generation of subgraphs from the comprehensive identity graph corresponding to specific error degree thresholds. We demonstrate that despite the prevalence of erroneous owl:sameAs statements within the Semantic Web, it remains feasible to leverage Semantic Web data while concurrently mitigating the deleterious effects of owl:sameAs misuse.

Full Text

Preamble

Constructing and Cleaning Identity Graphs in the LOD Cloud

Joe Raad[1]†, Wouter Beek[1], Frank van Harmelen[1], Jan Wielemaker[1], Nathalie Pernelle[2] & Fatiha Saïs[2]

[1]Department of Computer Science, Vrije Universiteit Amsterdam, De Boelelaan 1085, 1081 HV Amsterdam, The Netherlands
[2]Computer Science Research Laboratory (LRI) of the University Paris Sud, French National Centre for Scientific Research, Paris Saclay University, 91190 Saint-Aubin, France

Keywords: Linked Open Data; Identity; Quality; Reasoning

Citation: J. Raad, W. Beek, F. van Harmelen, J. Wielemaker, N. Pernelle & F. Saïs. Constructing and cleaning identity graphs in the LOD cloud. *Data Intelligence* 2(2020), 323-352. doi: 10.1162/dint_a_{00057}

Received: August 31, 2019; **Revised:** September 20, 2019; **Accepted:** September 30, 2019

Corresponding author: Joe Raad (E-mail: j.raad@vu.nl; ORCID: 0000-0002-7891-7738).

ABSTRACT

In the absence of a central naming authority on the Semantic Web, it is common for different data sets to refer to the same thing by different names. Whenever multiple names are used to denote the same thing, owl:sameAs statements are needed in order to link the data and foster reuse. Studies dating back as far as 2009 have observed that the owl:sameAs property is sometimes used incorrectly. In our previous work, we presented an identity graph containing over 500 million explicit and 35 billion implied owl:sameAs statements, along with a scalable approach for automatically calculating an error degree for each identity statement. In this paper, we generate subgraphs of the overall identity graph that correspond to certain error degrees. We show that even though the Semantic Web contains many erroneous owl:sameAs statements, it is still possible to use Semantic Web data while minimizing the adverse effects of misusing owl:sameAs.

---

1. INTRODUCTION

As the Web of Data continues to grow, more and more large data sets—covering a wide range of topics and domains—are being added to the Linked Open Data (LOD) Cloud. It is inevitable that different data sets, most of which are developed independently of one another, will come to describe (aspects of) the same thing, but will do so by referring to that thing by different names. This situation is not accidental: it is a defining characteristic of the (Semantic) Web that there is no central naming authority able to enforce a Unique Name Assumption (UNA). Thanks to identity links, data sets constructed independently are still able to make use of each other' s information. Consequently, identity link detection—the ability to determine with a certain degree of confidence that two different names in fact denote the same thing—is not a mere luxury but essential for Linked Data to work. The most common property used for interlinking data on the Web is owl:sameAs. An RDF statement of the form "x owl:sameAs y"

indicates that every property attributed to x must also be attributed to y, and vice versa.

Over time, an increasing number of Semantic Web studies have shown that the identity predicate is used incorrectly for various reasons (e.g., heuristic entity resolution techniques, lack of suitable alternatives for owl:sameAs, context-independent classical semantics). This misuse has resulted in the presence of incorrect owl:sameAs statements in the LOD Cloud, with some studies estimating this number to be around 2.8% [1] or 4% [2], while other studies suggest that possibly one out of five owl:sameAs statements on the Web is erroneous [3].

To limit the problem of identity links in the Semantic Web, various attempts have emerged in recent years (for a more exhaustive background, we refer the reader to [4]). In 2007, the authors of [5] proposed the centralized entity naming system OKKAM as a way to encourage reuse of existing names. This centralized solution did not receive uptake in the wider Semantic Web community. This is not entirely unexpected, since introducing a centralized naming authority amounts to a strong departure from original (Semantic) Web ideology, while simultaneously requiring significant changes to the use and interchange of Linked Data in practice. Other approaches tried to limit the Semantic Web identity problem by providing centralized access to identity statements published in a decentralized way. Such centralized identity services allow Linked Data consumers to make informed decisions regarding the quality of identity statements they encounter. Although such services can theoretically be useful for addressing the identity problem, current services suffer from semantic [6] and/or coverage [7] limitations.

Firstly, the "identity bundles" that the Consistent Reference Service [6] provides access to are the result of equivalence closure over owl:sameAs statements in addition to statements with different or no semantics. For example, umbel:isLike statements denote similarity instead of identity and are symmetric but not transitive; skos:exactMatch statements are symmetric and transitive, but indicate "a high degree of confidence that the concepts can be used interchangeably across a wide range of information retrieval applications," which is semantically very different from the notion of identity. As a result, the semantics of the closure calculated over this variety of statements is unclear. On the other hand, LODsyndesis [7] is a co-reference service based solely on owl:sameAs statements, but the number of triples it covers is an order of magnitude smaller than those we present in this work.

Besides the above-described services, various approaches have been proposed for detecting erroneous identity statements, based on similarity of textual descriptions associated with linked pairs [8], Unique Name Assumption (UNA) violations [9, 10], logical inconsistencies [1, 11], network metrics [12], and crowdsourcing [13]. However, existing approaches either do not scale to be applied to the LOD Cloud as a whole, or make assumptions about the data that may be valid in some data sets but not others. For example, in the LOD Cloud not

all names have textual descriptions, many data sets do not include vocabulary mappings, or they lack ontological axioms and assertions strong enough for deriving inconsistencies. While all mentioned approaches for erroneous identity link detection are useful in some cases, this paper presents a solution applicable to all data sets across the entire LOD Cloud.

In our previous work [14], we presented an identity graph of over half a billion owl:sameAs statements, obtained from a large crawl of the LOD Cloud, which includes links between resources from thousands of namespaces. We also presented the deductive closure of this identity graph, consisting of over 35 billion implied identity statements. In other previous work [15], we presented a scalable approach for automatically assigning an error degree to each owl:sameAs statement in a given RDF graph, based on a community detection algorithm. In an extension of both these works, we show in this paper that even though the LOD Cloud contains erroneous owl:sameAs statements, it is still possible to use a very large subset of the LOD Cloud while minimizing adverse effects of erroneous owl:sameAs statements. Specifically, we show for the first time that Linked Data practitioners can now control in practice the trade-off between (a) using more identity links, possibly not all correct, and benefiting from more contextual information from the LOD Cloud, and (b) using a smaller subset of correct identity links to limit the risk of propagating erroneous identity links and information through application of owl:sameAs semantics (i.e., transitive, symmetric, reflexive, and property sharing). Since the choice of a certain subset must depend on a quality metric, we rely on error degrees assigned in [15] as a metric of identity links' quality, and on the approach presented in [14] for deducing new—higher quality—identity sets in a computationally efficient manner. As a first experiment on how Linked Data practitioners can benefit from such a trade-off, we generate in this work two new identity subgraphs, compute their transitive closure, and analyze their resulting identity sets. In the first identity subgraph, we adopt a more conservative approach for dealing with the identity problem by considering only links of higher quality (i.e., with low error degree). In the second identity subgraph, we aim to limit the risk of incorrect identity assertions while minimizing loss of reachable information. This is done by discarding a smaller subset, consisting solely of highly questionable links (i.e., with high error degree). Both subgraphs with their resulting identity sets after transitive closure are publicly available[1].

