Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 159 (2019) 162–171

Procedia Computer Science

www.elsevier.com/locate/procedia

23rd International Conference on Knowledge-Based and Intelligent Information & Engineering Systems

Interlinking RDF-based datasets: A structure-based approach

Pierre-Henri Paris[a], Fayçal Hamdi[a], Samira Si-said Cherfi[a]

[a]Conservatoire National des Arts et Métiers, CEDRIC, 292 rue saint martin, Paris, France

Abstract

With an increase in the number of Linked Open Data datasets, insufficient interlinking quality can lead to a decrease in overall data quality. Therefore, it is necessary to keep the interlinking quality as high as possible. One of the main ways to link datasets is to use *owl:sameAs* links, i.e. to indicate that two things are the same. But with its strict semantics, there is a lot of misuse of *owl:sameAs* in the wild. Indeed, identity is often relative and depends on the context of use. We therefore propose an approach that enables considering the characteristics of involved datasets to interlink datasets thanks to *owl:sameAs* statements. The experimental results performed on real-world datasets show that the proposed approach is promising.

© 2019 The Authors. Published by Elsevier B.V.
This is an open access article under the CC BY-NC-ND license (https://creativecommons.org/licenses/by-nc-nd/4.0/)
Peer-review under responsibility of KES International.

Keywords: Linked Open Data; identity link discovery; SameAs; OWL

1. Introduction

Linked Data[1] (LD) datasets, also called Knowledge Bases (KBs), use ontologies backed by Description logics (DL) and *OWL* (see [9]) to define their schema. *owl:sameAs*[2] links are the most commonly used links to link datasets together and discover new knowledge. The role *owl:sameAs* from the OWL ontology language states that two things (two individuals, instances or resources) are the same and that all statements about one instance are also true for the other (the indiscernibility of identicals). In the Semantic Web field, retrieving all interchangeable knowledge between two things is still a major challenge. For example, suppose we have two KBs, each containing an instance 'Paris', the French capital. One can link these two instances thanks to the role *owl:sameAs*. Therefore, the knowledge about Paris in the first KB can be used with Paris in the second KB.

E-mail address: pierre-henri.paris@upmc.fr, faycal.hamdi@cnam.fr, samira.cherfi@cnam.fr ().

[1] http://lod-cloud.net/
[2] owl: http://www.w3.org/2002/07/owl#

1877-0509 © 2019 The Authors. Published by Elsevier B.V.
This is an open access article under the CC BY-NC-ND license (https://creativecommons.org/licenses/by-nc-nd/4.0/)
Peer-review under responsibility of KES International.

10.1016/j.procs.2019.09.171

There are many ways to find such linksets (see Definition 2) of pairwise identical things between datasets. The most obvious is by using OWL semantics itself, e.g. by using roles such as *owl:hasKey*, *owl:FunctionalProperty*, *owl:InverseFunctionalProperty*, etc. Another way to find such links is to use frameworks based on similarity measures between instances.

However, *owl:sameAs* has a strict semantics[3] (see Table 1) and is proved to be often misused in the wild (see [8]). A lot of *owl:sameAs* links are in fact inappropriate whether they are simply wrong, or because they are context-dependent. For example, the city of Paris is geographically the same as the department of Paris, but they are administratively different. So, in a geographic context, the statement holds, but in an administrative context it does not.

In this work we propose an approach to detect identity links between instances of two KBs. This approach takes into account the structure of each KB which consists of the use of explicit (ontology axioms) and implicit (statistics about properties) characteristics of properties.

The remainder of this work is structured as follows: in Sect. 2, we will see several facets of the identity problem. In Sect. 4, we detail our proposition to improve instance matching quality. The Sect. 5 is about implementation and results of our approach. Finally, in Sect. 6 we present the future works and our conclusions about our approach.

2. Related work

There are several ways to handle the identity problem. The first one is the historical way that has as a main goal to retrieve (create) a maximum of good links. A second way is to find wrong identity links among existing ones. By using either semantics or statistics, it is possible to find links that may be erroneous.

The term *instance matching* refers to the problem of finding equivalent resources. The goal is to produce links between a source dataset and a target dataset. For each couple of instances, a similarity score is produced and if the score is above some (user defined) threshold, the link is validated. Frameworks like Silk [19] or KnoFuss [14] allow creating links between datasets after a configuration step. Ferraram et al. [6] published a complete survey and more recently Achichi et al. [1] and Nentwig et al. [13] proposed complementary surveys. In Section 5, we compare our approach with four such instance matching systems competing during OAEI 2017. Khiat and Mackeprang [12] proposed I-Match that compute similarity between normalized strings thanks to NLP. Achichi et al. [2] proposed Legato, a multistage instance matching system that first create vectors from instances by using NLP techniques and then compute the correlation between vectors, and finally use a clustering algorithm to eliminate some false positives candidates. In [11], the authors proposed an instance matching system that is looping between link discovery and repairing, thus allowing reducing the number of wrong candidates. They used an external lexicon (like WordNet) to increase the matching capabilities.

