Profiling Similarity Links in Linked Open Data

Blerina Spahiu*, Cheng Xie†, Anisa Rula*, Andrea Maurino*, Hongming Cai†
*University of Milano - Bicocca
{spahiu, rula, maurino}@disco.unimib.it
†Shanghai Jiao Tong University
{chengxie, hmcai}@sjtu.edu.cn

***Abstract*—Usually the content of the dataset published as LOD is rather unknown and data publishers have to deal with the challenge of interlinking new knowledge with existing datasets. Although there exist tools to facilitate data interlinking, they use prior knowledge about the datasets to be interlinked. In this paper we present a framework to profile the quality of `owl:sameAs` property in the Linked Open Data cloud and automatically discover new similarity links giving a similarity score for all the instances without prior knowledge about the properties used. Experimental results demonstrate the usefulness and effectiveness of the framework to automatically generate new links between two or more similar instances.**

I. Introduction

The idea behind Linked Open Data (LOD) is that datasets should be linked in order to promote interoperability and integration among large data collections on the Web [4]. Data interlinking focuses on identifying equivalent entities by determining the similarity between their entity descriptions to represent the fact that they refer to the same real world entity in a given domain. This similarity between two entities often is represented by using the standard `owl:sameAs` property.

In the context of LOD 2014 [14], we count 1,532,323 `owl:sameAs` triples. DBpedia is the dataset with the highest number of `owl:sameAs` triples (792,268) followed by rdfize.com (215,716), ontologycentral.com (166,020) and linked-statistics.org(144,543). From this first analysis we may deduce that datasets in the LOD cloud are sparsely connected due to the fact that, usually, data publishers are not aware about the content of the datasets and thus the task of interlinking is not straightforward since it requires previous knowledge on the content of the datasets. In LOD cloud the resource *nyt:88184832497785382991*[1] from the NYTimes dataset is linked through the sameAs property with *dbpedia:Senegal*[2]. The first resource describes Woods Hole, a place in the town of Falmouth in Barnstable County, Massachusetts, US, while the second describes Senegal, a country in Africa. These two resources have wrong sameAs links because they do not represent the same entity in the real world. Also in the current LOD cloud *gn:2964180/gaillimh.html*[3] belonging to GeoNames dataset and the resource in LinkedGeoData *lgdo:node582043319*[4] are not linked with the sameAs property even though they refer to the same city in Ireland, Galway.

The problem of finding similar objects among heterogeneous data sources is a well studied problem in the Semantic Web Community. This task is performed on the basis of the evaluation of the degree of similarity among descriptions of entities. Two survey papers review and summarize the approaches on data interlinking [16], [17]. The work presented in this paper attempts to exploit the actual state of `owl:sameAs` links in the cloud, and investigate to which extent we can automatically find other similar pairs in the datasets without prior knowledge about their content. This will help applications built on top of LOD datasets to discover more links for the same entity, thus enriching the information about an instance with other properties found in other datasets. To achieve this goal, we developed a framework to identify ambiguities and suggest possible inconsistencies and incompleteness. The framework implements two similarity finding techniques one for string similarity and one for numeric similarity, to analyze the quality of existing links and propose new ones to resolve the identified ambiguities. First, we extract all properties for instances which have an `owl:sameAs` property between two datasetes in the cloud and transform them into tables. Secondly, we calculate a similarity score comparing each row between tables of datasets we want to find similar entities. We consider a similarity threshold greater than 0.9 for the instances to be categorised as similar. and test our framework on 13 LOD datasets.

This work provides contributions to: (1) a framework to automatically find similar pairs between datasets published as LOD, (2) an evaluation model to estimate the quality of existing sameAs links.

The rest of this paper is organized as follows: Section II discusses the approach to automatically find similar pairs between datasets; Section III introduces the experiments to evaluate the effectiveness and usefulness of the framework. Related work is discussed in Section IV. In Section V we draw conclusions and future work.

