Logical Detection of Invalid SameAs Statements in RDF Data

Laura Papaleo[1], Nathalie Pernelle[1], Fatiha Saïs[1], and Cyril Dumont[1]

Université de Paris-Sud, Laboratoire de Recherche en Informatique,
Bâtiment 650, F-91405 Orsay Cedex, France
firstname.lastname@lri.fr
http://www.lri.fr

Abstract. In the last years, thanks to the standardization of Semantic Web technologies, we are experiencing an unprecedented production of data, published online as *Linked Data*. In this context, when a typed link is instantiated between two different resources referring to the same real world entity, the usage of *owl:sameAs* is generally predominant. However, recent research discussions have shown issues in the use of *owl:sameAs*. Problems arise both in cases in which sameAs is automatically discovered by a data linking tool erroneously, or when users declare it but meaning something less 'strict' than the semantics defined by OWL. In this work, we discuss further this issue and we present a method for logically detect invalid sameAs statements under specific circumstances. We report our experimental results, performed on OAEI datasets, to prove that the approach is promising.

Keywords: RDF identity link, sameAs, Linking quality & validation.

1 Introduction

The Semantic Web is a *'Web of Data'*, where data can be processed by machines, extending the principles of the Web from *documents* to *data* [1]. In this context, resources can be accessed using the conventional Web architecture (URIs) and it is possible to link resources using named relations. Today, we are experiencing an unprecedented production of resources, published as *Linked Open Data* (LOD, for short). This is leading to the creation of a global data space containing billions of assertions [2]. RDF [13] provides formal ways to build these assertions. Working in the LOD is basically about using the Web to create *typed links* (in RDF) between data from different sources. Most of the RDF links connecting resources coming from different data sources are *RDF identity links*, called also *sameAs statements*. They are defined using the *owl:sameAs* property, thus expressing that two URI references actually refer to the same thing. Unfortunately, many existing identity links do not reflect such genuine identity, as argued recently within the research community [5,4]. So, as numerous independently developed data sources have been published over internet as Linked Data, the *problem of identity* is now casting a shadow over the shininess of the Semantic Web [11,5].

K. Janowicz et al. (Eds.): EKAW 2014, LNAI 8876, pp. 373–384, 2014.

It is becoming extremely important to develop means of *data and linking quality assurance*. The study of the quality of data and links in the LOD cloud may be particularly useful in applications that want to consume Linked Data as well as in Semantic Web frameworks dedicated to data linking or data integration.

In this work, we investigate and design a logical method to detect invalid sameAs statements, by looking at the descriptions associated to the instances involved. We suppose that, in case of multiple data sources, mappings between properties are provided. Our approach is local, in the sense that, we build a contextual graph 'around' each one of the two resources involved in the sameAs statement and we study the descriptions provided in these contextual graphs. The construction of the contextual graph is based on properties that have specific characteristics (functional, local completeness). We claim that, when logical conflicts are encountered, the initial RDF identity link is 'inconsistent', meaning that it requires further investigation (supervised or automatic). We tested the approach on sameAs statements provided by linking tools that have been applied on Ontology Alignment Evaluation Initiative (OAEI) datasets, showing that our research direction is promising.

The remainder of this paper is organized as follows. In Section 2 related works are described. In Section 3, we present the conceptual building blocks of our approach, while Sections 4 and 5 are dedicated to the formulation of the problem and the generation of the rules. In Sections 6 and 7 we present the logical method and the experimental results.Finally, in Section 8 some concluding remarks are drawn.

2 Related Works

How to evaluate and assess the quality of data and links in the Linked Data Cloud is a generally novel problem, growing its importance in the last years, as the research community can, now, work with a massive quantity of data coming from multiple data sources.