Identity link assessment approaches consist of checking if an *existing* link is true or false. Methods from those approaches may also be used to find links. Guéret et al. [7] proposed to use classical network measures to assess existing links. De Melo [5] proposed to use the unique name assumption within datasets (i.e. an instance has one name within a dataset) to spot sets of instances linked by *owl:sameAs* where at least one link may be wrong. Next, a linear programming algorithm is used to check if the link is wrong. In Papaleo et al. [15], the authors proposed a logical approach that tries to detect logical conflicts by using semantics features like functional properties in small sub-graphs containing the two involved instances to assess. Hence, this approach strongly relies on semantics. Paulheim [16] proposed to use data mining methods. Links are represented in an embedded space, then an outlier detection algorithm is used to detect links that may be wrong. Valdestilhas et al. [18] use a combination of semantics and graph partitioning algorithms to detect erroneous transitive properties. Raad et al. [17] proposed to use community detection on identity links network to detect erroneous links. Also using network structure, Idrissou et al. [10] proposed to combine several network metrics to find wrong links across multiple datasets.

Our work aims at finding identity links by considering the involved datasets structures. In other words, we use explicit characteristics (from the ontology) like functional properties, disjoint class axiom, i.e. all available semantics that can help us in our task. We also use implicit properties, i.e. how properties are used in datasets. A property is explicitly

[3] https://www.w3.org/TR/owl2-profiles/

defined in the ontology by its domain, its range, the fact that it can have a literal value or an object value (i.e. another IRI) and is implicitly defined within a dataset by the way it is used (e.g. how many instances use a given property?). Those explicit and implicit features may be of great help to discover new identity links. Concerning the implicit features, we propose to combine similarities function in conjunction with weights according to the use of each property within datasets. For instances of a given concept, we use two types of weights. One representing the importance of a property and one the discriminating power of this property.

Our work is complementary to the ones we have just seen. To the best of our knowledge, our approach is the first that uses information of the dataset structure to such an extent.

3. Background and Notation

In this section, we give the preliminary background, and introduce the notation. For a more exhaustive background, we refer the reader to [4].

In all the remainder of this article, $\mathcal{KB}$, $\mathcal{KB}_i$ denote KBs, C denotes a concept (i.e. a class or an instance type) and R denotes a role (i.e. a property or predicate). $R(a, b)$ denotes that an instance a has the role R with the value b. $C(a)$ denotes that a is an instance of the concept C. $sameAs(a, b)$ denotes that the instance a is the *same as* [4] the instance b. We also note $\mathcal{S}(\mathcal{KB})$ (resp. $\mathcal{O}(\mathcal{KB})$ and $\mathcal{R}(\mathcal{KB})$) the set of subjects (resp. objects and roles) belonging to triples from $\mathcal{KB}$. Definition 1 corresponds to the *input* of our approach:

Definition 1. (Source and target knowledge bases) *Let $\mathcal{KB}_s$ and $\mathcal{KB}_t$ be two RDF KBs such that $\mathcal{KB}_s = \langle \mathcal{T}_1, \mathcal{A}_1 \rangle$ and $\mathcal{KB}_t = \langle \mathcal{T}_2, \mathcal{A}_2 \rangle$. $\mathcal{T}_1$ and $\mathcal{T}_2$ are T-Boxes (see [4]), and $\mathcal{A}_1$ and $\mathcal{A}_2$ are A-Boxes (see [4]) such that $\mathcal{T}_1 \sqsubseteq \mathcal{T}_2$ $\mathcal{KB}_s$ and $\mathcal{KB}_t$ are called respectively the* source *and the* target *KBs.*

Therefore, for this approach, we require that the T-Box of the source KB be included in the T-Box of the target KB. We need for instances in both KBs to share the same roles (i.e. properties) and have the same concepts (i.e. classes). Indeed, in this work we do not deal with the ontology alignment problem for now. Moreover, if there is a need for alignment, it is possible to use multiple tools that exist and that gives good results.

