sameAs.cc: The Closure of 500M owl:sameAs Statements

Wouter Beek, Joe Raad, Jan Wielemaker, Frank van Harmelen

sameAs.cc: The Closure of 500M `owl:sameAs` Statements

Wouter Beek[1], Joe Raad[2,3], Jan Wielemaker[1], and Frank van Harmelen[1]

[1] Dept. of Computer Science, VU University Amsterdam, NL
{w.g.j.beek, J.Wielemaker, Frank.van.Harmelen}@vu.nl
[2] UMR MIA-Paris, AgroParisTech, INRA, Paris-Saclay University, Paris, France
[3] LRI, Paris-Sud University, CNRS 8623, Paris-Saclay University, Orsay, France
joe.raad@agroparistech.fr

Abstract. The `owl:sameAs` predicate is an essential ingredient of the Semantic Web architecture. It allows parties to independently mint names, while at the same time ensuring that these parties are able to understand each other's data. An online resource that collects all `owl:sameAs` statements on the Linked Open Data Cloud has therefore both practical impact (it helps data users and providers to find different names for the same entity) as well as analytical value (it reveals important aspects of the connectivity of the LOD Cloud).
This paper presents sameAs.cc: the largest dataset of identity statements that has been gathered from the LOD Cloud to date. We describe an efficient approach for calculating and storing the full equivalence closure over this dataset. The dataset is published online, as well as a web service from which the data and its equivalence closure can be queried.

Keywords: linked open data, identity, reasoning

1 Introduction & related work

The absence of a central naming authority for minting IRIs is essential to the architecture of the Semantic Web. Just as on the regular Web [13], allowing different organizations and individuals to mint their own IRIs, without the bottleneck of centralized coordination, is a precondition for the Semantic Web to scale, and has been a deliberate motivation in the design of OWL [1].

However, the absence of any central authority also makes it impossible to enforce the Unique Name Assumption on the Semantic Web. Despite attempts such as OKKAM to encourage the re-use of existing IRIs [3], the same thing is often denoted by many names on the Semantic Web. Because of this, the need arose to state that two names, possibly minted by different organizations or individuals, denote the same thing. For this purpose OWL introduced `owl:sameAs` [11]: the statement $\langle x, \texttt{owl:sameAs}, y\rangle$ asserts that x and y denote the same thing, formalized as follows (with I the interpretation function):

$$I(\langle x, \texttt{owl:sameAs}, y\rangle) \text{ is true iff } I(x) = I(y)$$

Such identity management is not merely a luxury but is essential for the Semantic Web. It allows parties to independently mint names, while at the same time ensuring that these parties are able to understand each other's data. In other words, `owl:sameAs` statements are an important part of the glue that connects different datasets on the Semantic Web, and they are indeed the most often used linking predicate across many domains on the Semantic Web [15].

1.1 Related Work

The special status of `owl:sameAs` links has motivated earlier studies into the use of these links on the Semantic Web, as well as the construction of specialized services to harvest and publish these links. An early analysis of the use of `owl:sameAs` links was performed by Ding et al. [6] in 2010, which extracted 8.7M `owl:sameAs` links from the 2010 Billion Triple Challenge dataset, resulting in a graph of 2.9M weakly connected components, most of which are very small (average size 2.4), only 41 components with hundreds of IRIs, and only two components with thousands of IRIs, the largest of which has size 5,000.

In the same year, Halpin et al. [9], retrieved 58M `owl:sameAs` links from 1,202 domains in the 2010 Linked Open Data Cloud, and provided an aggregated analysis at the level of datasets.

A later analysis, by Schmachtenberg et al. [15] from 2014, crawled 1,014 datasets containing 8B resources, and again analyzed the use of `owl:sameAs` links at the aggregation level of datasets. The entire graph of datasets was found to consist of 9 weakly connected components, the largest one contained 297 datasets, with `dbpedia.org` having the largest in-degree, with 89 datasets containing `owl:sameAs` links to it. This work is similar in spirit to ours, but we advance over it by (i) using a bigger corpus, (ii) analyzing the `owl:sameAs` graph at the level of individual resources instead of datasets (a graph of over 500M `owl:sameAs` links), and most importantly, (iii) computing and publishing the closure of this massive graph. Also in 2014, Schlegel et al. [14] queried 200 SPARQL endpoints to obtain 17.6M `owl:sameAs` links over 2.4M IRIs for which they did compute the transitive closure, obtaining 8.4M equivalence classes. The dataset and analysis we present in this paper is an order of magnitude larger.

The largest collection of RDF links hosted to date is at `http://sameas.org`. It provides a Consistent Reference Service (CRS) [8] over an impressive number of 203M IRIs that are combined into 62.6M 'identity bundles' based on 345M triples[4]. As such, it is the main predecessor of the work presented in this paper. However, the `sameas.org` collection mixes identity pairs (linked with the `owl:sameAs` property) together with pairs that are not identity pairs (linked with other properties, such as `umbel:isLike`, `skos:exactMatch`, and `owl:inverseOf`). This means that the overall closure is not semantically sound. The crucial difference with our work is that the identity closure that we calculate is *semantically interpretable*, because it is exclusively based on `owl:sameAs`

[4] Up-to-date numbers obtained by personal communication from Hugh Glaser.

statements, and the computed closure adheres to the OWL semantics. As a result, our dataset can be used by a DL reasoner in order to infer new facts.