In [6] the authors present a 'global approach' where they analyzed the structural properties of large graphs of identity links focusing the attention on general network properties such as degree distributions and URI counts, without analyzing the quality. Recently, in [10] the authors describe another global approach in a framework dedicated to the assessment of Linked Data mappings using network metrics. Five different metrics have been performed on a set of known good and bad links concluding that most of these metrics are not meaningful with respect to the evaluation of the quality of an identity link. In [4], the author illustrates how to assess the quality of *owl:sameAs* links, using a constraint-based method. In the work, an interesting formalization of the problem in a graph-based fashion is presented, but the evaluation of the quality of the identity link is, in the end, performed using only one property, namely the name of each entity. The results are interesting, but as claimed but the author himself, it could be important to include advanced similarity measures and the evaluation of more properties. In [5,12] the authors studied the problem of the quality of RDF identity links from a general point of view, making observations about the varying use of *owl:sameAs* in Linked Data. They proposed an ontology called the Similarity Ontology (SO) that aims at better classifying the different level of similarity between items in different data sources. However, the quality evaluation of the *owl:sameAs* links is performed manually, assessing around 250 *owl:sameAs* links in an Amazon Mechanical Turk experiment.

In this paper we propose a method which analyzes more information than simply the resource name, as opposite to [4]. Our approach is 'local', differently from [6,10] as we assess the correctness of a sameAs statement by studying the information described in contextual sub-graphs built according to specific criteria. To the extent of our knowledge, there not exist similar logical methods in the literature.

To complete this Section, we need to recall that there exist a lot of interesting methods related to *owl:sameAs* link discovery (see [7] as recent survey). This is also referred as the 'coreference problem' in Semantic Web. The reader can, for example, see the works in [9,21,18] or, more recently those in [25,23,16]. However, *sameAs statement quality assessment* is, generally, different from the coreference problem. From the 'coreference prospective', the goal is to analyze the knowledge related to two resources in order to decide if one new assertion can be added to the knowledge base. In various domains, there are generally accepted naming schemata [2]. If two resources in the knowledge base both support one or more of these identification schema, the implicit relationship between entities can be made explicit as identity link, automatically. This can be true, for example, in case of a unique code such as the italian 'Codice Fiscale' that can be derived through a deterministic algorithm from a person's name and his/her date and place of birth, or the International Standard Book Number (ISBN) which is a unique numeric commercial book identifier. When no shared naming schema exist, RDF identity links are usually generated by evaluating the similarity of entities using more or less complex similarity functions. These functions generally take into account sub-parts of resource description that is known to be discriminative enough as, for example, inverse functional properties or composite keys. Few linking tools are interested in generating *owl:differentFrom* links, as for example [20]. This idea of partitioning the resources into groups of 'different resources' is used also in blocking methods as [22]. Then, data linking tools will search for sameAs links only within a group. However, in such approaches, only direct data-type properties are taken into account. Instead, once a sameAs statement exists in the knowledge base, it could be interesting to analyze different properties (not only inverse functional). To clarify this point, let us consider a very simple example: we have two resources (books) b_1 and b_2 both described using two data-type properties *isbn* and *pages*. We assume, for example, that the property *isbn* is inverse functional and *pages* is only functional. In order to infer $sameAs(b_1, b_2)$, it is sufficient to check if the values of *isbn* are equal. Using the semantics of *owl:sameAs* it is possible to infer that the values of the property pages are equivalent. If they are not, one can detect a conflicting case contingent on the semantics of $sameAs(b_1, b_2)$. In conclusion, it is sure that the two problems are entailed and, as immediate future activities, we are planning to deepen the analysis of their interconnection, especially in the case of complex and hybrid linking methods that are recently emerging.

3 Preliminaries

In this Section we present the theoretical framework in order to define the building blocks for the logical invalidation approach.

Definition 1. ***RDF Graph.*** *[17]*
An RDF graph is a set of RDF triples. The set of nodes of an RDF graph is the set of subjects and objects of triples in the graph.

Given an infinite set U of URIs, an infinite set B of blank nodes and an infinite set L of literals, a RDF triple is a triple $\langle s, p, o\rangle$ where the subject $s \in (U \cup B)$, the predicate $p \in U$ and the object $o \in (U \cup B \cup L)$. A *RDF triple* represents an assertion: if the triple $\langle s, p, o\rangle$ exists, the logical assertion $p(s, o)$ holds. An RDF graph G is simply a collection of RDF triples and it can be seen as a set of statements describing partially (or completely) a certain knowledge.