[1] https://zenodo.org/record/3345674

The rest of this paper is structured as follows. Section 2 presents our approach for calculating the deductive closure of large identity graphs. Section 3 presents our approach for automatically assigning error degrees to owl:sameAs statements. Implementation and empirical evaluation of the deductive closure of a large identity graph obtained from the LOD Cloud are presented in Section 4. Section 5 experimentally shows how the two approaches can be combined for selecting new, higher quality identity subgraphs from the overall identity graph based on specified error degrees. Section 6 presents our conclusions on

the conducted work.

2. EQUIVALENCE CLOSURE COMPUTATION

This section presents our approach for calculating and storing the deductive closure of a large collection of owl:sameAs statements (see [14] for more details). The problem can be defined as follows: let N denote the set of RDF terms (i.e., IRIs, literals, and blank nodes) that appear in the subject or object position of at least one non-reflexive owl:sameAs triple. A partitioning of N is a collection of non-empty and mutually disjoint subsets Nk N (called partition members) that together cover N. In a network solely composed of N with their corresponding owl:sameAs edges, the connected components of this network are called identity clusters, and its partition members are called identity sets. According to owl:sameAs semantics, all RDF terms belonging to the same identity set denote the same real-world entity: x, y Nk $\rightarrow$ x = y. In this work, we do not consider singleton identity sets, which result from terms that appear only in reflexive owl:sameAs statements or terms that do not appear in any owl:sameAs statement.

To calculate the closure, each identity set should be closed under equivalence while considering multiple dimensions of complexity:

The closure can be too large to store. We will later see that the LOD Cloud contains identity sets with cardinality well over 100K. It is not feasible to store the closure materialization of each identity set, since space consumption of that approach is quadratic in the size of the identity set (e.g., the closure of an identity set of 100K terms contains 10B identity statements). Therefore, we do not store the materialization of the closure, but store the identity sets themselves, which is only linear in terms of the size of the universe of discourse (i.e., the set N of RDF terms).

|Nk| can be too large to store. Even the number of elements within one identity set can be too large to store in memory. Since our calculation of the closure must have a low hardware footprint and must be future-proof, we do not assume that every identity set is always small enough to fit in memory.

Data sets change over time. We calculate the identity closure for a large snapshot of the LOD Cloud. Since existing data sets in the LOD cloud are constantly changing, and new data sets are constantly added, our approach must support incremental updates of the closure. Specifically, our approach must allow for additions and deletions to be made over time, without requiring the entire closure to be recomputed.

Our approach consists of the following steps: (1) extract the explicit owl:sameAs statements, (2) remove owl:sameAs statements that are not necessary for calculating the deductive closure, and (3) compute the closure by partitioning the owl:sameAs network into several identity sets.

2.1 Explicit Identity Network

Given any RDF graph as input, e.g., the RDF merge of (a scrape of) the LOD Cloud, the first step of our approach is to extract identity statements from this data graph (Definition 1).

Definition 1 (Data Graph). A data graph is a directed and labelled graph G = (V, E, ΣE, lE). V is the set of RDF terms, E is the set of node pairs or edges, ΣE is the set of edge labels, and lE: ΣE → 2^E is a function that assigns to each edge e E a set of labels belonging to ΣE (with lE(e) representing the labels denoted to e).

From a given data graph G, we can extract the explicit identity network Gex (Definition 2), which is a directed labelled graph that only includes those edges whose labels include owl:sameAs.

Definition 2 (Explicit Identity Network). Given a graph G = (V, E, ΣE, lE), the corresponding explicit identity network Gex = (Vex, Eex) is the edge-induced subgraph G[{e E | {owl:sameAs} lE(e)}]. Vex is the set of RDF terms that appear in the subject and/or object position of at least one owl:sameAs statement (Vex V). Eex is the set of node pairs or edges for which a statement ⟨x, owl:sameAs, y⟩ has been asserted in G (Eex E).

2.2 Identity Network: Construction

Since identity is a reflexive, symmetric, and transitive relation, the size of the explicit identity network can be significantly reduced prior to calculating the deductive closure. Specifically, we can reduce the size of Gex into a more concisely represented undirected and labelled identity network I (Definition 3), without losing any information. Since reflexive owl:sameAs statements are implied by the semantics of identity, there is no need to represent them explicitly. In addition, since the symmetric statements eij and eji make the same assertion—that vi and vj refer to the same real-world entity—we can represent this more efficiently by including only one undirected edge with a weight of 2. A weight of 1 is assigned for edges where either eij or eji are present in Vex, but not both.

Definition 3 (Identity Network). Given Gex = (Vex, Eex), the identity network is an undirected labelled graph I = (VI, EI, {1, 2}, w), where VI is the set of RDF terms (VI Vex), and EI is the set of edges. {1, 2} are the edge labels, and w: EI → {1, 2} is the labelling function that assigns a weight wij to each edge eij. For an explicit identity network Gex = (Vex, Eex), the corresponding identity network I is derived as follows: - EI := {eij Eex | i < j} - VI := Vex[EI], i.e., the vertex-induced subgraph

2.3 Identity Sets: Compaction and Closure

In this step, we can further reduce the input for the closure algorithm to a more concise set of ordered pairs. We call this preparation step compaction. Assuming a lexicographic order over RDF terms, we can reduce the input for

the closure algorithm from an undirected labelled graph to a more concise set of pairs: {(x, y) | ex,y x < y}. After compaction, we partition the terms VI into different identity sets.

The partition of VI into different identity sets consists of a mapping[2] from nodes to identity sets (VI → P(VI)). For space efficiency, we store each identity set only once by associating a key with each identity set: ID →k P(VI); and map each RDF term to the key of the unique identity set that it belongs to: val: VI →v ID.

Hence, val(x) gives us the identity set ID of an RDF term x, and the composition key(val(x)) gives us the identity set of x. For partitioning VI into different identity sets, we use an incremental algorithm that parses each sorted identity pair (x, y), representing the output of the identity network compaction. The algorithm distinguishes between the following cases:

Case 1. Neither x nor y occurs in any identity set. A new identity set id is generated and assigned to both x and y: x →v id; y →v id; and id →k {x, y}

Case 2. Only x already occurs in an identity set. In this case, the existing identity set of x is extended to contain y as well: y →v val(x); val(x) →k key(val(x)) {y}

Case 3. Only y already occurs in an identity set. Similar to the previous case.

Case 4. x and y already occur, but in different identity sets. In this case, one of the two keys is chosen and assigned to represent the union of the two identity sets: val(x) →k key(val(x)) key(val(y)); (y' key(val(y)))(y' →v val(x)). This is the costliest step, especially when both identity sets are large, but it is also relatively rare due to the sorting applied as part of the compaction step. A further speedup is obtained by choosing to merge the smaller of the two sets into the larger one instead of the other way round.

---

3. IDENTITY LINKS RANKING

In the previous section, we presented our approach for calculating and storing the deductive closure of a large identity graph. In this section, we present a scalable approach for automatically assigning an error degree to each owl:sameAs statement (see [15] for more details). Our approach consists of applying an existing community detection algorithm to each identity cluster (i.e., connected component of I). Based on the detected communities in each identity cluster, an error degree is calculated and assigned to each owl:sameAs link. The error degree assigned to an owl:sameAs link depends on two factors: (a) the density of the community in which the two linked RDF terms reside, or the density of interlinks between the two communities in case the linked RDF terms are not in the same community; and (b) whether the identity link is reciprocally asserted (i.e., has a weight of 2).