Finally, the Schema.org vocabulary[5] includes the `schema:sameAs` property. However, the semantics of this property is substantially different from that of `owl:sameAs`. It states that two terms "are two pages with the same topic" and does not express equality.

1.2 Contributions & structure of this paper

This paper makes the following three contributions:

1. It presents the largest downloadable dataset of identity statements that have been gathered from the LOD Cloud to date, including its equivalence closure. The dataset and its closure are also exposed through a web service.
2. It gives an in-depth analysis of this dataset, its closure, and its aggregation to datasets.
3. It presents an efficient approach for extracting, storing, and calculating the identity statements and their equivalence closure. Even though the dataset and closure are quite large, they can be stored on a USB stick and queried from a regular laptop.

In Section 2 we discuss the requirements that a semantically interpretable `owl:sameAs` dataset and web service must satisfy. Section 3 describes the algorithm and implementation for calculating and storing the explicit and implicit identity relations that fulfills the requirements. Section 4 gives an analysis of some of the key properties of our dataset. Section 5 describes the sameAs.cc dataset and web service, and Section 6 concludes.[6]

2 Requirements

2.1 Preliminaries

We distinguish between two identity relations. The *explicit identity relation* ($\sim_e$) is the set of pairs (x, y) for which a statement $\langle x, \texttt{owl:sameAs}, y\rangle$ has been asserted in a publicly accessible dataset. The *implicit identity relation* ($\sim_i$) is the explicit identity relation closed under equivalence (reflexivity, symmetry and transitivity).

[5] `https://schema.org`

[6] This paper uses the following RDF prefixes for brevity:

`dbr`: `http://dbpedia.org/resource/`
`owl`: `http://www.w3.org/2002/07/owl#`
`rdf`: `http://www.w3.org/1999/02/22-rdf-syntax-ns#`
`rdfs`: `http://www.w3.org/2000/01/rdf-schema#`
`xsd`: `http://www.w3.org/2001/XMLSchema#`

Let N denote the set of RDF nodes: the RDF terms that appear in the subject- or object position of at least one triple. A *partitioning* of N is a collection of non-empty and mutually disjoint subsets $N_k \subseteq N$ (called *partition members*) that together cover N. We leave it to the reader to verify that the relations $\sim_e$ and $\sim_i$ both induce a partitioning of N when taking their connected components as the partition members. Adopting terminology from [8], we call these partition members *equality sets*, and the partition members of $\sim_i$ *identity sets*. The equality set of a term x is denoted $[x]$. Each equality set of $\sim_e$ is a connected directed graph; each equality set of $\sim_i$ is a fully connected graph.

2.2 Requirements

In order to calculate $\sim_i$, we have to close $\sim_e$ under equivalence. Existing approaches do not scale, due to multiple dimensions of complexity:

$\sim_i$ can be too large to store In Section 4, we will see that the LOD Cloud contains equality sets with cardinality well over 100K. It is not feasible to store the materialization of $\sim_i$, since the space consumption of that approach is quadratic in the size of the equality set. (E.g., the materialization of an equality set of 100K terms contains 10B identity pairs.)
We will not store the materialization, but the equality sets themselves, which is only linear in terms of the size of the universe of discourse (i.e., the set N of RDF nodes).

$|N_k|$ can be too large to store Even the number of elements within one equality set can be too large to store in memory. The current version of the resource already contains equality sets that contain over 100K terms. Since our calculation of $\sim_i$ must have a low hardware footprint and must be future proof, we will not assume that every individual equality set will always be small enough to fit in memory.

***sim*$_e$ changes over time** We calculate the identity closure for a large snapshot of the LOD Cloud. Since datasets in the LOD cloud are constantly changing, and datasets are constantly added, we want to update $\sim_i$ incrementally, allowing for both additions and deletions, without having to recompute the entire closure.

Even though applications of a LOD Cloud-wide identity service are beyond the scope of this paper, there are many use-cases for such a service:

Findability of backlinks Since the Semantic Web does not allow backlinks to be followed (an architectural property it shares with the World Wide Web), it is only possible to follow outgoing `owl:sameAs` links but not incoming ones. An identity service retrieves all IRIs that are linked through `owl:sameAs` links, and thereby allows the full set of assertions about a given resource to be retrieved from across the LOD Cloud.

Query answering A special case of the findability of links arises in distributed query answering over the LOD Cloud, which requires an overview of existing alignments between concepts and individuals [12].