Definition 2. ***SameAs Statement.*** *[6]*
A SameAs statement $sameAs(s, o)$ is an RDF triple $\langle s, owl : sameAs, o\rangle$ in an RDF graph G which connects two RDF resources s and o by means of the owl:sameAs predicate.

Such an *owl:sameAs* statement indicates that two URI references refer to the same *thing* : the individuals have the same 'identity' [15]. Given an RDF graph G as defined above, the OWL2 RL rules define the owl:sameAs as being reflexive, symmetric, and transitive, and they axiomatize the standard replacement properties.

Definition 3. ***Property-based walk of length n*** $w_{\{n,s,P\}}$**.**
Given an RDF graph G, a node s in G, given a set P of properties defined for G, a Property-based walk of length n $w_{\{n,s,P\}}$ is an alternating sequence of nodes and predicates $\{v_0 \equiv s, p_0, v_1, p_1, v_2, \ldots, v_{n-1}, p_{n-1}, v_n\}$, such that

- *$v_0, \ldots, v_{n-1}$ are resources in G, $\forall i = 0, \ldots, n-1$ $v_i \in U$,*
- *v_n is a literal in G, $v_n \in L$*
- *each triple $\{v_i, p_i, v_{i+1}\}$ in the sequence is an RDF triple in G such as $p_i \in P$*
- *all the resources in the walk are distinct from one another. Thus, for each pair of resource $\{v_i, v_j\}$, v_i and v_j are not the same resource, with $\{i, j\} \in [0, \ldots, n-1]$ (they have different URIs).*

In the definition we suppose that each predicate in G has an associated weight 1 that expresses its existence (its length). $w_{\{n,s,P\}}$ is basically a path in the RDF graph without cycle and of length n, involving $n + 1$ node, n resources defined by URIs and 1 node as a literal. It can be seen also as a collection of assertions selected according to specific conditions (the starting resource s and the set of properties P). In other words, with a walk $w_{\{n,s,P\}}$ in the graph G, we select a sequence of assertions in some way related to the resource s. This means also that, for every RDF triple $\langle v_i, p_i, v_{i+1}\rangle$ in $w_{\{n,s,P\}}$ the fact $p_i(v_i, v_{i+1})$ holds.

Definition 4. ***m-degree Contextual Graph*** $G_{\{m,s,P\}}$
Given and RDF graph G and a node $s \in G, s \in U$, an integer number m and a set P of properties defined for G, a m-degree Contextual Graph $G_{\{m,s,P\}}$ for s is a sub-graph of G such that every node $v_i \in G_{\{m,s,P\}}$ belongs to a property-based walk of length n, with $n \leq m$.

A m-degree contextual graph for a resource s can be seen as a subset of knowledge pertinent to s, bounded by the set of predicates P. Given an RDF graph G, in which circles identify resources with URI and rectangles represent literals, Figure 1-(left) shows a walk $w_{\{2,s,P=\{P_0,P_1\}\}}$ for the resource s. The walk has length 2 and involves the properties P_0 and P_1. Figure 1-(right) shows a 2-degree contextual graph $G_{\{2,s,P=\{P_0,\ldots,P_4\}\}}$ for the same resource s. It involves the properties P_0, P_1, P_2, P_3 and P_4.

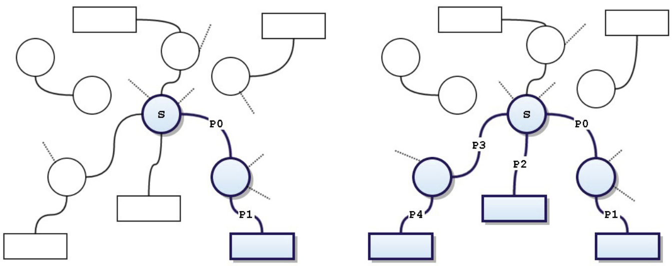

Fig. 1. (left) a walk of degree 2 for the resource s, $w_{\{2,s,P=\{P_0,P_1\}\}}$. (right) The contextual graph $G_{\{2,s,P=\{P_0,\ldots,P_4\}\}}$ of degree 2 for the same resource s.