This error degree is subsequently used for ranking identity links, allowing potentially erroneous links to be identified and potentially true links to be validated. We believe community detection is a particularly good fit for scalable identity error detection, since it only requires the identity network itself as input. Existing approaches require additional inputs that are not always available (e.g., resource descriptions, property mappings, vocabulary alignments) and/or that do not always scale to the LOD Cloud (e.g., crowdsourcing). Also, community detection for identity error detection does not require additional assumptions that may hold for some data sets but not others. For instance, the UNA is known not to hold for data sets constructed over longer periods and/or by larger groups of contributors.

To apply our approach on a large scale, including application to (a large scrape of) the LOD Cloud, we identify the following additional requirements for our algorithm:

Low memory footprint. Detection of erroneous identity links must not have a large memory footprint, since it must scale to very large identity networks. In addition, we want our approach to be accessible to most researchers by being able to run on a regular laptop.

Parallel computation. It must be possible to perform computation in parallel. With the increase in number of cores, even on regular laptops and consumer devices, this will allow identity link error degrees to be computed more efficiently. Preferably, error degrees can be calculated immediately after publication of an owl:sameAs link in the LOD Cloud.

Updates. Calculation must be resilient against monotonic and non-monotonic updates. Since data are added to and removed from the LOD Cloud constantly, adding or removing an owl:sameAs statement must only require re-ranking of the concerned links and must not require complete recalculation.

Our approach consists of the following main steps: (1) detect the community structure within each identity cluster, and (2) assign an error degree to each owl:sameAs link in the identity cluster. By relying on the identity network constructed in Section 2.2 and the resulting identity sets in Section 2.3, constructing the identity clusters is a straightforward phase.

3.1 Community Detection

Despite the absence of a universally agreed-upon definition, communities are typically thought of as groups that have dense connections among their members but sparse connections with the rest of the network. Community detection is a form of data analysis that seeks to automatically determine the community structure of a complex network, requiring solely information already encoded in the network topology. Detecting a network's community structure is of great importance in many concrete applications and disciplines such as computer science, biology, and sociology, where systems are often represented as graphs. This has

led to the emergence of several community detection algorithms, mostly making use of techniques from physics (e.g., spin model, optimization, random walks), as well as computer science concepts and methods (e.g., non-linear dynamics, discrete mathematics) [16]. All such techniques aim to identify groups of nodes connected "more densely" to each other than to nodes in other groups. Hence, differences between such methods ultimately come down to the precise definition of "more densely" and the algorithmic heuristic followed to identify such groups [17].

3.1.1 Community Detection Algorithms Having many community detection techniques available, we relied on existing surveys for choosing the best-performing algorithm for our task. In their 2009 survey, Lancichinetti and Fortunato [18] carried out a comparative analysis of 12 community detection algorithms that exploit some of the most interesting ideas and techniques developed in recent years. The tests were performed against benchmark graphs with heterogeneous distributions of degree and community size, including the GN benchmark [19], the LFR benchmark [20, 21], and some random graphs. This study concludes that the modularity-based method by Blondel et al. [22], the statistical inference-based method by Rosvall and Bergstrom [23], and the multi-resolution method by Ronhovde and Nussinov [24] all have excellent performance, with the additional advantage of low computational complexity. In a more recent study, Yang et al. [25] compared results of 8 state-of-the-art community detection algorithms in terms of accuracy and computing time. Interestingly, only half of these algorithms were considered in the previous survey, with tests also conducted on the LFR benchmark. This study concludes that by taking both accuracy and computing time into account, the modularity-based method by Blondel et al. [22] outperforms the other algorithms. Given that [22] outperforms the other 15 algorithms in two different studies, with an additional advantage of low computational complexity, we deploy this algorithm for detecting community structure in the owl:sameAs network. The next section presents an overview of this algorithm.

3.1.2 Louvain Algorithm Given an identity cluster Qk, the Louvain algorithm returns a set of non-overlapping communities C(Qk) = {C1, C2, ···, Cn} where: - A community C of size |C| (i.e., the number of nodes) is a subgraph of Qk such that the nodes of C are densely connected (i.e., the modularity of Qk is maximized). - $_1 \leq i \leq n$ Ci = Qk and Ci, Cj C(Qk) s.t. i ≠ j, Ci Cj = ∅.

The Louvain algorithm is a method for detecting communities in large networks. It is a greedy, non-deterministic method introduced for the general case of weighted graphs for the purpose of optimizing the modularity of partitions. The modularity of a partition is a scalar value between -1 and 1 that measures the density of links inside communities compared to links between communities. In the case of weighted networks, modularity is defined as follows:

$$Q = \frac{1}{2m} \sum_{i,j} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta(c_i, c_j)$$

where: - Aij represents the weight of the edge between nodes i and j - ki and kj represent the sum of weights of edges attached to nodes i and j, respectively - ci and cj represent the communities to which nodes i and j are assigned, respectively - m represents the sum of all edge weights in the graph - δ(u, v) is 1 if u = v and 0 otherwise

Modularity has been used to compare the quality of partitions obtained by different methods, but also as an objective function to optimize [26]. Networks with high modularity have dense connections between nodes within communities but sparse connections between nodes in different communities.

This is the intuition behind the Louvain algorithm. First, it starts by assigning a different community to each node of a given network. Then, it moves each node to one of its neighbor communities, specifically neighbors that result in the highest contribution to the modularity measure. In the next step, each community from the previous step is regarded as a single node, and the same procedure is repeated until modularity (which is always computed with respect to the original graph) no longer increases.

Although the exact computational complexity of the Louvain algorithm is not known, the method seems to run in time O(N log N), with N representing the number of nodes in the graph [22]. Exact modularity optimization is known to be NP-hard (non-deterministic polynomial-time hard), with most computational effort spent on optimization at the first level.

3.2 Error Degree Computation

After detecting the community structure in each identity cluster, our approach assigns an error degree to each identity link based on its weight and the structure of the community(ies) in which its terms occur.

We hypothesize that an identity link that is reciprocally asserted has a higher chance of being correct than a non-reciprocally asserted identity link. In addition, we hypothesize that not all resulting communities have the same quality. Based on these assumptions, we assign a lower error degree to owl:sameAs links that connect two terms within a densely connected community, or that connect two terms in two heavily interlinked communities. More precisely, to compute an error degree for each owl:sameAs link, we distinguish between two types of possible links: intra- and inter-community links.

Definition 4 (Intra-Community Link). Given a community C, an intra-community link in C, noted by eC, is a weighted edge eij where vi and vj C. We denote by EC the set of intra-community links in C.

Definition 5 (Inter-Community Link). Given two non-overlapping communities Ci and Cj, an inter-community link between Ci and Cj, noted by ijCe, is an edge eij where vi Ci and vj Cj. We denote by ijCE the set of inter-community links between Ci and Cj.

For evaluating an intra-community link, we rely on both the density of the community containing the edge and the weight of this edge. The lower the density of this community and the weight of an edge, the higher the error degree will be.