Definition 2 corresponds to the *output* of our approach:

Definition 2. (Linkset) *Let $L_{\mathcal{KB}_s,\mathcal{KB}_t}(R) = \{R(a, b) | a \in \mathcal{KB}_s \wedge b \in \mathcal{KB}_t\}$ be a linkset between $\mathcal{KB}_s$ and $\mathcal{KB}_t$ for the role R. $L_{\mathcal{KB}_s,\mathcal{KB}_t}(R)$ is thus the set of triples such that the subject comes from the source KB, the role is R and object is in the target KB.*

Example 1. *If $R = owl{:}sameAs$, $\{s{:}London, s{:}Paris, s{:}New\_York\} \subset \mathcal{S}(\mathcal{KB}_s)$ and $\{t{:}London, t{:}Dublin, t{:}Paris\} \subset \mathcal{S}(\mathcal{KB}_t)$ then $L_{\mathcal{KB}_s,\mathcal{KB}_t}(R) = \{\langle s{:}London, t{:}London \rangle, \langle s{:}Paris, t{:}Paris \rangle\}$.*

In the next section, we will detail our approach.

4. Approach

First, we will see an overview of the main algorithm in Sect. 4.1, and then a detailed description of the different phases of our approach in Sect. 4.2.

4.1. Approach summary

Our approach to find if an *owl:sameAs* link can be created between two instances x_1 and x_2 can be summarized by the following:

1. Find any semantic proof of identity. If there is one, stop there.
2. Otherwise, for each common role R between x_1 and x_2 (such that $R(x_1, o_1)$ and $R(x_2, o_2)$):

[4] https://www.w3.org/TR/owl-ref/#sameAs-def

Table 1. Examples of used semantics for equality

	If	**Then**
eq-trans	$T(?x, owl{:}sameAs, ?y)$ $T(?y, owl{:}sameAs, ?z)$	$T(?x, owl{:}sameAs, ?z)$
prp-fp	$T(?p, rdf{:}type,$ $owl{:}FunctionalProperty)$ $T(?x, ?p, ?y_1)$ $T(?x, ?p, ?y_2)$	$T(?y_1, owl{:}sameAs, ?y_2)$
prp-ifp	$T(?p, rdf{:}type,$ $owl{:}InverseFunctionalProperty)$ $T(?x_1, ?p, ?y)$ $T(?x_2, ?p, ?y)$	$T(?x_1, owl{:}sameAs, ?x_2)$
cls-maxc2	$T(?x, owl{:}maxCardinality,$ “1”^^$xsd{:}nonNegativeInteger)$ $T(?x, owl{:}onProperty, ?p)$ $T(?u, rdf{:}type, ?x)$ $T(?u, ?p, ?y1)$ $T(?u, ?p, ?y2)$	$T(?y_1, owl{:}sameAs, ?y_2)$

(a) Compute classical similarity between objects of current common role (e.g. between o_1 and o_2) (see Algorithm 2)
(b) Compute the *weight of the role* R (see Definition 5)
(c) Compute the *discriminating power* of the role-object pair $\langle R, o_1\rangle$ (see Definition 7)
(d) Aggregate in $SubAggregation$ the similarity, the *weight of the role* R and the *discriminating power* of the role-object pair $\langle R, o_1\rangle$

3. Compute the *weight of evidence* between x_1 and x_2 in *evidence* (see Definition 8)
4. Aggregate all $SubAggregation$ variables and the *evidence* weight

In the following section, we will see more details about each step.

4.2. In-depth approach

There are two distinct phases when two instances are compared to know if they are the same and if an *owl:sameAs* link must be created between them. The first one relies on *owl:sameAs* semantics. The second one is the heart of the approach.

4.2.1. Direct semantic proof

The first step of our approach is to look for direct semantic proof of equality (or inequality) between the two inspected resources. Table 1 shows some examples of the semantics of *owl:sameAs* that have been used in our approach. We basically search for any pattern corresponding to one of the cases contained in Table 1. Basically, if the conditions in the *If* column holds (are true), then the column *Then* can be applied, i.e. we can explicitly add the triples it contained in the KB.

Example 2. *(Rule prp-fp from Table 1) Let us suppose we are assessing if the instances x_1 and x_2 are the same. If we find that the role P is a functional property [5] ($FunctionalProperty(P)$) and $P(s, x_1)$ and $P(s, x_2)$ (either directly in $\mathcal{KB}_1$ and $\mathcal{KB}_2$ or by inference), then we know for sure that $sameAs(x_1, x_2)$ holds. In that case the main similarity algorithm stops and return 1.*

[5] https://www.w3.org/TR/owl-ref/#FunctionalProperty-def

If there is a proof that x_1 and x_2 are different, the main similarity algorithm stops and return 0. If no semantic clue has been found, the main similarity algorithm continues to the next step.