4 Problem Statement

The problem we are addressing is to check if a sameAs statement can be invalidated and eventually explain this deduction. We need to check for inconsistencies in the assertion $sameAs(x, y)$ according to the knowledge provided in the RDF graph G.

Our approach relies on building two contextual graphs (see Section 3), for x and y respectively and on reasoning on the assertions contained in these two graphs. The building blocks of the problem are the following:

- An RDF graph G
- two resources x and y, such that x, y are resources in G
- the triple $\langle x, owl : sameAs, y\rangle$ (or $sameAs(x, y)$) belonging to G
- a set of properties P in G

- a value n representing the depth of the contextual graphs
- the contextual graphs $G_{\{n,x,P\}}$ and $G'_{\{n,y,P\}}$ for x and y

The problem becomes the evaluation of the following rule:

$$G_{\{n,x,P\}} \wedge G'_{\{n,y,P\}} \wedge sameAs(x,y) \Rightarrow \bot$$

The construction of the contextual graphs depends on the predicates (properties) we select and the value n. Indeed, in complex RDF graph, which can combine data coming from multiple data sources, limiting the depth of a contextual graph could be wise. The main reason is that long property-based walks can lead to not relevant piece of information which can eventually confuse the validation process. In Figure 2 we show an example of what we want to build. In this case, the statement $sameAs(x,y)$ must be validated, and a value $n = 2$ has been selected. The set of properties P has been defined as $\{P_0, \ldots, P_4\}$. The image shows the two contextual graphs extracted for x and y. In the following Section we explain how we want to choose the predicates.

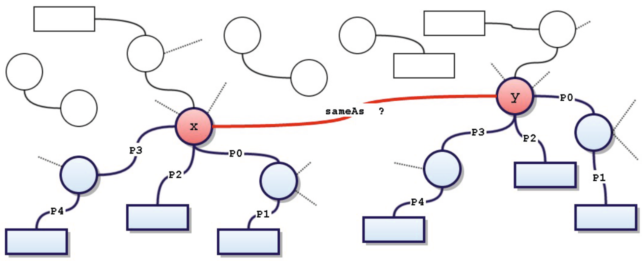

Fig. 2. The statement $sameAs(x,y)$ must be validated. The two 2-degree contextual graphs extracted for x and y are highlighted.

5 Properties Selection and Rules Generation

In this work, we chose to use (inverse) functional properties and those properties declared as local complete. Here, we explain and motivate this choice, describing the logical rules we add in the resolution system.

5.1 Functional and Inverse Functional Properties

Let us suppose that p_1 is a functional property. It can be expressed logically as follows [15]:

$$p_1(r,v) \wedge p_1(r,v^{'}) \Rightarrow v \equiv v^{'}$$

If we want to validate $sameAs(x,y)$ and we have a mapped functional property p_1, with $p_1(x,w)$ and $p_1(y,w_1)$, and we can assert in some way that $w \not\equiv w_1$ then:

$$sameAs(s,o) \wedge p_1(s,w) \wedge p_1(o,w_1) \wedge w \not\equiv w_1 \Rightarrow \bot$$

We have an inconsistency. A similar reasoning can be done for inverse functional properties. In these situations, if we assume that the assertions already in the RDF graph are true and we have 'doubts' only on the sameAs statement, we can conclude that this latter has problems. In our approach, taking into consideration functional properties, we basically add the following rules for every property p_i, p_j, p_k in the contextual graphs we are considering.

- $R1_{FDP}$: $sameAs(x, y) \wedge p_i(x, w_1) \wedge p_i(y, w_2) \rightarrow synVals(w_1, w_2)$
- $R2_{FOP}$: $sameAs(x, y) \wedge p_j(x, w_1) \wedge p_j(y, w_2) \rightarrow sameAs(w_1, w_2)$
- $R3_{IFP}$: $sameAs(x, y) \wedge p_k(w_1, x) \wedge p_k(w_2, y) \rightarrow sameAs(w_1, w_2)$

Note that $R1_{FDP}$ is for data-type properties and $R2_{FOP}$ and $R3_{IFP}$ are for object-type properties. $synVals$ and $\neg synVals$ are further described in Section 6. Given a property p in the graph G, the knowledge of p being a functional property can be already present among the assertions in G or derived after, collecting knowledge from experts or gathering it externally (existing ontologies, additional assertions on the Web and so on.)