II. Overview of the Approach

The problem of discovering same entities in different datasets is quite well known in record linkage [5] and ontology matching community [2]. The proposed approach to discover links between datasets in LOD is shown in Figure 1.

Our approach consists of four processes: i) Data Collection; ii) Data Preparation; iii) Similarity Model; and iv) Linkage Discovery. In the following we describe each process in detail.

[1] ny - http://data.nytimes.com/

[2] dbpedia - http://dbpedia.org/resource/

[3] gn - http://geonames.org

[4] lgdo - http://linkegeodata.org/tiplify/

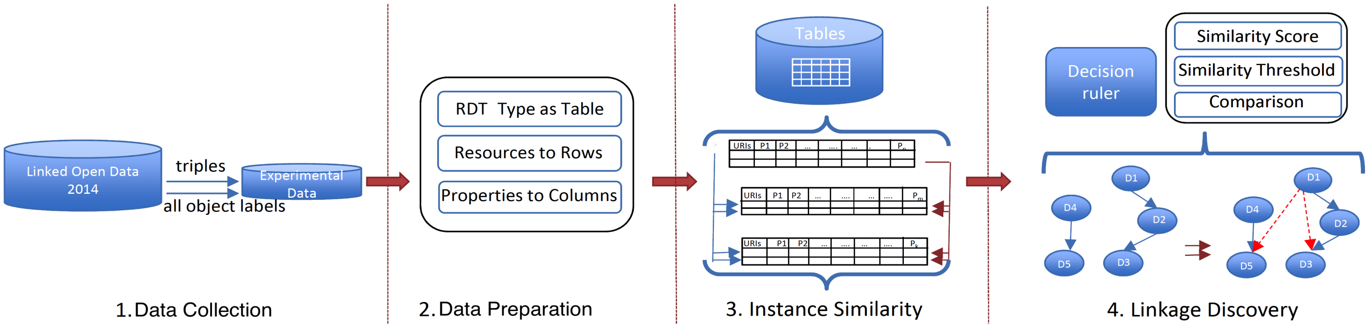


Fig. 1. Pipeline for similar entities finding in LOD

A. Data Collection

To evaluate the quality of owl:sameAs links we consider datasets from the LOD cloud 2014[5]. While the subject is considered to be internal to the dataset for which we want to find similar links, the object may be internal or external to this dataset. For each RDF triple we check if the extracted object is a resource or not. In case of a resource, the description of this resource occurring at subject position is also collected in order to have the complete information between the subject and the object linked through an owl:sameAs property.

B. Data Preparation

As our aim is to automatize the process of similar instance finiding it is more easy to work with tables rather than with triples. Similarity between instances can be seen as a problem of finding similar rows between different tables of different datasets. The idea of "DBpedia as Tables"[6] inspired us to analogously create our matching tables. For each class we create one table. Figure 2 shows how to create Linked Data as Tables. As an example, we consider two instances of the class State from GeoNames dataset, named Salvador and Norway having many properties (illustrated by arrows) and their corresponding values (illustrated by squares). We transpose this information into tables where in the first row, the local name of the properties are places, while the first column contains the URIs of the instances and the remaining cells contain the values for each property for the corresponding class. Once we built the tables we check if instances have more than four properties. If not they are removed from the tables. We create the LabelLike group which comprises the following properties: *label, name, title, text, comment, subject* and *abstract*. For each instance, we check if it has at least one property belonging to the LabelLike group. If instances do not have one of those properties, they are not considered for similarity calculations, because it is very difficult even for humans to find similar instances if they do not share at least one of the values for the properties in LabelLike group and if the number of properties is very low.

C. Similarity Model

All string property values are tokenized at special characters such as: /, _, :, ;, # and at capital letters. We use two formulas to calculate the similarity for properties value: String Similarity and Numeric Similarity.