Definition 6 (Intra-Community Link Error Degree). Let eC be an intra-community link of community C. The intra-community error degree of eC, denoted by err(eC), is defined as follows:

$$err(eC) = \left(1 - \frac{2|EC|}{|C|(|C|-1)}\right) \times (2 - w(eC))$$

For evaluating an inter-community link, we rely on both the density of inter-community connections and the weight of this edge. The less the two communities are connected to each other and the lower the weight of an edge, the higher the error degree will be.

Definition 7 (Inter-Community Link Error Degree). Let ijCe be an inter-community link of communities Ci and Cj. The inter-community error degree of ijCe, denoted by err(ijCe), is defined as follows:

$$err(ijCe) = \left(1 - \frac{|ijCE|}{|Ci| \times |Cj|}\right) \times (2 - w(ijCe))$$

4. IMPLEMENTATION AND EXPERIMENTS

In the previous two sections, we presented our approach for computing and storing the identity graph and its deductive closure, as well as our approach for assigning an error degree to each owl:sameAs link. In this section, we describe how these two approaches can be combined and used in practice to deal with the Semantic Web identity problem. The overall workflow of identity network extraction, compaction, and closure is displayed in Figure 1 [Figure 1: see original paper].

4.1 Data Graph

We conduct our experiments on the LOD-a-lot data set [27], though any crawl of the LOD Cloud can be used as the starting point of our approach. We refer to this initial large-scale crawl of the LOD Cloud as our data graph (Definition 1). LOD-a-lot is the graph merge of data obtained by the large-scale crawling and cleaning infrastructure of the LOD Laundromat [28]. The result of this merge is

published in a single, publicly accessible Header Dictionary Triples (HDT) file that is 524 GB in size. This data graph contains over 28.3 billion unique triples, over 5 billion unique terms, related by over 1.1 billion unique predicates.

4.2 Explicit Identity Network Extraction

We use the LOD-a-lot HDT Dump to extract the explicit identity network (Gex), and the HDT C++ library[3] to stream the result set of the following SPARQL query to a file. This process takes around 27 minutes:

```
select distinct ?s ?p ?o {
  bind (owl:sameAs AS ?p)
  ?s ?p ?o
}
```

The results of this query are unique (keyword distinct) and the projection (?s ?p ?o) returns triples instead of pairs, so that regular RDF tools for storage and querying can be used. The explicit identity statements are stored in the order in which they are asserted by the original data publishers. This SPARQL query returns more than 558.9 million triples that connect 179.73 million terms. These owl:sameAs triples are written to an N-Triples file, which is subsequently converted to an HDT file. The HDT creation process takes almost four hours using a single CPU core. The resulting HDT file is 4.5 GB in size, plus an additional 2.2 GB for the index file automatically generated upon first use. This large collection of identity statements can be queried[4] directly from the browser or downloaded as HDT[5].

[3] https://github.com/rdfhdt/hdt-cpp
[4] https://krr.triply.cc/krr/sameas/graphs or http://sage.univ-nantes.fr/sparql/sameAs
[5] https://zenodo.org/record/1973099

4.3 Identity Network Construction

From the extracted Gex, we build the identity network (Definition 3) containing around 331M weighted edges and 179.67M terms. As a result, we leave out around 2.8M reflexive edges and around 225M duplicate symmetric edges. We also leave out 67,261 nodes that appear only in such removed edges.

4.4 Identity Sets Compaction and Closure

The identity network can be reduced to a set of sorted pairs prior to calculating the closure. For this, we use GNU sort, which takes less than 35 minutes on an SSD disk. Then, we compute the identity closure that consists of a map from nodes to identity sets. To build an efficient implementation of this key-value scheme, we need a solution that (i) uses almost no memory but is allowed to use an SSD disk, (ii) can store billions of key-value pairs, and (iii) allows such pairs to be added/removed dynamically over time. For this, we use the RocksDB[6] persistent key-value store through a SWI Prolog API[7] designed for this purpose,

allowing simultaneous reading from and writing to the database. Since changes to the identity relation can be applied incrementally, the initial creation step only needs to be performed once.

The computation of the closure takes just under 5 hours using 2 CPU cores on a regular laptop. The result is a 9.3GB on-disk RocksDB database: 2.7GB for mapping each term to an identity set ID (VI →v ID), and 6.6GB for mapping each identity set ID to its corresponding identity set (ID →k P(VI)). These files are publicly available[8].

The number of non-singleton identity sets is 48,999,148. The LOD-a-lot file, from which we construct I, contains 5,093,948,017 unique terms. This means there are 5,044,948,869 singleton identity sets in the deductive closure. Figure 2 [Figure 2: see original paper] shows that the size distribution of non-singleton identity sets is very uneven and fits a power law with exponent 3.3 ± 0.04. Around 64% of non-singleton identity sets (31,337,556 sets) contain only two terms. There are relatively few large identity sets, with the largest one containing 177,794 terms. By examining the IRIs of this identity set, we observe that it contains many terms denoting different countries, cities, things, and persons (e.g., Bolivia, Dublin, Coca-Cola, Albert Einstein, the empty string, etc.). This observation shows the need for detecting erroneous owl:sameAs links that led to such false equivalences after transitive closure.

[6] https://rocksdb.org
[7] https://github.com/JanWielemaker/rocksdb
[8] https://zenodo.org/record/3345674

4.5 Links Ranking

Relying on the constructed identity network I and the resulting 48,999,148 identity sets in the deductive closure, we reconstruct the identity clusters of I. These identity clusters represent a partitioning of I into connected components. Then, we apply the Louvain algorithm for detecting community structure in each identity cluster. As discussed in Section 3.1.2, the Louvain method is a greedy and non-deterministic algorithm, meaning that in different runs, the algorithm might produce different communities with no guarantee that the global maximum of modularity will be attained. For this reason, we ran Louvain 10 times on each identity cluster and finally considered the community structure with the highest modularity. After detecting the communities, we assign an error degree to all edges of each identity cluster.

4.5.1 Quantitative Evaluation The process of community detection and error degree computation took 80 minutes, resulting in an error degree assigned to each owl:sameAs statement in the identity network (around 556M statements after discarding reflexive ones). The error degree distribution of these statements is presented in Figure 3 [Figure 3: see original paper]. This figure shows that around 73% of statements have an error degree below 0.4, while around 5% of

owl:sameAs statements have an error degree higher than 0.8. While this distribution is mainly caused by the high number of symmetrical identity statements in the LOD Cloud, it also indicates that most identity clusters have a rather dense structure. The 179.67M terms of the identity network were assigned into a total of 55.6M communities, with community sizes varying between 2 and 4,934 terms (averaging 3 terms per community). The JAVA implementation[9] of the ranking process and the computed error degrees[10] for all owl:sameAs statements are publicly available.

[9] http://github.com/raadjoe/LOD-Community-Detection
[10] https://sameas.cc

4.5.2 Qualitative Evaluation To evaluate the accuracy of our ranking approach, we conducted several manual evaluations. This evaluation extends those conducted in [15] in terms of the number of manually evaluated links and the analyses performed.