5.2 Local Completeness

The closed-world assumption is in general inappropriate for the Semantic Web due to its size and rate of change [14]. But in some domains and specific contexts, local-completeness for RDF predicates (properties) could be assured. A good example for a multi-valued local complete property could be one representing the authors of a publication. When a predicate is like that, it should be declared *closed* in the specific knowledge base, making a local completeness assumption. A Local Completeness (LC) rule specifies that the resource is complete for a subset(s) of information (on a particular ontology): the information contained in the resource is all the information for the subset (specified by the rule) of the domain. In an RDF graph G, we declare the following OWL2 RL rule for each property that fulfills LC:

- $R4_{LC}$: $sameAs(x, y) \wedge p(x, w_1) \rightarrow p(y, w_1)$

where p is a predicate defined in the RDF graph G, x and y are object-type resources in G ($x, y \in U$) and w_1 is a literal ($w_1 \in L$). This rule will be used to discover inconsistencies since negative facts can be inferred because of the local completeness, as explained in the next Section. Given a property p, the knowledge of 'local completeness' for p can be asserted by an expert or discovered using semi-automatic approaches.

6 The Invalidation Approach

In this Section we present our invalidation approach, on the basis of all the definitions and reasoning made so far. Given G the initial RDF graph with U the set of resources in G with URIs. Given $sameAs(x, y)$ the input sameAs statement to validate, where $x, y \in U$. Let F be a set of facts, initially empty, and L the set of literals for G.

1. Build a set F_1 of $\neg synVals(w_1, w_2)$, for each pair of semantically different w_1 and w_2, with $w_1, w_2 \in L$.
2. Choose a value n indicating the depth of the contextual graphs
3. Build the contextual graphs for x and y considering (inverse) functional properties and local complete properties
   – For all the (inverse) functional properties $p_{i_{FP}}$ add the relative set of RDF facts to F, considering the rules $R1_{FDP}, R2_{FOP}, R3_{IFP}$ in Section 5.
   – For each $p_{i_{LC}}$ that falls in the contextual graphs and fulfills the local completeness (i.e. $R4_{LC}$ is declared), add to F a set of facts in the form $\neg p_{i_{LC}}(s, w)$ if w is different to all the w' s.t. $p_{i_{LC}}(s, w')$ belongs to F, using F_1. Note that $w, w' \in L$.
4. Apply iteratively unit resolution until saturation [19] using $F \cup CNF^1\{R1_{FDP}, R2_{FOP}, R3_{IFP}, R4_{LC}\}$.

Note also that disjointness of classes can be provided as input and considered in the resolution.

The set of $\neg synVals(w_1, w_2)$ with $w_1, w_2 \in L$ can be obtained using different strategies. It is possible, for example, to perform a pre-processing step in which we build a clustering of the values according to specific criteria. To clarify, consider a simple example of names of cities in a specific domain: it is possible to pre-process all the possible values and assert that $synVals('Paris', 'ParisCity')$ and that $\neg synVals('Paris', 'Milan')$ and so on. Thus, the evaluation is based on determining if two values w_1, w_2 belong to the same cluster. Another situation arises when the values are 'well defined' as in the case of enumeration, dates, years, geographical data or some types of measures. In these cases, the evaluation is again a simple syntactic comparison of the values. If they are the same, they are equivalent, otherwise they are not equivalent.

7 Experimental Results and Discussion

A prototype of our validation framework has been implemented in Java using the AIMA library for the resolution. In this Section we present the results of the experiments we performed for assessing the quality of the set of sameAs statements computed by different linking methods, respectively presented in [20], [24] and [26].

In [20] the sameAs statements are computed according to similarity measures over specific property descriptions, as in [26] where similarity between entities is iteratively calculated by analyzing specific features. In [24], instead sameAs statements are computed on the basis of a novel algorithm for key discovery. All the above methods have produced results on the Person-Restaurants (PR) data test available for the instance matching contest OAEI 2010 (IM@OAEI2010) [3]. For the key discovery method, we started from the links obtained by the method considering only the *name* of the restaurant as a key. According to the knowledge

[1] CNF: Conjunctive Normal Form.