Algorithm 1 illustrate our approach.

```
input : $x_1 \in \mathcal{KB}_s$ and $x_2 \in \mathcal{KB}_t$ ($x_1$ and $x_2$ are instances)
output: the similarity score
1  if $IsSemProof(x_1, x_2)$ then return $SemProofValue(x_1, x_2)$;
2  $scores \leftarrow []$;
3  $C \leftarrow max_C\{depth_{\mathcal{KB}}(C) \mid C(x_1) \in \mathcal{KB}_s \wedge C(x_2) \in \mathcal{KB}_t\}$;
4  /* At worst, $C = owl{:}Thing$                                        */
5  foreach $R \in \mathcal{R}_{x_1} \cap \mathcal{R}_{x_2}$ do
6      /* see Algorithm 2 for the sim function                           */
7      $(maxSim, o) \leftarrow max\{sim(o_1, o_2) \mid (o_1, o_2) \in \{o : R(x_1, o)\} \times \{o : R(x_2, o)\}\}$;
8      // the max() function returns a tuple composed of $o_1$ or $o_2$ depending on which
           one has the highest similarity score, and this score
9      $subscore \leftarrow Aggregation_1(maxSim, (1 - W_{\mathcal{KB}_s}(R, C)), (1 - D_{\mathcal{KB}_s}(C, R, o)))$;
10     $scores.Add(subscore)$;
11 end
12 /* The more information (triples) there is, the stronger the decision must be. w
       represents this force.                                            */
13 $evidence \leftarrow \frac{|R_{x_1} \cap R_{x_2}|}{|R_{x_1}| + |R_{x_2}| - |R_{x_1} \cap R_{x_2}|}$;
14 return $Aggregation_2(evidence, SubAggregation(scores))$;
```

Algorithm 1: Calculate the similarity score between two instances

```
input : $o_1 \in \mathcal{KB}_s$ and $o_2 \in \mathcal{KB}_t$
output: the similarity score between two objects
$score \leftarrow 0$;
if $termType(o_1) \neq termType(o_2)$ then return $score$;
// The termType() function returns a value between resource or literal, whether
    the parameter is an RDF resource or a literal value. Blank nodes are ignored
    for simplicity
if $termType(o_1) = resource$ then
    if semantic proof that $o_1 = o_2$ then
        $score \leftarrow 1$;
    else
        $score \leftarrow stringSim(fragment(o_1), fragment(o_2))$;
else if $dataType(o_1) = dataType(o_2)$ then
    // The dataType() function returns the data type of a literal, e.g. string or
        date.
    $score \leftarrow sim_{dataType(o_1)}(o_1, o_2)$;
else
    $score \leftarrow stringSim(o_1, o_2)$;
return $score$;
```

Algorithm 2: the sim function

The $IsSemProof$ function (line 1 in Algorithm 1) returns a boolean. If a semantic proof of equality or inequality is found, it return true. In the same line, $SemProofValue$ return either 1 or 0 (one if the evidence is in favor of $owl{:}sameAs$, zero otherwise).

4.2.2. The use of properties

In this second step of our approach, there are two main ideas. The first one is that rarely occurring role (among instances of a given concept) may be stronger to evaluate identity between two instances than an omnipresent role (see Example 3). The second one is that a role-object couple that occurred less is more helpful to find an identity link (see Example 4). We have been inspired by the fact that when looking for evidence we seek for a specificity, since peculiarities narrow down the possibilities.

Example 3. *If 90% of the* People*'s instances use the role* name *but only 8% of those instances use the role* ownerOf*, then* ownerOf *might help more to determine (the absence of) an identity relation between two instances.*

Example 4. *Let's say we have the following triples: $town(a, t_1)$ and $town(b, t_2)$. The discriminating power of the values is important. Suppose we have 100 instances with the role-object $\langle town, t_1\rangle$ but only 3 instances with the role-object $\langle town, t_2\rangle$, then evidence having the role-object $\langle town, t_2\rangle$ are stronger than evidence having $\langle town, t_1\rangle$, since $\langle town, t_2\rangle$ enable to discriminate more instances. We name this the discriminating power of a role-object pair.*

Hence, for each common role between the two instances x_1 and x_2, a classical similarity score is computed between objects. For example, if we have $country(x_1, o_1)$ and $country(x_2, o_2)$, we compute the similarity $sim(o_1, o_2)(\in [0, 1])$. This similarity depends on the nature of o_1 and o_2 (IRIs, typed literals, etc.). We will see more details later. Each of those similarities will be weighted based on intuitions explain in Example 3 and 4. More formally, before defining the *weight of a role*, we need two preliminary definitions:

Definition 3. *Let $NS_{\mathcal{KB}}(C) = |\{s : \exists(R, o) \in \mathcal{R}(\mathcal{KB}) \times \mathcal{O}(\mathcal{KB}), R(s, o) \in \mathcal{KB} \wedge C(s) \in \mathcal{KB}\}|$ the number of subjects from $\mathcal{KB}$ that are of the concept C.*

