On Evaluating the Quality of RDF Identity Links in the LOD

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

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

Abstract. The notion of Linked Data is based on the idea that resources from different data sources can be connected by typed links to enable new knowledge that isolated data sources cannot provide on their own. Today, as the Web of Data is proving its importance, and as an huge amount of data is published on the web in form of RDF triples, the quantity of *sameAs* links is extremely growing as different data sources often describe equivalent resources. Since most of the *sameAs* links are discovered automatically and the quality of the data sources can be poor, it is becoming crucial to develop methods for evaluating the quality of these RDF identity links, thus providing a ranking measure of their reliability. In this context, this paper defines a initial methodology for analyzing and evaluating a set of given RDF identity links in the Web of Data, ranking their reliability.

Keywords: Linked Data, Linking Verification, Link Quality

1 Introduction

The Semantic Web [22] aims at providing a common framework that allows data to be shared and reused across applications, enterprises, and communities. As we all know, the term was first coined by Berners-Lee, Hendler and Lassila in 2001 and the authors describes it as a *'Web of Data' that can be processed by machines* [1].

Today, thanks to the standardization and the real adoption of Semantic Web technologies, we are experiencing an unprecedented production of data, published as *Linked Open Data* (LOD, for short). This is leading to the creation of a global data space containing billions of assertions, representing one of the interesting outcome of the Semantic Web: the *Web of Data* [2].

The integration across this Web of Data, however, is performed controversially, due to an operating philosophy, very reminiscent of the one used at the time of the growth of the traditional Web (HTML pages): *'publish anyway -if you can, refine later if necessary -really necessary'*. With the ultimate goal in mind to establish the massive adoption of the Web, this choice has been proven then, as now, really successful but, on the other side, it is generating various

quality problems in the underlying data such as incompleteness, inconsistency and incomprehensibility. These problems affect every application domain, spanning form scientific research to governmental applications, life science and so on.

Linked Data is basically about using the Web to create *typed links* between data from different sources: providers set RDF links [8] from their data sources entities (described by URIs) to related entities in other data sources (other URIs), improving the knowledge related to a specific entity and, thus, the global knowledge in the Web of Data. Most of the RDF links in these sources are *RDF identity links*, defined using the $owl : sameAs$ property, thus expressing that two URI references actually refer to the same thing: the individuals have the same identity.

The discovery and definition of all the RDF identity links involving one or more data sources can be a very long procedure (depending also on the dimension of the data sources) and this process can be performed manually or automatically.

In various domains, there are generally accepted naming schemata [2]. If the link source and the link target data sources both support one or more of these identification schema, the implicit relationship between entities in both data sets can be made explicit as identity links, automatically. When no shared naming schema exist, RDF identity links are usually generated evaluating the similarity of entities within both data sources. This is generally referred as the 'coreference problem' in Semantic Web and there exist already different approaches in the literature (see, for example, [12, 13, 5, 18, 17, 21, 19, 4]).

In the process of injecting all these RDF identity links in the LOD cloud, some errors could have been inserted due both to human mistakes or incorrectness of the used methods [3]. Coreference algorithms, in fact, usually rely on the good quality of the resources, while in many cases, we have to work with resources having poor descriptive values or not precise information.

In this work we investigate and design a general methodology to *evaluate the quality* of existing RDF identity links in the LOD, by looking at the descriptions associate to the instances involved. We suppose that, in case of multiple data sources, a mapping between the schema is already provided. We essentially look at all the functional properties of the two instances and we assess the similarity of their values (in both cases of data- and object-types). We claim that, if conflicts are encountered, the initial RDF identity link can be considered 'nogood', meaning that it requires further investigation (supervised or automatic). In all the other cases, we keep track of the computed similarity values and use them in order to calculate the overall quality of the given RDF identity link.

The remainder of this paper is organized as follows. In Section 2 we present the problem we want to address, including specific reasoning and considerations, while in Section 3 we describe our approach with examples. Finally in Section 4 some concluding remarks and future works are drawn.

2 Formulation of the Problem and Basic Considerations

2.1 The Problem in a Nutshell

The main problem we are addressing is the following.