Query answering under entailment When a SPARQL query is evaluated under OWL entailment, the query engine must follow a large number of `owl:sameAs` links in order to retrieve the full result set. With an identity service, a query engine can translate the terms in the query to identity set identifiers (see Section 3.3), calculate the SPARQL query using these identifiers, and translate the identifiers to terms in the result set.

Ontology alignment Existing algorithms for assessing whether or not two IRIs denote the same resource are currently evaluated on relatively small datasets [5]. The availability of a large dataset of real-world identity links can help quantify the utility of existing alignment algorithms such as [4].

3 Algorithms & Implementation

In this section we describe our approach for calculating and storing the identity relations $\sim_e$ and $\sim_i$ under the above requirements.

3.1 Explicit identity relation

The explicit identity relation ($\sim_e$) is obtained from the *LOD-a-lot* dataset[7] [7], a compressed data file that contains the unique triples from the 2015 LOD Laundromat corpus [2]. We use the HDT C++ library[8] to stream the result set of the following SPARQL query to a file, which takes ∼27 minutes:

```
select distinct ?s ?p ?o {
  bind (owl:sameAs ?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 558.9M triples that are the result of this SPARQL query 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 4.5GB in size, plus an additional 2.2GB for the index file that is automatically generated upon first use.

3.2 Explicit identity relation: Compaction

Since the implicit identity relation is closed under reflexivity and symmetry, the size of the input data can be significantly reduced prior to calculating the identity closure. We call this preparation step *compaction*. Assuming an alphabetic order $\leq$ on RDF terms, we can reduce the input for the closure algorithm to: $\{(x, y) \,|\, x \sim_e y \wedge x \leq y\}$. For this we use GNU sort unique, which takes 35 minutes on an SSD disk.

[7] `http://lod-a-lot.lod.labs.vu.nl`
[8] `https://github.com/rdfhdt/hdt-cpp`

The impact of the compaction step is significant: from ∼558.9M to ∼331M identity pairs, leaving out ∼2.8M reflexive and ∼225M duplicate symmetric pairs. As a result, the input size for the identity closure algorithm has been reduced by over 40%.

3.3 Implicit identity relation: Closure

Now that we have a compacted version of $\sim_e$, we calculate the identity closure $\sim_i$. As before, let N denote the set of RDF nodes. The implicit identity relation $\sim_i$ consists of a map from nodes to identity sets ($N \mapsto \mathcal{P}(N)$). For space efficiency, we store each identity set only once by associating a key with each identity set: $key : ID \mapsto_k \mathcal{P}(N)$; and map each RDF term to the key of the unique identity set that it belongs to $val : N \mapsto_v ID$.[9] The composition $key(val(x))$ gives us the identity set of x. We built an efficient implementation of this key-value scheme using the RocksDB persistent key-value store through a SWI Prolog API that we designed for this purpose[10].

When computing $\sim_i$ we successively derive new pairs (x, y). To store these efficiently we distinguish three cases:

Neither x nor y occurs in any identity set Then both x and y are assigned to the same new unique key for a new identity set: $x \mapsto_v id$, $y \mapsto_v id$, and $id \mapsto_k \{x, y\}$.