Example 5. *In following examples, we will use this $\mathcal{KB} = \{Woman(ada), Man(lennon), Woman(kahlo),$
$Man(obama), Man(einstein), age(lennon, 26), age(obama, 47), age(einstein, 26), age(kahlo, 37),$
$nationality(ada, brithish)\}$.*

*If $C = Woman$, then $NS_{\mathcal{KB}}(Woman) = 2$ (*kahlo *and* ada*).*

Definition 4. *Let $NS_{\mathcal{KB}}(C, R) = |\{s : \exists o \in \mathcal{O}(\mathcal{KB}), R(s, o) \in \mathcal{KB} \wedge C(s) \in \mathcal{KB}\}|$ be the number of subjects of C participating in triples having the role R in $\mathcal{KB}$.*

Example 6. *With KB from Example 5, $C = Woman$ and $R = Age$, then $NS_{\mathcal{KB}}(Woman, Age) = 1$ (only* kahlo *has an age provided).*

Now, we can define the *weight of a role* (as explain in Example 3):

Definition 5. (weight of a role) *The* weight of the role *R on the concept C in $\mathcal{KB}$ is defined by $W_{\mathcal{KB}}(R, C) = \frac{NS_{\mathcal{KB}}(C,R)}{NS_{\mathcal{KB}}(C)}$ and $W_{\mathcal{KB}}(R, C) \in [0, 1]$.*

This weight $W_{\mathcal{KB}}(R, C)$ represents the percentage of instances of a class (or concept) C having the role R in their description.

Example 7. *With KB from Example 5, then $W_{\mathcal{KB}}(Age, Woman) = \frac{NS_{\mathcal{KB}}(Woman,Age)}{NS_{\mathcal{KB}}(Woman)} = \frac{1}{2} = 50\%$.*

We will next define the second intuition seen in Example 4, that is the *discriminating power* of role-object pair:

Definition 6. *Let $NS_{\mathcal{KB}}(C, R, o) = |\{s : \exists s, R(s, o) \in \mathcal{KB} \wedge C(s) \in \mathcal{KB}\}|$ be the number of subjects of concept C participating in triples having the role R and the object o in $\mathcal{KB}$.*

Example 8. *With KB from Example 5, $C = Man$, $R = Age$ and $o = 26$, then $NS_{\mathcal{KB}}(Man, Age, 26) = 2$ (*einstein *and* lennon*).*

Now, we can define the *discriminating power*:

Definition 7. (discriminating power) *The* discriminating power *of a role-value pair $\langle R, o\rangle$ on the concept C in $\mathcal{KB}$ is defined by $D_{\mathcal{KB}}(C, R, o) = \frac{NS_{\mathcal{KB}}(C,R,o)}{NS_{\mathcal{KB}}(C,R)}$ and $D_{\mathcal{KB}}(C, R, o) \in [0, 1]$.*

The lower the number $D_{\mathcal{KB}}(C, R, o)$, the more the role-object $\langle R, o\rangle$ makes it possible to differentiate between two instances.

Example 9. *With KB from Example 5, then* $D_{\mathcal{KB}}(Man, Age, 26) = \frac{NS_{\mathcal{KB}}(Man,Age,26)}{NS_{\mathcal{KB}}(Man,Age)} = \frac{|\{lennon,einstein\}|}{|\{lennon,einstein,obama\}|} = \frac{2}{3} = 66\%$ *and* $D_{\mathcal{KB}}(Man, Age, 47) = \frac{NS_{\mathcal{KB}}(Man,Age,47)}{NS_{\mathcal{KB}} Man,Age} = \frac{|\{obama\}|}{|\{lennon,einstein,obama\}|} = \frac{1}{3} = 33\%$*. In combination with the role Age and for Man instances, the object "47" selects only one instance (*obama*) with its discriminating power of 33%, against two (*lennon, einstein*) for the object "26" with a discriminating power of 66%.*

To sum up, for each common role between two instances, we compute a similarity score weighted by the *weight of the role* and the *discriminating power* of the role-object pair.

Finally, we need to aggregate those weighted similarity scores. In the aggregation process we take into account the fact that the more evidence there is, the more trustworthy the result will be (see line 13 in Algorithm 1).

Example 10. *Let's say that we have three instances a, b and c where a is from* $\mathcal{KB}_s$ *and b and c are from* $\mathcal{KB}_t$*. On the one hand, if we have four evidence between a and b, and on the other hand, eight evidence between a and c then we strengthen the second result. We give a bonus to the comparison with the more evidence to present.*

Therefore, after aggregating all weighted similarity scores, we use a last weight to either penalize or reward the result according to the number of common roles between the compared instances. To do this, we compute the following weight:

Definition 8. (weight of evidence) $evidence = \frac{|R_{x_1} \cap R_{x_2}|}{|R_{x_1}|+|R_{x_2}|-|R_{x_1} \cap R_{x_2}|}$*, where* $evidence \in [0, 1]$.