In this evaluation, we aim to define a threshold x of the error degree, where owl:sameAs links with error degree $\leq$ x have high probability of correctness, and links with error degree $>$ x have high probability of being erroneous. To determine this threshold, four of the paper' s authors evaluated a number of owl:sameAs links. The judges relied on descriptions[11] associated with terms in the LOD-a-lot data set [27] and had no prior knowledge about each link' s error degree (i.e., whether they were evaluating a well-ranked link or not). To avoid incoherence between evaluations, judges were asked to justify all their evaluations and were given the following instructions: (a) **the same**: if two terms denote the same entity (e.g., Obama and the First Black US President); (b) **related**: not intended to refer to the same entity but closely related (e.g., Obama and the Obama Administration, or Obama and the Wikipedia article about Obama); (c) **unrelated**: not the same nor closely related (e.g., Obama and the Indian Ocean); (d) **can' t tell**: in case insufficient descriptions are available for determining the meaning of both terms (i.e., non-dereferenced IRIs appearing solely as subjects or objects of owl:sameAs statements in the LOD Cloud). Based on the judges' evaluations, we deploy the following terms:

- **True Positives (TP)**: owl:sameAs links with error degree $>$ x evaluated by judges as incorrect links (related or unrelated).
- **False Positives (FP)**: owl:sameAs links with error degree $>$ x evaluated by judges as true identity links.
- **True Negatives (TN)**: owl:sameAs links with error degree $\leq$ x evaluated by judges as true identity links.
- **False Negatives (FN)**: owl:sameAs links with error degree $\leq$ x evaluated by judges as incorrect links (related or unrelated).

An approach for erroneous link detection can be evaluated using the classic measures of precision, recall, and accuracy defined as follows:

$$precision = \frac{TP}{TP + FP}$$

$$recall = \frac{TP}{TP + FN}$$

$$accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

We consider that when a Semantic Web expert cannot confirm the correctness of a certain identity link due to absence of necessary descriptions for one of the two involved terms, an automated approach will have the same limitations. Following this assumption, we do not consider links judged by experts as "can' t tell" in accuracy, precision, and recall evaluations.

First, judges evaluated 200 owl:sameAs statements (50 each), representing a sample from each bin of the error degree distribution shown in Figure 3. The main goal of this first evaluation of a small set of links is to test whether the defined error degree is indeed an indicator for correctness (i.e., whether the probability of finding erroneous identity links increases with error degree).

From the results presented in Table 1 , we observe the following: - The higher an error degree, the more likely the link is erroneous. - 100% of evaluated links with error degree ≤ 0.4 are correct. - When error degree is between 0.4 and 0.8, 83.3% of owl:sameAs links are correct, while in 13.3% of cases such links might have been used to refer to two different but related terms. - An owl:sameAs with error degree > 0.8 is a less reliable identity statement, referring in 31.8% of cases to two different terms.

By taking the highest available threshold in this table (x = 0.8), the evaluation suggests a precision of 31.8% (15/(7+15)) and an accuracy of 81.6% (97/(97+12+7+15)). This suggests a higher threshold is required to detect erroneous owl:sameAs statements with higher precision. Therefore, we increased the threshold to 0.99 and manually evaluated additional samples to better assess accuracy and precision.

The 200 owl:sameAs statements evaluated in Table 1 represent our first sample. We briefly describe three additional samples and their purposes:

Sample 2. Represents a set of 60 owl:sameAs statements with error degrees between 0.9 and 1, randomly chosen from identity clusters of various sizes. We note that 21 of these owl:sameAs were judged as "can' t tell" by the judges.

Sample 3. In 2013, the authors of [13] manually evaluated 95 owl:sameAs statements linking Freebase with DBpedia terms, and all were judged as correct. This is the only publicly available external gold standard from the LOD Cloud. Out of these 95 links, only 78 were found in our data graph. Verifying their

error degrees, only one link was assigned an error degree higher than 0.99 (FP), with the other 77 links having error degrees ranging from 0.52 to 0.94 (TN).

Sample 4. To evaluate the recall of our approach, we verified how it can rank newly introduced erroneous owl:sameAs statements. First, we chose 40 random terms from the identity network, ensuring all these terms are different and not explicitly owl:sameAs[12] (e.g., dbr:Paris, dbr:Strawberry, dbr:Facebook). From the 40 selected terms, we generated all 780 possible undirected edges between them. We added each edge eij separately to the identity network with w(eij) = 1, calculated its error degree, and removed it from the identity network before adding the next one. The resulting error degrees of newly introduced erroneous identity links range from 0.87 to 1. When the threshold is fixed at 0.99, the recall for detecting erroneous identity links is 93%, with 725 (TP) out of 780 added links (TP+FN) having error degree > 0.99.

Evaluation of these owl:sameAs samples is presented in Table 2 with a threshold of 0.99. The results suggest our approach can classify an identity link (as correct or erroneous) with 91.9% accuracy when the threshold is set at 0.99. We are aware that accuracy can be misleading in imbalanced data sets, as it fails to detect approaches biased toward the dominated class. However, in the case of the four samples we considered, we believe accuracy is a good estimation of how this automatic approach matches human judgment, since it performs well on samples dominated by correct owl:sameAs (e.g., sample 3) and samples dominated by erroneous links (e.g., sample 4).

[12] But including some terms that currently belong to the same identity set.

---

5. FIXING OWL:SAMEAS IN THE LOD CLOUD

This section combines our approach for computing the transitive closure of owl:sameAs (Section 2) with our approach that assesses the quality of owl:sameAs (Section 3). Specifically, we use the error degrees computed in Section 4.5 to generate new, higher quality identity networks and sets in the LOD Cloud. This section makes the following contributions: - Presents two new identity networks, with the first discarding "potentially erroneous" owl:sameAs and the second containing only "high quality" owl:sameAs. - Presents the transitive closure of the two newly constructed identity networks. - Presents a use case and benchmark for evaluating advantages and limitations of using these new identity networks and their deducted identity sets.

5.1 Link Selection Criteria

To classify an owl:sameAs as "potentially erroneous" or of "high quality," we rely on manually evaluated links previously described in Table 1 and Table 2. Specifically, Table 1 shows that 100% of manually evaluated links with error degree ≤ 0.4 are correct (57/57), suggesting that links with such error degrees

can be classified as "probably correct." Additionally, the manual evaluation in Table 2 shows our approach has high accuracy when the threshold is fixed at 0.99 (~92% accuracy), suggesting that identity links with error degree > 0.99 can be classified with high confidence as "potentially erroneous." To test these two hypotheses, we construct two new subgraphs of the identity network with the following characteristics:

1. All owl:sameAs statements in the identity network with error degree > 0.99 are discarded.
2. All owl:sameAs statements in the identity network with error degree > 0.4 are discarded.

Below, we present quantitative and qualitative evaluation of these two identity subgraphs. We also generate and evaluate their corresponding identity sets and discuss their impact. These new subgraphs, with their deducted identity sets, are publicly available[13].

[13] https://zenodo.org/record/3345674

5.2 Quantitative Evaluation

Figure 4 [Figure 4: see original paper] compares the number of identity pairs, terms, and resulting identity sets between the closure of (a) the original identity network containing all 331M identical pairs[14] in the LOD Cloud; (b) a subset containing only links with error degree ≤ 0.99; and (c) a subset containing only links with error degree ≤ 0.4. First, this figure shows that the number of RDF terms belonging to non-singleton identity sets decreases from 179.67M in the original closure to 179.34M in the closure, indicating that about 330K RDF terms are solely involved in "potentially erroneous" owl:sameAs statements. Additionally, this evaluation shows that discarding about 500K pairs with error degree > 0.99 results in about 75K additional non-singleton identity sets. These sets result from partitioning several large—probably incorrect—identity sets. On the other hand, considering only the 205K "highly probable" identical pairs in the LOD Cloud (i.e., linked by owl:sameAs with error degree ≤ 0.4) decreases the number of resulting non-singleton identity sets by more than 70% (from 48.9M non-singleton identity sets in the original closure to 15.9M). This result is expected since only 53M out of 179M terms in the LOD Cloud (about 30%) are involved in identity statements with such high confidence.