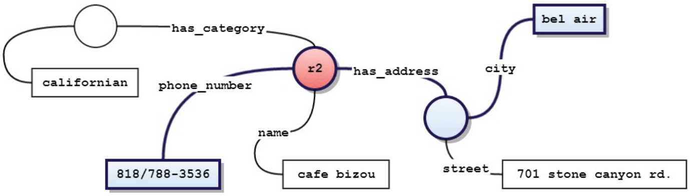

Fig. 3. An instance of restaurant in the dataset 'restaurant1' Given the functional properties *phone_number*, *has_address* and *city*, a contextual graph of degree 2 is depicted

base, we considered as 'meaningfully' functional the properties *phone_number* and *has_address* that describe a restaurant and *city* that describes an address[2]. Thus, given a sameAs statement in the form $sameAs(x, y)$ we computed the contextual graph of degree 2 considering the three functional properties listed before. Figure 3 shows an example of the contextual graph computed for a restaurant in the first dataset 'restaurant1' (already mapped). To build the $\neg synVals$ (set F_1) for the values of the properties selected, we did a normalization of the values. For *phone_number*, we removed all the additional characters (e.g. '/', '-', and so on), leaving only the numbers. We note that the same number of digits are given for all the phone numbers. For *city*, we removed words which can be not meaningful such as 'city', the character '(' and so on. A $\neg synVals$ is declared for each pair of syntactically different pairs of values. To explain the results obtained, let us consider the answers collected by applying our invalidation approach on the sameAs statements computed by [24]. Note that, in this case, the analysis is performed using properties completely different from those used in the computation of the identity links. Over the 90 sameAs statements computed, 4 were wrong with respect to the gold standard. We are able to detect 3 of these 4 erroneous links. The only one we cannot detect is the one linking restaurant 91 defined in dataset 1 to restaurant 711 defined in dataset 2. By looking at the properties, the two restaurants share the same phone number and the same city. They even share the same street name. So an inconsistency cannot be detected. Most probably, they represent the same commercial site providing different services (they have different categories). In addition we classify as 'wrong sameAs' 5 statements which, with respect to the gold standard, are in fact good sameAs statements. The reason is that, in every statement $sameAs(x, y)$, the restaurants x and y have different phone number or different city (or both). This type of result can be seen dually. On the one hand this could mean, for example, that a restaurant can have two phone numbers, so maybe the property *phone_number* is not functional. On the other hand, there can be errors in the data (for example 'los angeles' and 'los feliz') and the computation of the $\neg synVals$ has been imprecise. In any case, the good idea is to highlight the inconsistency to the

[2] Note that both the previous methods aligned the two initial datasets in order to compute the sameAs statements. We considered the same alignment in the explanation of the results.

user (expert) and ask for confirmation or correction. Table 1 shows a tabular summary of our tests, including accuracy, recall and precision of the method. The table indicates as: *(i)* **TC**: total cases to be considered, namely the number of *sameAs* found by the linking algorithm. *(ii)* **RG**: the number of the sameAs statements really wrong, wrt the gold standard. *(iii)* **TN**: (true negative), the number of statements which we detected 'good' and were actually correct (wrt the gold standard). *(iv)* **TP**: (true positive), the number of statements which we detected 'wrong' and were actually wrong (wrt the gold standard). *(v)* **FP** (false positive), the number of statements which we detected 'wrong' but were actually correct. *(vi)* **FN**: (false negative), the number of real wrong statements which we could not detect. Additionally, by definition, $accuracy = (TP + TN)/TC$, $recall = TP/(TP + TN)$ and $precision = TP/(TP + FP)$.

Table 1. Results of our approach on the sameAs links provided by the linking methods. We report the accuracy, recall and precision for the invalidation approach (IA) and the overall precision (LM+IA) in the last column.