Example 11. *If* $R_{x_1} = \{rdfs{:}label, foaf{:}name, dbo{:}birthPlace, dbo{:}birthDate\}$*,* $R_{x_2} = \{rdfs{:}label, dbo{:}birthDate, dbo{:}deathDate\}$ *and* $R_{x_3} = \{rdfs{:}label, foaf{:}name, dbo{:}birthPlace, dbo{:}deathDate\}$*, then* $R_{x_1} \cap R_{x_2} = \{rdfs{:}label, dbo{:}birthDate\}$*,* $evidence_{x_1,x_2} = \frac{2}{4+3-2} = \frac{2}{5} = 0.4$ *and* $R_{x_1} \cap R_{x_3} = \{rdfs{:}label, foaf{:}name, dbo{:}birthPlace\}$ *and* $evidence_{x_1,x_3} = \frac{3}{4+4-3} = \frac{3}{5} = 0.6$*. We can see in the second case that there is more information to support the approach.* x_1 *and* x_3 *have more information to be compared than* x_1 *and* x_2*.*

Some additional points need explanation. Line 1 from Algorithm 1 corresponds to the stage where we search for direct semantic features. Furthermore, in line 3, we use the $depth_{\mathcal{KB}}(C)$ function that is defined by the following:

Definition 9. (depth of a concept) *Let* $depth_{\mathcal{KB}}(C)$ *be the distance between the concept C and the concept T (i.e. owl:Thing in RDF) in* $\mathcal{KB}$*.* $depth_{\mathcal{KB}}(C)$ *is called the depth of C.*

By definition, $\forall \mathcal{KB}, depth_{\mathcal{KB}}(T) = 0$. (The *Top* concept is equivalent to *owl:Thing* in RDF, see [3] for more details.)

Example 12. *If* $\mathcal{KB} = dbo$[6] *then* $depth_{dbo}(Agent) = 1$ *and* $depth_{dbo}(Biologist) = 4$ *since Agent is a direct sub concept of owl:Thing and* $Biologist \sqsubseteq Scientist \sqsubseteq Person \sqsubseteq Agent \sqsubseteq owl{:}Thing$*.*

We retrieve the deepest common concept between x_1 and x_2, i.e. the most specific. It is this concept that will be used to compute the *weight of the roles* and the *discriminating power* of the role-object pairs. The idea is that if, for example, two instances are scientists, we will obtain better results if we use the *Scientist* concept than the *Human* concept.

Finally, Algorithm 2 shows how our approach handles similarity between two objects. The *fragment* function gives the last part of an IRI, i.e. after the last / or #, to compare IRIs in last resort. We compare data types and if they match we use an appropriate similarity measure (line 11), e.g. if they are both dates, then we use a similarity function working with dates. If both are resources, we check for direct semantic proof (as we have seen it before) and if we do not find anything we trivially compare IRIs. There has been no attempt to use recursivity in this part yet.

Furthermore, in the main algorithm (Algo. 1), there are three different aggregation functions (lines 9 and 14) that are used. We used the mean for all three.

[6] DBpedia ontology : http://wiki.dbpedia.org/services-resources/ontology

5. Experiments

In this section, to evaluate our approach, we will present two experiments performed on DBpedia/Wikidata and the SPIMBENCH SANDBOX track from OAEI 2017 and then discuss their results.

5.1. Results

We have developed a prototype in C# that can be run either on Linux or Windows machines. It is also available as a Docker container. We choose the open source library dotNetRDF[7] to handle SPARQL, OWL and RDF parts. All experiments are performed on a computer with an I7 processor (3.10 GHz) and 16 Go of RAM. All our code and datasets can be found on this Github repository so our experiments are reproducible: `https://github.com/PHParis/im_prototype`.

For each experiment, we have calculated the precision, recall and F-measure with result linkset as follows:

- True positive (tp) : number of alignments predicted (by our approach) that are actually true
- False positive (fp) : number of alignments predicted and actually wrong
- False Negative (fn) : number of true alignment not found among those predicted by our approach
- $Precision = \frac{tp}{tp+fp}$
- $Recall = \frac{tp}{tp+fn}$
- F-measure = $\frac{2 \times Precision \times Recall}{Precision + Recall}$

5.1.1. First experiment

For the first experiment, we performed an instance matching task on real-world and well-known datasets. Thus, we used subsets of DBpedia and Wikidata. More precisely, we used *DBpedia Wikidata*[8] in place of Wikidata since it is expressed with DBpedia ontology. Hence, we respect the following condition from Definition 1: $\mathcal{T}_1 \sqsubseteq \mathcal{T}_2$ (source TBox must be a subset of target TBox). The version of DBpedia used is “2016-10” and the version of *DBpedia Wikidata* is “03.30.2015”.