[14] Note that in the identity network, reflexive and duplicate symmetric links are discarded.

To analyze closely the impact of these new computed closures, we focus first on the largest identity set. This identity set originally includes 177,794 terms connected by 2.8M edges that result from compaction of 5.5M owl:sameAs statements (about 1% of the total). This identity set includes terms referring to many different real-world entities falsely claimed to be the same (explicitly or implicitly). When owl:sameAs links with error degree > 0.99 are discarded (re-

spectively discarding links with error degree > 0.4), the terms of this identity set are partitioned into 1,086 non-singleton identity sets (resp. 640 sets), with an average of 160 terms per set (resp. 50 terms). The largest derived non-singleton identity set contains 3,686 terms (resp. 157 terms), and the smallest set in each closure contains only two terms. As a result of discarding links with error degree > 0.99, we find that 4,146 terms originally belonging to the largest identity set (2.3% of terms) now appear in singleton identity sets. On the other hand, discarding links with error degree > 0.4 results in 145,934 terms originally belonging to the largest identity set (82% of terms) being present in singleton identity sets (i.e., not identical to any other term).

Finally, in terms of computational efficiency, the two new deductive closures are much faster to compute. Compared to the original closure that takes about 5 hours using two CPU cores on a regular laptop, closures (b) and (c) take about 2.5 hours and about 1 hour, respectively. This significant difference in computation time between closure (a) and (b), despite the relatively small difference in included pairs (about 500K pairs), is consistent with our observation in Section 2.3 that computation time for calculating deductive closure is dominated by time needed to merge large identity sets.

5.3 Qualitative Evaluation

Thus far, we have computed two additional equivalence closures based on discarding owl:sameAs with certain error degrees and compared these new subgraphs to the original identity graph in terms of number of resulting identity sets, number of terms included in non-singleton identity sets, and computation time for deducing their identity sets. In this section, we analyze the quality of these subgraphs and compare it to the quality of the original identity graph (i.e., all owl:sameAs links in the LOD Cloud).

For this qualitative analysis, we examine the specific use case of the identity cluster containing the DBpedia term for former US president Barack Obama (dbr:Barack_{Obama}). This identity cluster, presented in Figure 5 [Figure 5: see original paper], includes 440 terms connected by 7,615 edges. These edges result from compaction of 14,917 owl:sameAs statements. Applying the Louvain algorithm to this identity cluster results in four non-overlapping communities[15], presented in Figure 6 [Figure 6: see original paper].

[15] https://www.sameas.cc/explicit/obama-class.svg

To analyze the impact of discarding links with certain error degrees, the authors manually annotated each RDF term in this identity set. By dereferencing IRIs and examining their descriptions in the LOD-a-lot data set, the authors annotated 338 out of 440 RDF terms. These 338 terms were judged to refer to 8 different real-world entities (e.g., Barack Obama the person, his presidency, his senate career). The other 102 RDF terms lack sufficient descriptions for confident annotation and are discarded for the remainder of this evaluation. With all terms in this identity set annotated, we conduct two complementary evalu-

ations. The first evaluation at the link level (Section 5.3.1) assesses how accurately our approach can classify each identity link based on different thresholds. The second evaluation at the closure level (Section 5.3.2) assesses the impact of discarding few identity links on overall closure quality. Both code and data necessary for replicating these analyses are publicly available[16].

[16] https://github.com/raadjoe/obama-lod-identity-analysis

5.3.1 "Barack Obama" Identity Cluster–Links Evaluation The identity cluster containing the DBpedia term for former US president "Barack Obama" contains 14,917 owl:sameAs statements between 440 RDF terms. After discarding the 102 terms that could not be manually annotated and their corresponding links[17], this identity cluster now contains 14,613 owl:sameAs statements linking 338 RDF terms. Based on our manual annotation, we find 26 owl:sameAs statements that link two RDF terms referring to different entities (0.17% of links are erroneous). This evaluation shows that even though this identity cluster mostly contains correct owl:sameAs links, the presence of a small number of erroneous ones led to false equivalence of RDF terms referring to 8 different real-world entities.

[17] Note these terms were not discarded during error degree computation, only in the evaluation.

Table 3 presents evaluation of our approach in classifying owl:sameAs links in this identity cluster according to thresholds 0.4 and 0.99. From this table, we observe only two owl:sameAs links with error degree > 0.99, both of which are indeed erroneous (TP=2 and FP=0 in closure b). These results suggest 100% precision (2/2) and 7.69% recall (2/26) for our approach when threshold is fixed at 0.99. Conversely, when threshold is set at 0.4, recall increases to 100% (i.e., all 26 erroneous links have error degree > 0.4), with precision decreasing to 3.16% (due to 795 additional flagged links that are actually correct). For both thresholds, accuracy is considerably high, as expected for an imbalanced data set.

5.3.2 "Barack Obama" Identity Cluster–Closure Evaluation The previous evaluation showed that the "Barack Obama" identity cluster in the LOD Cloud contains 26 erroneous owl:sameAs statements that led to false equivalence of RDF terms referring to 8 different real-world entities. Table 3 shows that depending on the chosen threshold, a Linked Data practitioner can decide whether to remove only a few erroneous owl:sameAs statements without removing any correct ones (i.e., closure b), or remove all erroneous owl:sameAs but simultaneously discard many correct ones (i.e., closure c). This section evaluates the impact of both approaches on the overall closure and compares it to the original closure of all owl:sameAs links in the LOD-a-lot data set.

Table 4 presents differences in resulting identity sets between the three closures, where we observe:

Gold Standard. Our manual annotation shows that the 338 RDF terms in the original identity cluster refer to 8 different real-world entities. Ideally, we expect these terms to be partitioned into 8 identity sets {GS1, ···, GS8}, with GS1 containing 260 RDF terms referring to the person Barack Obama, GS2 containing 47 terms referring to his presidency, and so forth.

Closure (a). The original closure of all owl:sameAs links in LOD-a-lot results in all 338 RDF terms belonging to one identity set A1. These 338 terms are explicitly connected by 14,613 owl:sameAs links, resulting in 56,953 distinct pairs.

Closure (b). After discarding two owl:sameAs links with error degree > 0.99, the closure of the remaining 14,611 owl:sameAs links partitions these 338 RDF terms into two non-singleton identity sets B1 and B2. The former contains 270 RDF terms and the latter contains the remaining 68 terms.

Table 4 shows that by solely discarding two erroneous owl:sameAs links from this identity cluster, our approach separated all terms referring to Obama' s presidency and presidential transition[18] from the rest. However, despite this significant quality improvement, both identity sets remain logically inconsistent (i.e., contain terms referring to different real-world entities).

Closure (c). After discarding 821 owl:sameAs links with error degree > 0.4, the closure of the remaining 13,792 correct identity links partitions the 338 RDF terms into 219 identity sets. Of these, only identity set C1 is non-singleton, containing 120 RDF terms referring to the person Barack Obama. Although this closure results in many identity sets (compared to the expected 8), the resulting closure is now semantically correct.

[18] Terms referring to the period when Obama won the US election in November 2008 until his inauguration in January 2009 when his presidency started.