Given the RDF identity link $sameAs(s, o)$
where s and o are resources of two data sources D_i and D_j respectively
which is the quality of this link?

In order to evaluate the assertion $sameAs(s, o)$, we want to be able to provide a value expressing in some way the *quality* of the assertion. The *quality* value will be a real number in the interval $[0, \ldots, 1]$ and it will represent the reliability value for the identity link. When *quality* is 0 the assertion $sameAs(s, o)$ is considered *nogood* since strong defects have been encountered.

In the following we make some general considerations related to the problem we want to address. These considerations have been the starting point for designing the method we will present in Section 3.

2.2 The Importance of Functional Properties

One first consideration is that, in order to evaluate the quality level of identity between o and s we have to analyze their properties and the associate semantics. This is, in general, a domain-dependent problem, as the importance and the significance of a given property depends on the domain which led to its definition.

One idea is to look at all the *functional* properties involved. Let us suppose that P a functional property. It can be expressed logically as follows [15, 9]:

$$P(x, y) \wedge P(x, y^{'}) \Rightarrow y \equiv y^{'}$$

The semantics of $sameAs$ [15, 9] can be expressed as:

$$sameAs(x, z) \wedge P(x, y) \Rightarrow P(z, y)$$

So if we want to validate the link $sameAs(s, o)$ and we have a mapped functional property P_1, with $P_1(s, w)$ and $P_1(o, w_1)$, and we can assert $w \not\equiv w_1$ then:

$$sameAs(s, o) \wedge P_1(s, w) \wedge P_1(o, w_1) \wedge w \not\equiv w_1 \Rightarrow \bot$$

In case in which P **is a functional datatype property**, the targets w, w_1 are datatype values and the evaluation of the equivalence between them can be performed in different ways. In case of literals, it is possible 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 $'Paris' \equiv' ParisCity'$ and that $'Paris' \not\equiv' Milan'$ and so on. Thus the evaluation is based on understanding if two values w, w_1 belong to the same cluster

class. Another situation arises when the possible values of the property P are well defined as in the case of enumeration, dates, geographical data, some types of measures (numeric) and so on. In this cases, the evaluation is again a simple comparison of the values. If they are the same, their are equivalent, otherwise they are not equivalent. Finally, in case in which pre-processing classification is not possible, we have to compute each time a similarity measure between w and w_1. This is always a tricky part, because it is impossible to establish 'a-priori' a good similarity function (as discussed below).

In case in which P **is a functional object property** then the targets w, w_1 are instances and, thus, we need to evaluate the RDF identity link $sameAs(w, w_1)$. This can be done recursively using the same type of reasoning (or similar) as for the case of $sameAs(s, o)$.
It is tricky (and still an open problem) to understand how long it is necessary to navigate the RDF graph in order to be able to collect enough information (but not too much) for assessing the quality of the assertion $sameAs(s, o)$. It is obvious that, the more details we analyze the more we can obtain accurate evaluation, but - as always - we need a tradeoff between performance and accuracy. Moreover, it is possible that, by browsing the LOD cloud, we could end up in analyzing instances belonging to different data sources which could be erroneous and incomplete, thus guiding us in a wrong evaluation of a specific identity link. This wrong assumption will have a weight in the evaluation of the initial $sameAs(s, o)$ even if it actually should not be there since the beginning.

Thus, our main idea is that, in order to evaluate the quality of identity links of two instances s and o, we may consider their description in terms of datatype properties and object properties. This intuition may appear as analogous as what is used during the discovery of identity links, where the property values are used to compute/infer a *similarity* degree between s and o. Nevertheless, in case of *quality* evaluation, the instance description is exploited in a different way and for a different objective. Indeed, to evaluate the link quality, we exploit the properties in order to measure the quality of information that can be inferred thanks to the *owl:sameAs* semantics. For instance, if we consider a very simple example of two instances (books) b_1 and b_2 described using two datatype properties *isbn* and *pages*. We assume 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* (see rule (1) above), we infer that the values of the property *pages* are equivalent. If they are not, one can detect a conflicting case entailed from the semantics of $sameAs(b_1, b_2)$.