Only x already occurs in an identity set In this case, the existing (key for the) identity set of x is extended to contain y as well: $y \mapsto_v val(x)$, and $val(x) \mapsto_k key(val(x)) \cup \{y\}$. (The case of only y occurring in an identity set is analogous.)

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: $val(x) \mapsto_k key(val(x)) \cup key(val(y))$ and $y' \mapsto_v val(x))$ for every $y' \in key(val(y))$.

This is the most costly step, especially when both identity sets are large, but it is also relatively rare. Since the input pairs are sorted during the compacting stage, this case only occurs when there are pairs (a, x), (x, b) and (c, y) such that $a < x$, $c < x$ and $b < y$. A further speedup is obtained by choosing to merge the smaller of the two sets into the larger one. The merging of values is performed efficiently by RocksDB.

The calculation of the identity 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 $\mapsto_v$, and 6.6GB for $\mapsto_k$). RocksDB allows to simultaneously read from and write to the database. Since changes to the identity relation can be applied incrementally, the initial creation step only needs to be performed once.

[9] Note that each IRI in N does indeed belong to a unique identity set, because the identity sets of $\sim_i$ form a partitioning of N.

[10] See `https://rocksdb.org` and `https://github.com/JanWielemaker/rocksdb`.

3.4 Identity schema

In addition to the explicit and implicit identity relation, which use `owl:sameAs` to say something about other resources, we also extract the schema statements about `owl:sameAs` itself. This is obtained by storing the result of the following SPARQL Query in an HDT file.

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

4 Data analytics

In this section we perform several analyses over the dataset created with the algorithm described in Section 3.

4.1 Explicit identity relation

Terms in $\sim_e$: In the LOD Laundromat corpus, 179,739,567 unique terms occur in `owl:sameAs` assertions. As to be expected, the vast majority of these are IRIs (175,078,015 or 97.41%). Only a few literals are involved in the identity relation (3,583,673 or 1.99%), and even fewer blank nodes (1,077,847 or 0.60%). The majority of IRIs contain the HTTP(S) scheme (174,995,686 or 97.36.). Figure 1 gives an overview.

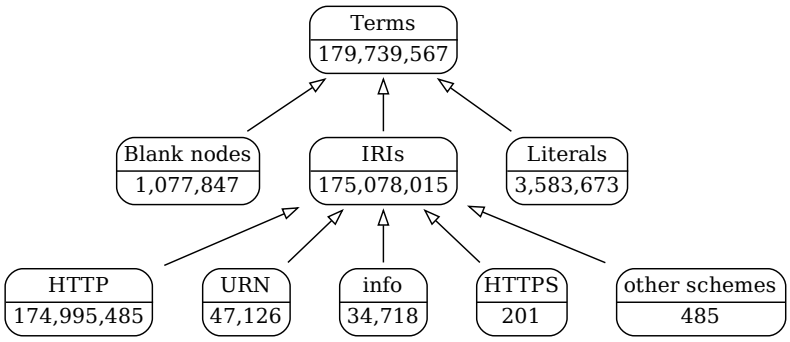

Fig. 1: Overview of the terms involved in the identity relation. Blank nodes, IRIs and literals do not sum to the number of terms exactly, because there are 32 terms that are neither (they are syntactically malformed IRIs).

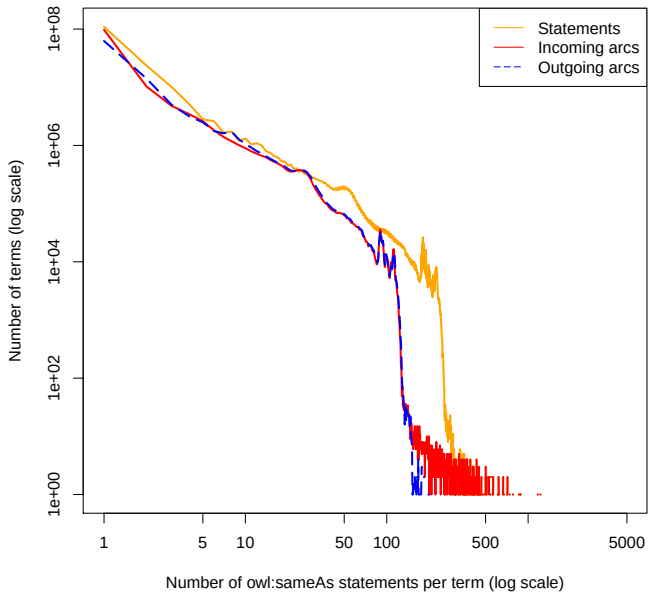

Fig. 2: The number of `owl:sameAs` statements per term.

Statements in $\sim_e$: The LOD Laundromat corpus contains a total of 558,943,116 `owl:sameAs` statements. Based on the 2011 Billion Triple Challenge dataset, the authors of [16] observed that the number of `owl:sameAs` statements per term approximated a power-law distribution with coefficient -2.528. In contrast to this, we find that in the 2015 LOD Laundromat corpus, although most terms do appear in a small number of statements, this distribution does not display a power-law distribution. The patterns for the distribution of **incoming arcs** (identity statements where the term appears in the object position) and the distribution of **outgoing arcs**, (identity statement where the term appears in the subject position) all follow a similar distribution pattern (Figure 2).