To further evaluate the quality of these three closures, we rely on classic evaluation measures of precision, recall, and accuracy. However, since this evaluation is conducted at the level of resulting identity clusters, we redefine True/False Positives/Negatives in terms of the number of remaining correct/erroneous distinct pairs (instead of explicit owl:sameAs). All possible 56,953 distinct pairs X between the 338 manually annotated RDF terms represent the universe of discourse in this evaluation.

- **True Positives (TP).** Pairs (a,b) X where a and b are in the same identity cluster in both the gold standard and resulting closure.
- **False Positives (FP).** Pairs (a,b) X where a and b are in the same identity cluster in the resulting closure but in different clusters in the gold standard.
- **True Negatives (TN).** Pairs (a,b) X where a and b are in different clusters in both the gold standard and resulting closure.
- **False Negatives (FN).** Pairs (a,b) X where a and b are in the same identity cluster in the gold standard but in different clusters in the closure.

Table 5 presents recall, precision, and accuracy evaluation of each closure. For closure (a), we expect 100% recall since all 338 considered terms originally belong to the same identity cluster. Since ideally these terms should be partitioned into 8 clusters, this table shows the original closure of all owl:sameAs links contains 21,961 erroneous pairs, resulting in 61% precision. For closure (c), we also expect 100% precision since all pairs (a,b) in the same resulting identity cluster are indeed identical (TP). This can also be observed in Table 4, where all identity sets resulting from closure (c) are either singleton or, in the case of C1, refer solely to the person Obama. However, since it partitions the 338 RDF terms into 219 identity clusters (where the ideal partition is 8), recall is significantly lower (20%) due to many correct pairs separated in this closure (FN). Finally, closure (b) achieves both high recall (99.9%) and high precision (90.6%) due to many non-identical pairs separated (TN) and many identical pairs kept in the same cluster (TP) after discarding only two owl:sameAs links with error degree > 0.99. For instance, Table 4 shows all 260 terms referring to the person Obama remain in cluster B1, separated from all 47 terms referring to his presidency.

---

6. CONCLUSION

In this paper, we showed that even though a global knowledge graph like the LOD Cloud may contain incorrect identity statements, it is still possible to use subgraphs productively. This is done by extracting all owl:sameAs links, computing their transitive closure, and assigning an error degree to each link based on the identity cluster' s community structure. Since the presented approach only takes a couple of hours to compute over a large scrape of the LOD Cloud (over 500 million explicit and 35 billion implicit owl:sameAs statements) while using a regular laptop, this trade-off between identity subgraph size and link validity can be used in practice. Our qualitative evaluation on a benchmark of about 1K owl:sameAs links and a manually annotated identity cluster containing about 14K owl:sameAs links suggests that discarding a small number of links with error degree higher than 0.99 (about 0.17% of links) can significantly enhance closure quality (correctness increases from 61% to 93%). Our results also suggest that discarding a larger number of links with error degree higher than 0.4 (about 27% of links) can lead to semantically valid identity clusters (i.e., clusters that do not contain two terms referring to different real-world entities). These results could be further improved by combining or comparing results from multiple community detection methods.

Finally, based on the level of correctness required within a given application, a data practitioner may choose a different error degree, resulting in a smaller or larger subgraph depending on whether erroneous identity statements are considered more or less acceptable. Additionally, some applications might requalify links with certain error degrees to a less strict notion of identity, such as that encoded in skos:closeMatch. This would allow practitioners to benefit from these asserted links while reducing semantic inconsistencies when reasoning is

applied. We believe automated and scalable data selection approaches like the one presented here will see uptake over time, as they allow data practitioners to extract maximal value from the LOD Cloud, thereby making the Semantic Web succeed as an integrated and reliable knowledge space.

REFERENCES

[1] A. Hogan, A. Zimmermann, J. Umbrich, A. Polleres & S. Decker. Scalable and distributed methods for entity matching, consolidation and disambiguation over linked data corpora. *Web Semantics: Science, Services and Agents on the World Wide Web* 10(2012), 76–110. doi:10.1016/j.websem.2011.11.002.

[2] J. Raad. *Identity management in knowledge graphs*. PhD dissertation, University of Paris-Saclay, 2018. Available at: https://tel.archives-ouvertes.fr/tel-02073961.

[3] H. Halpin, P.J. Hayes, J.P. McCusker, D.L. McGuinness & H.S. Thompson. When owl:sameAs isn' t the same: An analysis of identity in Linked Data. In: *International Semantic Web Conference*, 2010, pp. 305–320. doi:10.1007/978-3-642-17746-0_{20}.

[4] J. Raad, N. Pernelle, F. Saïs, W. Beek & F. van Harmelen. The sameas problem: A survey on identity management in the web of data. arXiv preprint. arXiv:1907.10528, 2019.

[5] P. Bouquet, H. Stoermer & D. Giacomuzzi. Okkam: Enabling a web of entities. In: *CEUR Workshop Proceedings*, 2007, pp. 1–8. Available at: http://www.ceur-ws.org/Vol-249/submission_{150}.pdf.

[6] H. Glaser, A. Jaffri & I. Millard. Managing co-reference on the semantic web. In: *WWW2009 Workshop: Linked Data on the Web LDOW*, 2009, pp. 1–6. Available at: http://www.ceur-ws.org/Vol-538/ldow2009_{paper11}.pdf.

[7] M. Mountantonakis & Y. Tzitzikas. On measuring the lattice of commonalities among several linked data sets. In: *Proceedings of the VLDB Endowment*, 2016, pp. 1101–1112. doi:10.14778/2994509.2994527.

[8] J. Cuzzola, E. Bagheri & J. Jovanovic. Filtering inaccurate entity co-references on the linked open data. In: *International DEXA Conference*, 2015, pp. 128–143. doi:10.1007/978-3-319-22849-5_{10}.

[9] G. de Melo. Not quite the same: Identity constraints for the web of linked data. In: *The Twenty-Seventh AAAI Conference on Artificial Intelligence*, 2013, pp. 1092–1098. Available at: http://www.aaai.org/ocs/index.php/AAAI/AAAI13/paper/download/6313/6872

[10] A. Valdestilhas, T. Soru & A.C.N. Ngomo. Cedal: Time-efficient detection of erroneous links in large-scale link repositories. In: *International Conference on Web Intelligence*, 2017, pp. 106–113. doi:10.1145/3106426.3106497.

[11] L. Papaleo, N. Pernelle, F. Saïs & C. Dumont. Logical detection of invalid sameas statements in RDF data. In: *International Conference EKAW*, 2014, pp. 373–384. doi:10.1007/978-3-319-13704-9_{29}.

[12] C. Guéret, P. Groth, C. Stadler & J. Lehmann. Assessing linked data mappings using network measures. In: *Extended Semantic Web Conference*, 2012, pp. 87–102. doi:10.1007/978-3-642-30284-8_{13}.

[13] M. Acosta, A. Zaveri, E. Simperl, D. Kontokostas, S. Auer & J. Lehmann. Crowdsourcing linked data quality assessment. In: *International Semantic Web Conference*, 2013, pp. 260–276. doi:10.1007/978-3-642-41338-4_{17}.

[14] W. Beek, J. Raad, J. Wielemaker & F. van Harmelen. sameas.cc: The closure of 500m owl:sameAs statements. In: *Extended Semantic Web Conference*, 2018, pp. 65–80. doi:10.1007/978-3-319-93417-4_5.