2.3 Similarity Measures between datatypes

Another consideration is that, for better assessing the quality of an RDF identity link, a *suitable* similarity measure must be chosen, possibly a personalized similarity measure for each functional property considered. We do know that a huge

number of similarity measures already exist in the literature, and to present a complete survey is out of the scope of this paper.

In general, similarity measures aim at quantifying the extent to which objects resemble each other. In particular, given two sequences of measurements X and Y the similarity (dissimilarity) between them is a *measure that quantifies the dependency (independency) between the sequences* [6].

To clarify these concepts, let us consider two objects A and B, a is the number of features (characteristics) present in A and absent in B, b is the number of features absent in A and present in B, c is the number of features common to both objects, and d is the number of features absent from both objects. Thus, c and d measure the present and the absent matches, respectively, i.e., *similarity*; while a and b measure the corresponding mismatches, i.e., *dissimilarity*.

Since the accuracy relies on the choice of an *appropriate measure* (which is still a very complex task), many researchers have taken elaborate efforts to find the most *meaningful* similarity and distance measures over a hundred years. Each of them is differently defined by its own synthetic properties. Some include negative matches such as the *Pearson* measure [16] or the *Tanimoto* (cited for example in [10]) and some do not as the *Dice & Sorenson* [20] or the *Jaccard* [11]. Some use simple count difference and some utilize complicated correlation. In our approach, we are planning to utilize specific similarity measures for each functional datatypes property, trying to improve the accuracy of the results.

3 Evaluating the Quality of an Existing RDF Entity Link

As said before, in order to evaluate the quality of an existing RDF identity link in the LOD, we decided to look at the descriptions associated to the instances involved. In this paper, we assume that, in case of multiple data sources, equivalence mappings between properties are provided. In this Section we describe our method, providing explanatory examples in order to let the reader understand our reasoning.

3.1 General Idea

Let us consider a RDF identity link $l = sameAs(x, y)$ where x and y are two instances belonging to (possibly) different data sources. A very general explanation of the method is that in evaluating the RDF identity link l, we need to compute a value $q_{<x,y>}$ which quantifies the *quality* of l, according to specific criteria. In our case $q_{<x,y>}$ is computed with the following formula:

$$q_{<x,y>} = \prod_{i=1}^{k} b_{P_i}$$

where k is the number of functional properties considered and, for every i, b_{P_i} is the similarity value computed regarding the functional property P_i.

3.2 Computing the Similarity values for Functional Properties

To compute $q_{<x,y>}$ we need to compute every b_{P_i}. Let us define $\{P_{P_i}\}$ with $i = \{1, \ldots, n\}$ as the set representing all the functional datatype properties holding for both x and y and $\{P_{Q_j}\}$ with $j = \{1, \ldots, m\}$ as the set representing all the functional object properties holding again for both x and y.

In order to compute every b_{P_i}, we define two evaluation functions:

1. $eval_P()$, for computing a partial similarity value related to all the functional *datatype* properties
2. $eval_Q()$, for computing a partial similarity value using only functional *object* properties

Successively, we combine the results in order to obtain the overall evaluation.

Computing $eval_P()$: In Algorithm 1 we present the pseudo-code for computing $eval_P()$. In the code, the function $computeSimilarity(a, b)$ can be any similarity measure useful for datatypes (see Section 2.3). The underlying idea is to combine together all the similarity values of each functional datatype properties.

Algorithm 1: $eval_P()$, $SimDP$ provides a similarity score for all the functional datatype properties $\{P_{P_i}\}$ of x and y

Input: Two finite sets $O = \{o_1, \ldots, o_n\}$ and $O^{'} = \{o_1^{'}, \ldots, o_n^{'}\}$ of the values of the functional datatype properties DP holding for x and y
Output: The similarity score for all the functional datatype properties

```
1 SimDP ← 1
2 for i ← 1 to n do
3 |   b_i ← computeSimilarity(o_i, o_i')
4 |   SimDP ← SimDP × b_i
5 return SimDP
```

The multiplication $SimDP = SimDP \times b_i$ is done in order to keep track of all the possible inconsistencies. When two datatype values $o_i, o_i^{'}$ are exactly the same, $computeSimilarity(o_i, o_i^{'})$ returns 1 which basically does not alter the value of $SimDP$.