Linking Method	LM precision	TC	RG	TN	TP	FN	FP	accuracy	recall	IA precision	LM+IA precision
[24]	95.55%	90	4	81	3	1	5	93,34%	75%	37%	98.85%
[20]	69.71%	142	43	94	38	5	5	92.9%	88.4%	88.4%	95.19%
[26]	90.17%	112	11	86	11	0	16	86.60%	100%	42.30%	100%

In conclusion, our results showed that, when our validation tool is applied after one of the linking tool, the precision of each tool can be improved, namely for [24] we pass from a precision of 95.55% to 98.85%, for [20] from a precision of 69.71% to 95.19% and finally for [26] from a precision of 90.17% to 100%.

8 Concluding Remarks

In the last years, the amount of data published on online as Linked Data is growing significantly. In this context, the usage of *owl:sameAs* is generally predominant when linking resources from different data sources. Recent research discussions within the Linked Data community have shown that the use of *owl:sameAs* may be incorrect. Hence, the needs of methods to assure and validate the quality of links in RDF stores.

In this paper we argued on the problem of evaluating sameAs statements.We designed a logical evaluation method which relies on the descriptions associated to the resources involved in the sameAs statement.Our method analyzes the functional properties and the properties defined as local complete. It builds a contextual graph for each resource and assesses the equality of each description involved. We formulated the necessary concepts and formally presented the approach, indicating the set of rules we use. We experimented the method with 3 datasets of sameAs statements produced by 3 different linking tools. The analysis of the results proved that, by applying our method after the linking, the precision is improved. We are working on completing the comparison with other methods, e.g.[8,16].

In the future, we are planning to explore different research directions. First, we are going to run experiments on more complex datasets. Second, we are working in the formalization of a set of rules for the re-qualification of a 'wrong' sameAs statement, in cases in which two resources represent the same conceptual element but at *different levels of details*. We want extend our approach using similarity measures on property values, allowing us to work with data with typos errors. As ultimate goal, we are aiming at designing an integration framework where knowledge base can be assessed, enhanced and visualized, using inferences on data and links, including data fusion, links corrections, and organization of the knowledge and the data at different levels of abstraction.

Acknowledgment. Work supported by the French National Research Agency: Quality and Interopera bility of Large Catalogues of Document project (QUALINCA-ANR-2012-CORD-012-02), by the Digiteo research cluster (digiteo.fr) within the framework of its OMTE programme and by the software group of the LRI lab.

References

1. Berners-Lee, T., Hendler, J., Lassila, O.: The semantic web. Scientific American 284(5), 34–43 (2001)
2. Bizer, C., Heath, T., Berners-Lee, T.: Linked data - the story so far. International Journal Semantic Web Information Systems 5(3), 1–22 (2009)
3. OEAI Campaign. Im@oaei2010 - persons-restaurants (pr) dataset (April 2014)
4. de Melo, G.: Not quite the same: Identity constraints for the Web of Linked Data. In: Proc. of the 27th Conference on Artificial Intelligence. AAAI Press (2013)
5. Ding, L., Shinavier, J., Finin, T., McGuinness, D.L.: owl:sameAs and Linked Data: An Empirical Study. In: International Web Science Conference (2010)
6. Ding, L., Shinavier, J., Shangguan, Z., McGuinness, D.L.: SameAs networks and beyond: Analyzing deployment status and implications of owl:sameAs in linked data. In: Patel-Schneider, P.F., Pan, Y., Hitzler, P., Mika, P., Zhang, L., Pan, J.Z., Horrocks, I., Glimm, B. (eds.) ISWC 2010, Part I. LNCS, vol. 6496, pp. 145–160. Springer, Heidelberg (2010)
7. Ferrara, A., Nikolov, A., Scharffe, F.: Data linking. J. Web Semantics 23(1) (2013)
8. Ghazvinian, A., Noy, N.F., Jonquet, C., Shah, N., Musen, M.A.: What four million mappings can tell you about two hundred ontologies. In: Bernstein, A., Karger, D.R., Heath, T., Feigenbaum, L., Maynard, D., Motta, E., Thirunarayan, K. (eds.) ISWC 2009. LNCS, vol. 5823, pp. 229–242. Springer, Heidelberg (2009)
9. Glaser, H., Jaffri, A., Millard, I.: Managing co-reference on the semantic web. In: WWW 2009 Workshop: Linked Data on the Web, LDOW 2009 (April 2009)
10. Guéret, C., Groth, P., Stadler, C., Lehmann, J.: Assessing linked data mappings using network measures. In: Simperl, E., Cimiano, P., Polleres, A., Corcho, O., Presutti, V. (eds.) ESWC 2012. LNCS, vol. 7295, pp. 87–102. Springer, Heidelberg (2012)
11. Halpin, H., Hayes, P.J.: When owl: sameas isn't the same: An analysis of identity links on the semantic web. In: Proc. of the WWW 2010 Workshop on Linked Data on the Web. CEUR-WS.org Proceedings, vol. 628 (2010)