[15] J. Raad, W. Beek, F. van Harmelen, N. Pernelle & F. Saïs. Detecting erroneous identity links on the web using network metrics. In: *International Semantic Web Conference*, 2018, pp. 391–407. doi:10.1007/978-3-030-00671-6_{23}.

[16] S. Fortunato. Community detection in graphs. *Physics Reports* 486(2010), 75–174. doi:10.1016/j.physrep.2009.11.002.

[17] M.A. Porter, J.P. Onnela & P.J. Mucha. Communities in networks. *Notices of the AMS* 56(2009), 1082–1097. doi:10.1016/j.cnsns.2012.03.023.

[18] A. Lancichinetti & S. Fortunato. Community detection algorithms: A comparative analysis. *Physical Review E* 80(2009), 056117. doi:10.1103/PhysRevE.80.056117.

[19] M. Girvan & M.E. Newman. Community structure in social and biological networks. In: *Proceedings of the National Academy of Sciences*, 2002, pp. 7821–7826. doi:10.1073/pnas.122653799.

[20] A. Lancichinetti, S. Fortunato & F. Radicchi. Benchmark graphs for testing community detection algorithms. *Physical Review E* 78(2008), 046110. doi:10.1103/PhysRevE.78.046110.

[21] A. Lancichinetti & S. Fortunato. Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities. *Physical Review E* 80(2009), 016118. doi:10.1103/PhysRevE.80.016118.

[22] V.D. Blondel, J. Guillaume, R. Lambiotte & E. Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment* 10(2008), 10008. doi:10.1088/1742-5468/2008/10/P10008.

[23] M. Rosvall & C.T. Bergstrom. Maps of random walks on complex networks reveal community structure. In: *Proceedings of the National Academy of Sciences*, 2008, pp. 1118–1123. doi:10.1073/pnas.0706851105.

[24] P. Ronhovde & Z. Nussinov. Multiresolution community detection for megascale networks by information-based replica correlations. *Physical Review*

E 80(2009), 016109. doi:10.1103/PhysRevE.80.016109.

[25] Z. Yang, R. Algesheimer & C.J. Tessone. A comparative analysis of community detection algorithms on artificial networks. *Scientific Reports* 6(2016), 30750. doi:10.1038/srep30750.

[26] M.E.J. Newman & M. Girvan. Finding and evaluating community structure in networks. *Physical Review E* 69(2004), 026113. doi:10.1103/PhysRevE.69.026113.

[27] J.D. Fernández, W. Beek, M.A. Martínez-Prieto & M. Arias. Lod-a-lot. In: *International Semantic Web Conference*, 2017, pp. 75–83. doi:10.1007/978-3-319-68204-4_7.

[28] W. Beek, L. Rietveld, H.R. Bazoobandi, J. Wielemaker & S. Schlobach. Lod laundromat: A uniform way of publishing other people' s dirty data. In: *International Semantic Web Conference*, 2014, pp. 213–228. doi:10.1007/978-3-319-11964-9_{14}.

AUTHOR BIOGRAPHIES

Joe Raad is a post-doctoral researcher in the Knowledge Representation & Reasoning group at Vrije Universiteit Amsterdam, The Netherlands. His research focuses on identity management in Knowledge Representation systems, large-scale empirical study of semantics, and semantic technologies deployment for Digital Humanities. As part of his research, he co-developed sameAs.cc and MetaLink. ORCID: 0000-0002-7891-7738

Wouter Beek is a post-doctoral researcher in the Knowledge Representation & Reasoning group at Vrije Universiteit Amsterdam and co-founder of Triply. He is interested in the Semantic Web as a platform for knowledge-intensive applications, deployment of large-scale knowledge bases for innovative reuse, and the interaction between Web semantics and pragmatics, including empirical study of semantics. As part of his research, he co-developed the LOD Laundromat, LOD Search, and sameAs.cc. ORCID: 0000-0003-0250-9655

Frank van Harmelen is a Professor in Knowledge Representation & Reasoning in the Computer Science department (Faculty of Science) at Vrije Universiteit Amsterdam, The Netherlands. Since 2000, he has played a leading role in Semantic Web development. He was co-PI on the first European Semantic Web project (OnToKnowledge, 1999), which laid foundations for the Web Ontology Language OWL. OWL has become a worldwide standard, is in wide commercial use, and forms the basis for an entire research community. He co-authored the *Semantic Web Primer*, the first academic textbook in the field, now in its third edition and in worldwide use (translations in 5 languages, 10,000 copies sold of English edition alone). He was one of the architects of Sesame, an RDF storage and retrieval engine in wide academic and industrial use with over 200,000

downloads. This work received the 10-year impact award at the 11th International Semantic Web Conference in 2012, the most prestigious award in the field. In recent years, he pioneered development of large-scale reasoning engines. He was scientific director of the 10m euro EU-funded Large Knowledge Collider, a platform for distributed computation over semantic graphs with billions of edges. The prize-winning work with his student Jacopo Urbani improved the state of the art by two orders of magnitude. He is scientific director of The Network Institute, an interdisciplinary research institute where 150 researchers from Faculties of Social Science, Humanities, and Computer Science collaborate on computational Social Science and e-Humanities. He is a fellow of the European AI Society ECCAI (membership limited to 3% of European AI researchers), was admitted to Academia Europaea in 2014 (limited to top 5% of researchers in each field), and admitted as Member of the Royal Netherlands Society of Sciences and Humanities in 2015 (450 members across all sciences). ORCID: 0000-0002-7913-0048

Jan Wielemaker is a senior researcher at Vrije Universiteit Amsterdam (VUA) and the Centrum voor Wiskunde en Informatica (CWI, The Netherlands). He is the lead developer of SWI-Prolog and responsible for development and maintenance of linked data infrastructure libraries for SWI-Prolog. ORCID: 0000-0001-5574-5673

Nathalie Pernelle is a Professor of Computer Science, member of LIPN (Laboratoire d' Informatique de Paris Nord), at the University of Paris in France, where she moved from Paris-Saclay University in 2019. Her research relates to knowledge discovery in data graphs. She has particularly studied models and algorithms for data linking, rule mining, and semantic annotation of unstructured documents. She has been involved in many academic and industrial projects across domains such as biological or geographical data, bibliographical knowledge bases, asbestos diagnoses, or problems related to the General Data Protection Regulation. ORCID: 0000-0003-1487-393X

Fatiha Saïs is currently an associate Professor - HDR at the Computer Science Research Laboratory (LRI) of Paris Saclay University, France. She is co-head of LaHDAK group (Large-scale Heterogeneous Data and Knowledge). Her research focuses on: identity management in the Web of data; knowledge graph fusion; knowledge discovery from RDF graphs; and more recently veracity assessment in knowledge graphs. Her work has been included in more than 20 national, industrial, and European research projects. She has published over 60 research papers in national and international conferences and journals such as ISWC (International Semantic Web Conference), *Journal of Web Semantics*, and *Journal of Data Semantics*. She served as PC member for international conferences (ECAI, ESWC, K-Cap, ICCS, etc.), national conferences (EGC, IC, BDA), and organized and chaired several national and international workshops and conferences (WebToTouch, EGC, Verita, SoWedo, JDSE, etc.). ORCID: 0000-0002-6995-2785

Note: Figure translations are in progress. See original paper for figures.