Note that, in case of a pre-processing step in which all the possible values have been already organized in clusters, the function in $computeSimilarity(o_i, o_i^{'})$ simply verifies if o_i and $o_i^{'}$ belong to the same cluster, thus speeding the entire process.

Computing $eval_Q()$: In Algorithm 2 we present the pseudo-code for the computation of $eval_Q()$, which returns a value representing the combination of all

Algorithm 2: $eval_Q()$, $SimOP$ provides a similarity score for all the functional object properties $\{P_{Q_j}\}$ of x and y

Input: Two finite sets $O = \{o_1, \ldots, o_m\}$ and $O' = \{o_1', \ldots, o_m'\}$ of instances of the range of the object properties OP holding for x and y
Output: The similarity score

```
SimOP ← 1
for j ← 1 to m do
    b_j ← eval_P(o_j, o_j')
    SimOP ← SimOP × b_j
return SimOP
```

the similarity values between the instances in all the functional object properties related to x and y.

Note that, in the computation of $SimOP$, we consider only the functional datatype properties of the instances target in the initial properties. We, thus, browse the RDF graph only for one level. This is done because we believe that deepening the browsing will eventually add too much noise and the type of computation we have chosen (multiplication) does not 'forgive' any inconsistency.

Computing $q_{<x,y>}$: The overall computation of the quality of the identity link between x and y is shown in Algorithm 3. In this case we combine both the b_i coming from functional datatype properties and those coming from functional object properties.

$q_{<x,y>}$ provides the quality evaluation for the identity link $sameAs(x, y)$. When $q_{<x,y>}$ equals 0 the pair (x, y) is added to the *noGood* set.

3.3 Understanding by-examples

In Figure 1 and Figure 2 we depict two examples in order to understand the general idea of our approach.

In both cases, we suppose that the instances x and y belong to two different data sources D_1 and D_2 (let be ns_1 and ns_2 the relative namespaces), and there exists an RDF identity link $sameAs(x, y)$ in the LOD that we want to evaluate. Additionally, an equivalence mapping between existing properties is provided so that:

- $ns_1 : hasTitle$ is mapped into $ns_2 : hasName$,
- $ns_1 : authoredBy$ is mapped into $ns_2 : hasAuthor$,
- $ns_1 : hasName$ is mapped into $ns_2 : hasCompleteName$,
- $ns_1 : wasBornIn$ is mapped into $ns_2 : wasBornIn$,
- and so on ...

Also we know that $ns_1 : hasTitle$, $ns_2 : hasName$, $ns_1 : authoredBy$, $ns_2 : hasAuthor$, $ns_1 : hasName$, $ns_2 : hasCompleteName$ are all functional

Algorithm 3: $eval()$, $q_{<x,y>}$ provides a the quality evaluation for the identity link $sameAs(x, y)$

Input:
- D_i, D_j: two RDF datasets,
- $sameAs(x, y)$: the identity link to evaluate, where $x \in D_i$ and $y \in D_j$,
- M: the set of equivalent functional properties (p_i, p'_i) holding for both x and y.

Output: $q_{<x,y>}$: the quality of the identity link between x and y

```
1  DP ← getDataTypeProperties(M)
2  OP ← getObjectProperties(M)
3  q_<x,y> ← 1; O ← ∅; O' ← ∅
4  for i ← 1 to |DP| do
5  |   O ← O ∪ {o_i | p_i(x, o_i) ∈ D_i}
6  |   O' ← O' ∪ {o'_i | p'_i(y, o'_i) ∈ D_j}
7  q_<x,y> = q_<x,y> × eval_P(O, O')
8  O ← ∅; O' ← ∅
9  for j ← 1 to |OP| do
10 |   O ← O ∪ {o_i | p_i(x, o_i) ∈ D_i}
11 |   O' ← O' ∪ {o'_i | p'_i(y, o'_i) ∈ D_j}
12 q_<x,y> ← q_<x,y> × eval_Q(O, O')
13 return q_<x,y>
```

properties (at least - some of them could be inverse functional properties but we do not care).