Now we describe how we built our two test KBs from real-world datasets. In the source dataset KB_s, we have selected instances of persons from DBpedia having at least 15 homonyms (according to *rdfs:label*) in Wikidata. We arbitrarily chose 15 homonyms to have a sufficient challenge without having to recover too many instances (the scaling of our approach is not our main concern at the moment). We queried all triples mentioning one of these instances to construct the source dataset. In the target dataset KB_t , we retrieved all homonyms (belonging to *DBpedia Wikidata*) of instances from our DBpedia selection. For example, the instance *dbpedia:John_Williams* has the label “John Williams” and there are 88 distinct instances in Wikidata with the label “John Williams”. The DBpedia selection contains all triples having *dbpedia:John_Williams* as subject or object and the *DBpedia Wikidata* contains all triples having one of the 88 instances as subject or object. From both source and target datasets, we obviously deleted *owl:sameAs* links after having assessed them (none of them were erroneous). There were 36 *owl:sameAs*. Those links were then used as a gold standard. The source KB contains 277 instances, 3468 triples and 36 candidate instances belonging to the concept to be matched with the target KB. The target KB contains 1170 instances, 7667 triples and 552 candidate instances belonging to the concept to be matched with the source KB.

After applying our approach (that took 12 seconds), we evaluated the generated linkset. We obtained a precision and a recall of 91.7%. Moreover, our approach produced 33 true positives, 3 false positives and 3 false negatives.

Therefore, our approach works well on real-world data. Next, we go further and compare with other approaches.

5.1.2. Second experiment

The goal of this experiment is two folds. First, we want to compare our approach with state of the art approaches. Secondly, other approaches we selected all use advanced terminology or structural comparison techniques (see Sect. 2)

[7] https://github.com/dotnetrdf/dotnetrdf
[8] http://wikidata.dbpedia.org/

to perform their task. Hence, we want to prove that a structure-based approach with simple string matching can perform as well as these approaches that use more advanced techniques. To compare our approach with others (see Sect. 2), we performed tests using the SPIMBENCH SANDBOX task from OAEI 2017[9]. This task has a source and a target dataset and a gold standard. A concept name is provided, so only the instances of this concept might be linked. SPIMBENCH SANDBOX datasets are alterations of an original one through value-based, structure-based, and semantics-aware transformations. The source KB contains 1432 instances, 10883 triples and 349 candidate instances belonging to the concept to be matched with the target KB. The target KB contains 1453 instances, 10868 triples and 443 candidate instances belonging to the concept to be matched with the source KB.

Table 2. Comparison with other approaches

Participants	Precision	Recall	F-Measure
SPIMBENCH Sandbox			
AML	0.849	**1.000**	0.918
I-Match	0.854	0.997	**0.920**
Legato	**0.980**	0.730	0.840
LogMap	0.938	0.763	0.841
Our approach	0.854	0.996	**0.920**

Table 2 shows a comparison of our results with each participant of the OAEI 2017 Instance Matching Track for the SPIMBENCH SANDBOX's task. Moreover, our approach produced 298 true positives, 51 false positives and one false negative.

5.2. Discussion

In the first experiment, results are good since we obtain 91.7% for all measures. Good links are well found, indicating that our approach is promising. Precision and recall are the same because, several times, a wrong candidate has been selected instead of the good one. This candidate selection issue is due to similarity scores that are too close between the good candidate and the (wrongly) selected one. It seems that our aggregation functions (see Section 4.2.2) are responsible for both true and false negative.

In the second experiment, for the SPIMBENCH SANDBOX dataset, our approach performed well since recall is 99.6% and precision 85.4%. We reached the same F-Measure as I-Match which was the best competitor on F-measure. In the same way as in the first experiment, several times, a wrong candidate has been selected instead of the correct one. The simple aggregation of the weights is responsible for most of the false positives. In addition, the use of more advanced similarity calculation techniques could improve candidate selection and thus reduce the number of false positives. For example, external KB or NLP techniques can improve comparison of strings.

One of the weaknesses of our approach we observed is the case where a wrong candidate is proposed with more corresponding property/value pairs of lesser importance than the right candidate has corresponding property/value pairs of importance. When instance descriptions are too close, our approach may not detect false positives.