12. Halpin, H., Hayes, P.J., Thompson, H.S.: When owl: sameas isn't the same redux: A preliminary theory of identity and inference on the semantic web. In: Workshop on Discov. Meaning OntheGo in Large Heter. Data, pp. 25–30 (2011)
13. Heath, T., Bizer, C.: Linked Data: Evolving the Web into a Global Data Space, 1st edn. Morgan & Claypool (2011)
14. Heflin, J., Muñoz-Avila, H.: LCW-based agent planning for the semantic web. In: Ontologies and the Semantic Web Workshop, pp. 63–70. AAAI Press (2002)
15. Hitzler, P., Krötzsch, M., Parsia, B., Patel-Schneider, P.F., Rudolph, S. (eds.): OWL 2 Web Ontology Language: Primer. W3C Recommendation (2009), http://www.w3.org/TR/owl2-primer/
16. Hogan, A., Zimmermann, A., Umbrich, J., Polleres, A., Decker, S.: Scalable and distributed methods for entity matching, consolidation and disambiguation over linked data corpora. J. Web Sem. 10, 76–110 (2012)
17. Lassila, O., Swick, R.R.,. WWW Consortium.: Resource description framework (RDF) model and syntax specification (2004)
18. Pernelle, N., Saïs, F., Safar, B., Koutraki, M., Ghosh, T.: N2R-Part: Identity Link Discovery using Partially Aligned Ontologies. In: International Workshop on Open Data, Paris, France (June 2013)
19. Robinson, G., Wos, L.: Paramodulation and theorem-proving in first-order theories with equality. In: Automation of Reasoning 2: Classical Papers on Computational Logic, pp. 298–313. Springer (1969)
20. Saïs, F., Niraula, N.B., Pernelle, N., Rousset, M.C.: LN2R a knowledge based reference reconciliation system: OAEI2010 results. In: International Workshop on Ontology Matching OM2010, vol. 689. CEUR-WS.org Proceedings (2010)
21. Saïs, F., Pernelle, N., Rousset, M.-C.: Combining a logical and a numerical method for data reconciliation. In: Spaccapietra, S. (ed.) Journal on Data Semantics XII. LNCS, vol. 5480, pp. 66–94. Springer, Heidelberg (2009)
22. Song, D., Heflin, J.: Automatically generating data linkages using a domain-independent candidate selection approach. In: Aroyo, L., Welty, C., Alani, H., Taylor, J., Bernstein, A., Kagal, L., Noy, N., Blomqvist, E. (eds.) ISWC 2011, Part I. LNCS, vol. 7031, pp. 649–664. Springer, Heidelberg (2011)
23. Song, D., Heflin, J.: Domain-independent entity coreference for linking ontology instances. J. Data and Information Quality 4(2), 7:1–7:29 (2013)
24. Symeonidou, D., Armant, V., Pernelle, N., Saïs, F.: SAKey: Scalable Almost Key Discovery in RDF Data. In: Janowicz, K., et al. (eds.) ISWC 2014. LNCS, vol. 8796, pp. 33–49. Springer, Heidelberg (2014)
25. Taheri, A., Shamsfard, M.: Instance coreference resolution in multi-ontology linked data resources. In: Takeda, H., Qu, Y., Mizoguchi, R., Kitamura, Y. (eds.) JIST 2012. LNCS, vol. 7774, pp. 129–145. Springer, Heidelberg (2013)
26. Yves, J.R., Shironoshita, E.P., Kabuka, M.R.: Ontology matching with semantic verification. Web Semantics 7(3), 235–251 (2009)