In the first example (Figure 1), the two instances x and y actually represent the very same artwork, namely the painting 'La Gioconda', thus we expect to compute an high quality value (possibly 1). In order to evaluate $sameAs(x, y)$, we analyze the knowledge associated to the instances, looking at all the attached the functional properties, both datatype properties ($ns_1 : hasTitle$, $ns_2 : hasName$) and object properties ($ns_1 : authoredBy$, $ns_2 : hasAuthor$).

For the datatype properties (in this case only the title of the artwork) we compute $eval_P()$ for the pair (w, w_1): w and w_1 are exactly the same and thus $SimDP = 1$. The message is that we have not found inconsistencies, regarding the functional datatype properties and thus we can continue the evaluation. For the object properties (in this case only the authors of the painting) we look at the functional datatype properties related to the targets (the authors a and b), here only the 'name' of the author. Also in this case, the strings w_2 and w_3 are exactly the same so, $SimDP$ (for a and b) will be 1 and thus will be $SimOP$ for x and y.
$q_{<x,y>}$ will be simply $SimDP \times SimOP = 1$. We conclude that, at the level of details for which we studied the $sameAs$ assertion, no inconsistencies have been found, thus the identity link is solid enough.

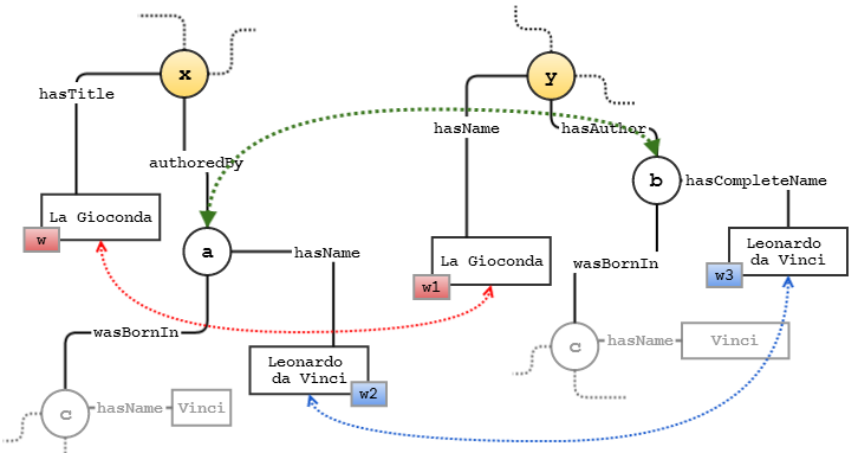

Fig. 1. An example to clarify our approach. The instance x and y belong to two different data sources D_1 and D_2, and we want to evaluate $sameAs(x, y)$. The rounded colored arrows show the targets we take into account in the overall evaluation process. Note that we stop at the first level of the RDF graph (starting from x or y). In this case the result is 1, thus the identity link can be considered *solid enough*

In the second example (Figure 2), the two instances x and y represent two different artworks, namely the painting 'La Gioconda' and the opera 'La Gioconda' respectively, and we expect to compute a low quality value (possibly 0). For sake of simplicity, let us suppose also that all the datatype properties' values have been organized in clusters (pre-processing step). Thus, for example, the values '*Leonardo da Vinci*' and '*Amilcare Ponchielli*' belong to different clusters. Again, in order to evaluate $sameAs(x, y)$, we analyze the knowledge associated to the instances, looking at all the attached the functional properties, both data- and objectype properties.

For the datatype properties, we compute $eval_P()$ for the pair (w, w_1). As in the case of the first example, w and w_1 are exactly the same and $SimDP = 1$. For the object properties, the authors of the artworks a and b, we look at the related functional datatype properties, namely their name (w_2, w_3). We detect that w_2 ='*Leonardo da Vinci*' $\not\equiv$ w_3 ='*Amilcare Ponchielli*' (since they are not in the same cluster) and their similarity is 0. As a consequence, also $SimDP$ (for the authors a and b) will be 0.
$q_{<x,y>}$ will be simply $SimDP \times SimOP = 0$, so our method will return *nogood* as evaluation of the quality of the identity link $sameAs(x, y)$. Basically there are inconsistencies, and the *sameAs* link needs to be discarded.