There are several areas for improvement, like the three different aggregation functions as mentioned in the previous section. We use a simple arithmetic mean and we must investigate other ways to aggregate the sub-scores (i.e. scores for each common role from the loop line 5 in Algorithm 1) and the three weights (i.e. discriminating power and weight of a role, and the quantification of information). Also, we focus on the source KB for computing the discriminating power and weight of a role. It may be interesting to use both KBs in the process. Likewise, we use only one common concept between the two instances, although to use all common concepts may strengthen the score. Scalability should also be addressed, because if there are more than 1000 instances to match, our approach takes more than ten minutes to complete. This is mainly due to the absence of any code optimization for the moment. We also need to improve the

[9] http://islab.di.unimi.it/content/im_oaei/2017/

post-processing part concerning the validation of matches found. In fact, for now, we simply select for each source instance the best candidate in the target KB, but if the target does not contain an identical instance, then we produce a false positive. Finally, roles that are not shared by instances (we are trying to match) are discarded, but they may provide hints too. Unlike some other approaches, we do not use external resources as background knowledge. In addition, some approaches perform post-processing to eliminate false positives. We could benefit from this last two points, so our approach combining statistics on structure and semantics can be improved.

6. Conclusion

In this work, we have proposed a fully automatized approach to perform an instance matching task between two Knowledge Bases sharing their T-Boxes. This approach uses semantics at its disposal, but also uses statistics about roles and role-object pairs according to the most specific common concept between the compared instances.

The results show that our approach is a promising way towards better interlinking. The recall is good, which means that our approach works well to find links. Nevertheless, there are some improvements that can be made on the algorithm to better take into consideration the structure of the datasets, as discussed in the previous section. Especially to address our false positive detection that can clearly do better. Thus, a first step could be to test other ways to aggregate the different weights. In the future we may also investigate other ways to refine linkset we produced to have fewer false positives results. In addition, more parallelization can improve both scalability and speed performance.

References

[1] Achichi, M., Bellahsene, Z., Todorov, K., 2016. A survey on web data linking. Revue des Sciences et Technologies de l'Information-Série ISI: Ingénierie des Systèmes d'Information .

[2] Achichi, M., Bellahsene, Z., Todorov, K., 2017. Legato results for oaei 2017, in: OM@ISWC.

[3] Baader, F., Horrocks, I., Sattler, U., 2008. Description logics. Foundations of Artificial Intelligence 3, 135–179.

[4] Baader, F., Sattler, U., 2001. An overview of tableau algorithms for description logics. Studia Logica 69, 5–40.

[5] De Melo, G., 2013. Not quite the same: Identity constraints for the web of linked data., in: AAAI.

[6] Ferraram, A., Nikolov, A., Scharffe, F., 2013. Data linking for the semantic web. Semantic Web: Ontology and Knowledge Base Enabled Tools, Services, and Applications 169, 326.

[7] Guéret, C., Groth, P., Stadler, C., Lehmann, J., 2012. Assessing linked data mappings using network measures, in: Extended Semantic Web Conference, Springer. pp. 87–102.

[8] Halpin, H., Hayes, P.J., McCusker, J.P., McGuinness, D.L., Thompson, H.S., 2010. When owl: sameas isn't the same: An analysis of identity in linked data, in: International Semantic Web Conference, Springer. pp. 305–320.

[9] Horrocks, I., Kutz, O., Sattler, U., 2006. The even more irresistible sroiq. Kr 6, 57–67.

[10] Idrissou, A.K., van Harmelen, F., den Besselaar, P.V., 2018. Network metrics for assessing the quality of entity resolution between multiple datasets, in: EKAW.

[11] Jiménez-Ruiz, E., Grau, B.C., 2011. Logmap: Logic-based and scalable ontology matching, in: International Semantic Web Conference.

[12] Khiat, A., Mackeprang, M., 2017. I-match and ontoidea results for oaei 2017, in: OM@ISWC.

[13] Nentwig, M., Hartung, M., Ngonga Ngomo, A.C., Rahm, E., 2017. A survey of current link discovery frameworks. Semantic Web 8, 419–436.

[14] Nikolov, A., Uren, V., Motta, E., De Roeck, A., 2008. Integration of semantically annotated data by the knofuss architecture, in: International Conference on Knowledge Engineering and Knowledge Management, Springer. pp. 265–274.

[15] Papaleo, L., Pernelle, N., Saïs, F., Dumont, C., 2014. Logical detection of invalid sameas statements in rdf data, in: International Conference on Knowledge Engineering and Knowledge Management, Springer. pp. 373–384.

[16] Paulheim, H., 2014. Identifying wrong links between datasets by multi-dimensional outlier detection., in: WoDOOM, pp. 27–38.

[17] Raad, J., Beek, W., van Harmelen, F., Pernelle, N., Saïs, F., 2018. Detecting erroneous identity links on the web using network metrics, in: International Semantic Web Conference.

[18] Valdestilhas, A., Soru, T., Ngomo, A.C.N., 2017. Cedal: time-efficient detection of erroneous links in large-scale link repositories, in: Proceedings of the International Conference on Web Intelligence, ACM. pp. 106–113.

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