Dataset links in $\sim_e$: Because `owl:sameAs` is the most frequently used predicate to link between datasets [15], we also analysed $\sim_e$ at the aggregation level of links between datasets[11]. Unfortunately, there is no formal definition of what a dataset is. Since most of the terms involved in `owl:sameAs` assertions are HTTP(S) IRIs (Section 4.1), the notion of a *namespace* is a good proxy. According to the RDF 1.1 standard, IRIs belong to the same namespace if they have "a common substring". Obviously not every common substring counts as a namespace, otherwise all IRIs would be in the same namespace. A good pragmatic choice for a namespace-denoting substring is to take the prefix of HTTP(S) IRIs

[11] In this section, a *link* is an `owl:sameAs` statement between terms that belong to different datasets

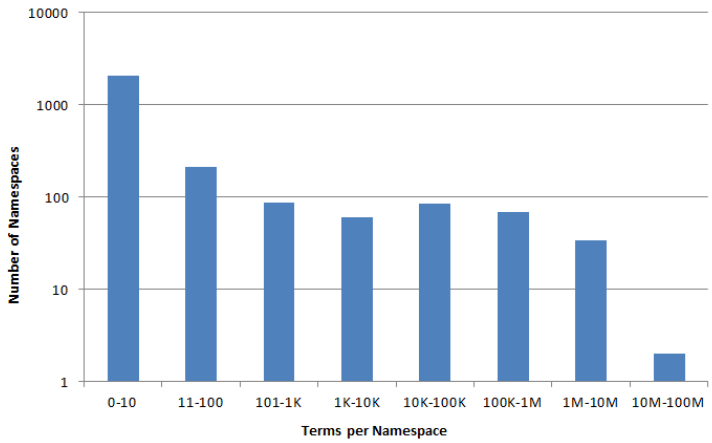

Fig. 3: The number of terms in identity links by namespace.

that ends with the *host name*. The host name is part of every syntactically valid HTTP(S) IRI, and denotes a physical machine that is located on the Internet.

Using this interpretation, Figure 3 shows that the number of terms occuring in `owl:sameAs` links is very unevenly distributed over namespaces (which we use as proxies of datasets). For each namespace we calculated the number of *incoming* and *outgoing* links (statements whose subject, respectively object, term is in a different namespace.) The remaining statements are *internal edges* (they either have two HTTP(S) IRIs that belong to the same namespace, or they have at least one node that is not an HTTP(S) IRI (i.e., either a blank node or a literal).

Figure 4 shows the distribution of internal edges, incoming links, and outgoing links over namespaces. While the majority of namespaces have incoming links, far fewer namespaces have outgoing links. This means that a relatively small number of namespaces is linking to a relatively large number of them. These namespaces are responsible for interlinking in the LOD Cloud. Finally, an even smaller number of namespaces have internal `owl:sameAs` edges. This means that most namespaces only use identity statements for linking to other datasets, but not for equating dataset-internal resources. This is a strong indication that most datasets enforce the Unique Name Assumption internally.

To give a high level impression, we have visualised the entire identity-graph at namespace level in Figure 5. This graph contains 2,618 host-based namespaces/datasets, that are connected through 10,791 edges, and consists of 142 components. The large black cluster at the bottom of the figure is the densely interconnected set of multilingual variants of `dbpedia.org`, with the two high centrality nodes for `dbpedia.org` and `freebase.com` clearly visible just above the black cluster.

The figure shows that there are high-centrality nodes that act as domain-specific naming authorities/hubs. For example, the central node in the large top cluster is `www.bibsonomy.org`, which links to a large number of bibliographic

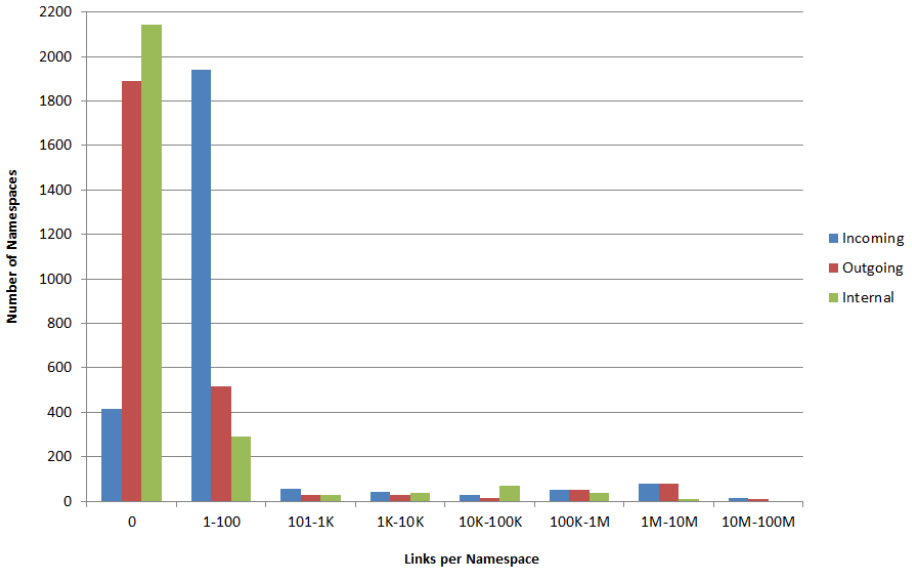

Fig. 4: The distribution of internal edges, incoming links, and outgoing links by namespaces.

datasets. A similar role is fulfilled by `geonames.org`, for interlinking geographic datasets; `bio2rdf.org`, for interlinking biochemistry datasets; and `revyu.com` (appearing at the right hand-side of the figure), for interlinking datasets that contain online reviews. A high-resolution version of this figure, together with textual namespace labels, is available at `https://sameas.cc/explicit/img`.

4.2 Implicit identity relation