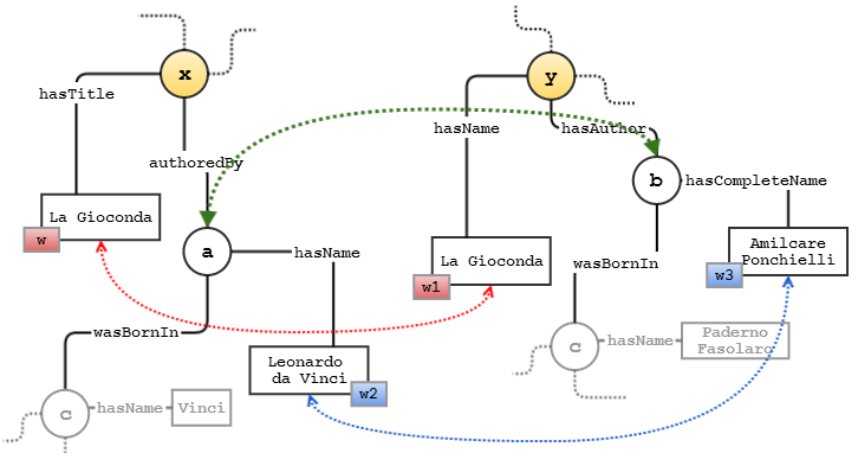

Fig. 2. Another example. Again, the instance x and y belong to two different data sources D_1 and D_2, and we want to evaluate $sameAs(x, y)$. In this case the result is 0 thus the identity link cannot be considered *solid enough*, there are inconsistencies (in this case the name of the author). $sameAs(x, y)$ needs to be discarded.

4 Concluding Remarks

In this paper we argued on the problem of automatically evaluating RDF identity links defined in the LOD cloud. We designed a general evaluation method which basically relies on the descriptions associated to the instances x and y involved in an identity links $sameAs(x, y)$. In [7] a framework is described which assesses the quality of Linked Data mappings using specific network metrics. In that case the main focus is on the overall network stability, and the authors study different quality metrics with respect to the insertion of identity links in the LOD graph. The paper proves that those metrics are not good enough, expressing the need of a new metric for correctly detecting 'bad links'.

Given an identity links $sameAs(x, y)$ in the LOD, our method analyzes all the functional properties of x and y and assesses the similarity of their values (in both cases of datatype and object properties). We provided examples to illustrate our strategy.

We know that we designed a 'rude' approach, since we '*do not forgive*' conflicts in properties immediately related to the initial instances or to those instances directly connected to the initial ones. This is the first attempt to design an evaluation strategy for RDF identity links. We are now testing it with respect to benchmark data sources of different sizes and from different domains (such those available in the Instance Matching Track, via the Ontology Alignment Evaluation Initiative [14]).

The next step will be to study new formulas for computing $q_{<x,y>}$, possibly taking into account the significance of every property. This means, for example,

that it could be possible to associate a weight $weight_i$ to each property P_i (and thus to each similarity measure b_{P_i}) in order to correctly 'balance' each property in the evaluation process. Also we want to take into account not-functional datatype properties, such as those that respect 'local completeness'. In such cases, the set of property values can be compared, providing a evaluation of their resemblance.

Finally, we would like to provide the 'explanation' of the evaluation value *quality*, maybe by keeping track of the properties involved in the conflicts encountered.

Acknowledgment

This work is supported by the French National Research Agency: Quality and Interoperability of Large Catalogues of Documents project (QUALINCA-ANR-2012-CORD-012-02). The authors would like to thank the reviewers for the helpful suggestions.

References

1. Tim Berners-Lee, James Hendler, and Ora Lassila. The semantic web. *Scientific American*, 284(5):34–43, May 2001.
2. Christian Bizer, Tom Heath, and Tim Berners-Lee. Linked data - the story so far. *Int. J. Semantic Web Inf. Syst.*, 5(3):1–22, 2009.
3. Li Ding, Joshua Shinavier, Tim Finin, and Deborah L. McGuinness. owl:sameAs and Linked Data: An Empirical Study . In *Proceedings of the Second Web Science Conference*, Raleigh NC, USA, April 2010.
4. Alfio Ferrara, Andriy Nikolov, and François Scharffe. Data linking. *J. Web Sem.*, 23:1, 2013.
5. 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.
6. A.Ardeshir Goshtasby. Similarity and dissimilarity measures. In *Image Registration*, Advances in Computer Vision and Pattern Recognition, pages 7–66. Springer London, 2012.
7. Christophe Guéret, Paul Groth, Claus Stadler, and Jens Lehmann. Assessing linked data mappings using network measures. In *Proceedings of the 9th International Conference on The Semantic Web: Research and Applications*, ESWC'12, pages 87–102, Berlin, Heidelberg, 2012. Springer-Verlag.
8. Tom Heath and Christian Bizer. *Linked Data: Evolving the Web into a Global Data Space*. Morgan & Claypool., 1st edition edition, 2011.
9. Pascal Hitzler, Markus Krötzsch, Bijan Parsia, Peter F. Patel-Schneider, and Sebastian Rudolph, editors. *OWL 2 Web Ontology Language: Primer*. W3C Recommendation, 27 October 2009. Available at http://www.w3.org/TR/owl2-primer/.
10. Yangsheng Xu Huihuan Qian, Xinyu Wu. *Intelligent Surveillance Systems*. Springer-Verlag, 2011.
11. Paul Jaccard. Étude comparative de la distribution florale dans une portion des alpes et des jura. *Bulletin del la Société Vaudoise des Sciences Naturelles*, 37:547–579, 1901.

12. J. Murdock M. Yatskevich, C. Welty. Coreference resolution on rdf graphs generated from information extraction: first results. In *In Proceedings of Web Content Mining with Human Language Technologies Workshop*, 2006.
13. Andriy Nikolov, Victoria S. Uren, Enrico Motta, and Anne N. De Roeck. Handling instance coreferencing in the knofuss architecture. In P. Bouquet, H. Halpin, H. Stoermer, and G. Tummarello, editors, *Identity and Reference on the Semantic Web International Workshop*, volume 422 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2008.
14. OAEI. Ontology alignment evaluation initiativ. http://oaei.ontologymatching.org/. Accessed: 2014-03-01.
15. W3C OWL Working Group. *OWL 2 Web Ontology Language: Document Overview*. W3C Recommendation, 27 October 2009. Available at http://www.w3.org/TR/owl2-overview/.
16. K. Pearson. Contributions to the mathematical theory of evolution. *A Philosophical Transactions of the Royal Society of London*, 185:71–110, 1894.
17. Nathalie Pernelle, Fatiha Saïs, Brigitte Safar, Maria Koutraki, and Tushar Ghosh. N2R-Part: Identity Link Discovery using Partially Aligned Ontologies. In *International Workshop on Open Data*, Paris, France, June 2013.
18. Fatiha Saïs, Nathalie Pernelle, and Marie-Christine Rousset. Combining a logical and a numerical method for data reconciliation. *Journal of Data Semantics*, 12:66–94, 2009.
19. Dezhao Song and Jeff Heflin. Domain-independent entity coreference for linking ontology instances. *J. Data and Information Quality*, 4(2):7:1–7:29, March 2013.
20. T. Sørensen. A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on Danish commons. *Biologiske Skrifter / Kongelige Danske Videnskabernes Selskab*, 5:1–34, 1948.
21. Aynaz Taheri and Mehrnoush Shamsfard. Instance coreference resolution in multi-ontology linked data resources. In Hideaki Takeda, Yuzhong Qu, Riichiro Mizoguchi, and Yoshinobu Kitamura, editors, *JIST*, volume 7774 of *Lecture Notes in Computer Science*, pages 129–145. Springer, 2012.
22. World Wide Web Consortium (W3C). W3c semantic web activity. http://www.w3.org/2001/sw/. Accessed: 2014-01-30.