Terms in $\sim_i$: The number of unique terms in $\sim_i$ is 179,672,306. This is less than the number of unique terms in $\sim_e$ (179,739,567), because 67,261 terms (or 0.037%) *only* appear in reflexive `owl:sameAs` assertions.

Identity sets in $\sim_i$ For the identity closure, it makes sense to separate out singleton identity sets, i.e., terms x for which $[x]_\sim = \{x\}$. A term has a singleton identity set if it never appears in a `owl:sameAs` assertion, or if all its `owl:sameAs` assertions are reflexive. We will not include singleton identity sets in our figures because they are conceptually trivial and their inclusion sometimes makes it hard to discern interesting aspects about the rest of $\sim_i$.

The number of non-singleton identity sets is 48,999,148. The LOD-a-lot file, from which we extract $\sim_e$, contains 5,093,948,017 unique terms. This means that there are 5,044,948,869 singleton identity sets in $\sim_i$. The distribution of identity set size (Figure 6) is very uneven and fits a power law with exponent 3.3 $\pm$0.04.

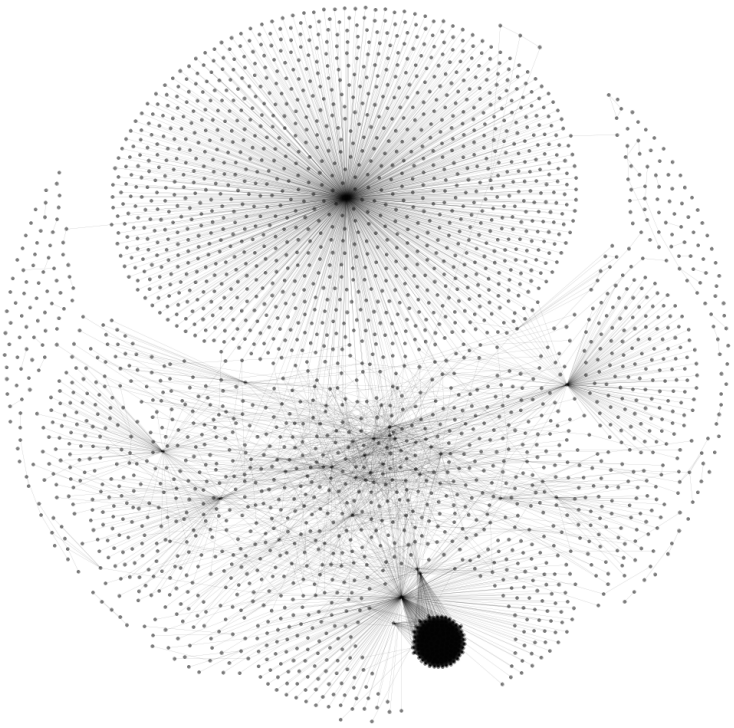

Fig. 5: All inter-dataset links in the LOD cloud. Thicker edges represent more identity links. The full diagram is available at `https://sameas.cc/explicit/img`.

The majority of non-singleton identity sets (31,337,556 sets; 63.96%) has size 2. There are relatively few large identity sets, and the largest identity set has cardinality 177,794. It includes Albert Einstein, the countries of the world, and the empty string.

Edges in $\sim_i$ We calculate the number of directed edges (or arcs) in the identity closure. This is the number of `owl:sameAs` triples that would be needed in order to express the full materialization of $\sim_e$. This calculation requires us to query and stream through the full RocksDB closure index, and therefore gives a good indication of the processing time required for running large-scale jobs over the sameAs.cc dataset. The calculation (i) retrieves all identity sets, (ii) calculates their cardinality, and (iii) sums the squares of the cardinalities. This operation takes only 55.6 seconds and shows that the materialization consists of 35,201,120,188 `owl:sameAs` statements. Notice that almost 90% (or 31,610,706,436 statements) of the materialization is contributed by the single largest identity set (i.e., $[\texttt{dbr:Albert\_Einstein}]_\sim$).

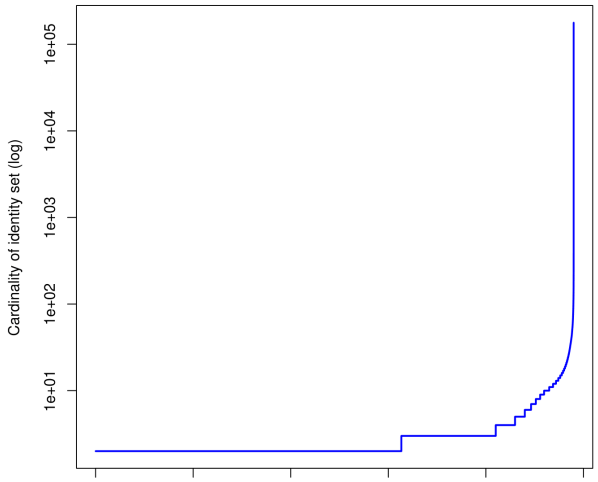

Fig. 6: The distribution of identity set cardinality in $\sim_i$. The x-axis lists all 48,999,148 non-singleton identity sets.

In addition to calculating the number of `owl:sameAs` statements in the identity closure, we can also calculate the minimal number of identity statements that would result in the same closure. We call such a minimal identity relation *a kernel*, and calculate it as the number of terms whose equivalence set is not a singleton set, minus the number of non-singleton identity sets. The kernel identity relation for $\sim_i$ consists of 130,673,158 statements (or 0.37% of $\sim_i$). This also means that 76.6% of the explicit identity statements ($\sim_e$) are redundant.

4.3 Schema assertions about identity

There are 2,773 assertions about `owl:sameAs` that extend the schema as defined in the OWL vocabulary in interesting ways. The dataset is available at `https://sameas.cc/schema`. We observe the following kinds of schema extensions:

Super-properties of `owl:sameAs` As [9] indicate, there is a need for properties that are weaker than `owl:sameAs` that express different shades of similarity and relatedness:

```
owl:sameAs rdfs:subPropertyOf <http://lexvo.org/ontology#nearlySameAs> .
```

Some super-property assertions introduce semantic bugs. For instance, since identity is the strongest equivalence relation, it does not make sense to assert new *identity* relations that are superproperties of it. The following statement introduces the semantic bug that everything is an individual:

```
owl:sameAs rdfs:subPropertyOf owl:sameIndividualAs .
```

Sub-properties of `owl:sameAs` Several datasets introduce sub-properties of `owl:sameAs`, i.e., strengthenings of the identity relation, without a clear use case. Our hypothesis is that these datasets intend to *weaken* the `owl:sameAs` property instead, since there are many use cases for weaker forms of similarity, relatedness, and context-dependent identity. For example:

```
bbc:sameAs rdfs:subPropertyOf owl:sameAs .
```

Other `rdfs:subPropertyOf` assertions cannot be easily corrected by swapping the subject and object term. For instance, from the fact that two things are the same link it does not follow that they are identical *sui generis* (since the same link may appear on different web pages):

```
<http://spitfire-project.eu/ontology/ns/sameAsLink> rdfs:subPropertyOf owl:sameAs .
```

Domain/range declarations As observed earlier by [10], the intersection-based semantics of `rdfs:domain` and `rdfs:range` is often not followed. The following classes are asserted as the domain of `owl:sameAs`, effectively stating that all resources are both legal entities, anniversaries, strings, etc.

```
owl:sameAs rdfs:domain <http://govwild.org/0.6/GWOntology.rdf#LegalEntity> .
owl:sameAs rdfs:domain <http://s.opencalais.com/1/type/em/e/Anniversary> .
owl:sameAs rdfs:range xsd:string .
```

Properties identical to `owl:sameAs` Several datasets mint alternative names for `owl:sameAs`. This is mainly used in combination with the introduction of sub- and super-properties of `owl:sameAs`, e.g.:

```
<http://rhm.cdepot.net/xml/#is> owl:sameAs owl:sameAs .
<http://sw.opencyc.org/concept/Mx4robv6phbFQdiM86Z2jmH52g> owl:sameAs owl:sameAs .
```

5 sameAs.cc: dataset & web service

The sameAs.cc dataset consists of the following components:

sameAs.cc The explicit identity relation ($\sim_e$) (`https://sameas.cc/explicit`) can be browsed online, queried for Triple Patterns, and downloaded as N-Triples and HDT.

sameAs.cc Closure We publish the implicit identity relation ($\sim_i$) (`https://sameas.cc`) as a downloadable snapshot of the RocksDB index (instead of a materialized RDF file). When RocksDB is installed, this snapshot can be queried locally.

sameAs.cc Schema The identity schema (`https://sameas.cc/schema`) can be browsed online, queried for Triple Patterns, and downloaded in N-Triples, and HDT.

The sameAs.cc web service[12] consists of the following components:

sameAs.cc Triple Pattern API The explicit identity relation web service (`https://sameas.cc/explicit/tp`) allows all `owl:sameAs` assertions to be queried with Triple Patterns. Queries are expressed through (combinations of) the HTTP query parameters `subject`, `predicate`, and `object`.

[12] Code is available at `https://github.com/wouterbeek/SameAs-Server`.

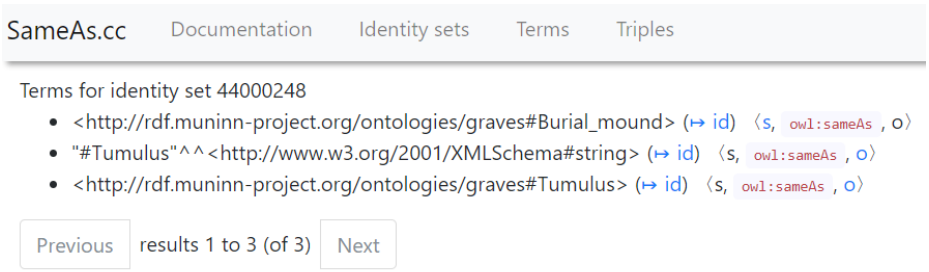

Fig. 7: Screenshot of the *sameAs.cc Closure API*. The screenshot shows the little known fact that tumulus is a synonym for burial mound.

sameAs.cc Closure API The implicit identity relation ($\sim_i$) can be queried through the following URI paths:

`https://sameas.cc/id` Enumerates all identity set IDs. Each member of the identity closure is assigned such a unique ID.

`https://sameas.cc/id?term=dbr:Albert_Einstein` Returns the ID of the identity set to which the given RDF term belongs. This view is shown in Figure 7.

`https://sameas.cc/term` Enumerates all RDF terms that appear in the identity relation.

`https://sameas.cc/term?id=44000248` Enumerates only the RDF terms that appear in the identity set with ID 44000248 as key.

We deliberately expose the internal key-value mechanism explained in Section 3.3 to the users of the sameAs.cc Closure API. The typical use case that we envision is one in which (i) terms are replaced by identity set identifiers, (ii) efficient computation is performed with the much more compact identifiers, and (iii) only when computation is done and end results need to be displayed are identifiers translated back to the potentially many terms that make up the respective identity sets.

6 Conclusion

In this paper we have presented *sameAs.cc*, the largest and most versatile dataset and web service of semantic identity links to date. The resource that we provide includes a large collection of `owl:sameAs` assertions and the closure calculated over it. The data can be freely downloaded and queried. Even though the datasets are large, our algorithms and data-structures ensure that the resources can be stored on and queried from a regular laptop.

In addition to the dataset and web services themselves, we have also presented several analytics over the data, including calculations of the size of the identity relation, its closure and its kernel, and various distributions. We hope that these resources will be used by other researchers in order to uncover aspects of identity that have not been studied before.

6.1 Acknowledgment

This work was partially conducted within the MaestroGraph project (612.001.553), funded by the Netherlands Organization for Scientific Research (NWO), and was partially supported by the Center for Data Science, funded by the IDEX Paris-Saclay, ANR-11-IDEX-0003-02.

References

1. S. Bechhofer, F. van Harmelen, J. Hendler, I. Horrocks, D.L. McGuinness, P.F. Patel-Schneider, and A.L. Stein. OWL Web Ontology Language Reference. Technical report, W3C, http://www.w3.org/TR/owl-ref/, February 2004.
2. W. Beek, L. Rietveld, and S. Schlobach. LOD Laundromat (archival package 2016/06). `https://doi.org/10.17026/dans-znh-bcg3`, 2016.
3. P. Bouquet, H. Stoermer, and B. Bazzanella. An entity name system (ENS) for the semantic web. In *ESWC*, pages 258–272. Springer, 2008.
4. G. Correndo, A. Penta, N. Gibbins, and N. Shadbolt. Statistical Analysis of the owl:sameAs Network for Aligning Concepts in the Linking Open Data Cloud. In *DEXA*, pages 215–230. Springer, 2012.
5. Jomar da Silva, Fernanda Araujo Baiao, and Kate Revoredo. Alin results for oaei 2017. In *OM-2017: Proceedings of the Twelfth International Workshop on Ontology Matching*, page 114, 2017.
6. L. Ding, J. Shinavier, Zh Shangguan, and D.L. McGuinness. SameAs Networks and Beyond: Analyzing Deployment Status and Implications of owl:sameAs in Linked Data. In *ISWC*, pages 145–160. Springer, 2010.
7. J. Fernández, W. Beek, M.A. Martínez-Prieto, and M. Arias. LOD-a-lot - A Queryable Dump of the LOD Cloud. In *ISWC*, 2017.
8. Hugh Glaser, Afraz Jaffri, and Ian Millard. Managing co-reference on the semantic web. In *WWW2009 Workshop: Linked Data on the Web (LDOW2009)*, April 2009.
9. H. Halpin, P.. Hayes, J. McCusker, D. McGuinness, and H. Thompson. When owl:sameAs Isn't the Same: An Analysis of Identity in Linked Data. In *ISWC*, pages 305–320. Springer, 2010.
10. A. Hogan, A. Harth, A. Passant, S. Decker, and A. Polleres. Weaving the pedantic web. *Linked Data on the Web Workshop*, 2010.
11. I. Horrocks, P.F. Patel-Schneider, and F. van Harmelen. From SHIQ and RDF to OWL: the making of a Web Ontology Language. *Journal of Web Semantics*, 1(1):7–26, 2003.
12. A.K. Joshi, P. Jain, P. Hitzler, P. Yeh, K. Verma, A. Sheth, and M. Damova. Alignment-based querying of linked open data. In *Confederated International Conferences: CoopIS, DOA-SVI, and ODBASE*, pages 807–824. Springer, 2012.
13. E. James Whitehead Jr. Control choices and network effects in hypertext systems. In *HYPERTEXT '99*, pages 75–82, New York, NY, USA, 1999. ACM.
14. K. Schlegel, F. Stegmaier, S. Bayerl, M. Granitzer, and H. Kosch. Balloon Fusion: SPARQL rewriting based on unified co-reference information. In *30th Int. Conf. on Data Engineering Workshops*, pages 254–259, March 2014.
15. M. Schmachtenberg, Ch. Bizer, and H. Paulheim. Adoption of the linked data best practices in different topical domains. In *ISWC*, pages 245–260. Springer, 2014.
16. X. Wang, Th. Tiropanis, and H.C. Davis. Optimising linked data queries in the presence of co-reference. In *ESWC*, pages 442–456. Springer, 2014.