1) String Similarity: For each cell containing a string value we used the following formula to calculate the similarity score [3]:

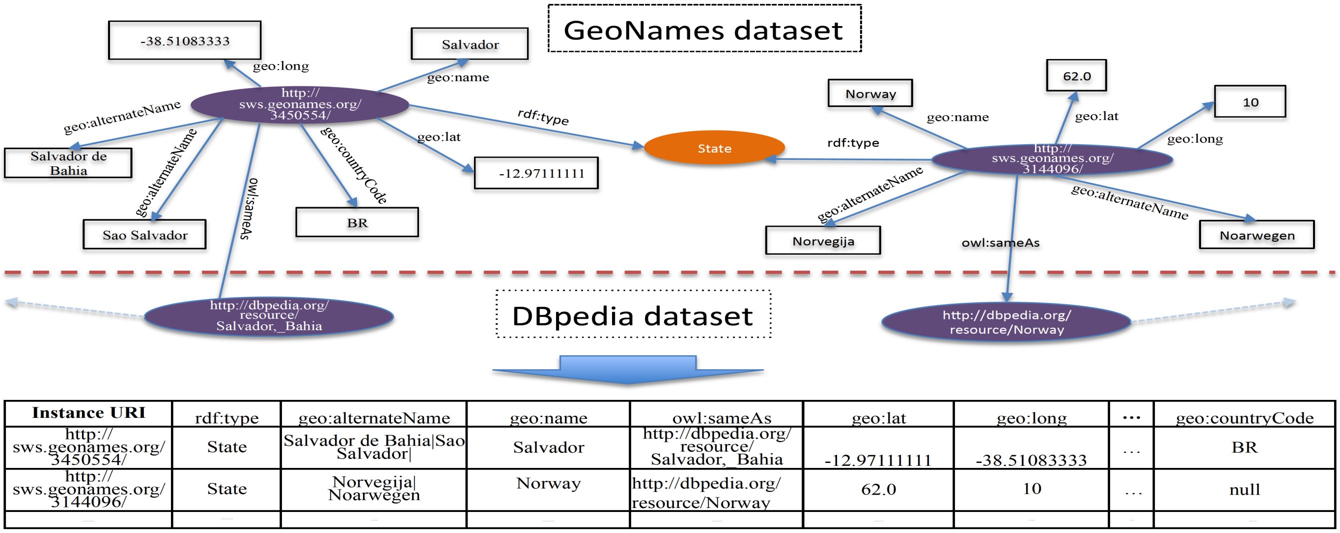
$$S(s,l) = \frac{\sum_{i=1}^{|s|} Max\{Edit(s[i], l[1..|l|])\}}{|s| + |l| - \sum_{i=1}^{|s|} Max\{Edit(s[i], l[1..|l|])\}} \qquad (1)$$

where s and l are string sets, s refers to the shortest set while l refers to the longest set. $S(s,l)$ gives the similarity score between set s and l. $Edit(s[i], l[1..|l|])$ calculates the similarity between s[i], where *i=0,...n* and n is the number of strings in the set to all elements in l by using Levenshtein distance metrics. $Max\{Edit(s[i], l[1..|l|])\}$ has a value from [0,1]. In Figure 3, the property *geo:alternateName* has two values. In cases when a property has more than one value the similarity is calculated for each of them. In this example two values of the property alternateName, from Geonames dataset are, *Salvador de Bahia* | *Sao Salvador*. In LinkedGeo dataset the value of the property *label* is *Salvador*. In the above formula the shortest set s(*i*) is *Salvador* equal to 1, while the longest set l(*l*) is *Salvador de Bahia* | *Sao Salvador* equal to 2. We use Levenshtain distance to measure the similarity score between the values *Salvador de Bahia* and *Sao Salvador* from GeoNames and *Salvador* from LinkedGeoData, respectively 0.3 and 0.67. In the numerator part of the formula we select the maximum value between them, which in our example is 0.67. The denominator is equal to 2.23 (as *s=2, l=1*and *Max Edit = 0.67*). The similarity score between the values for the property alternateName and label is 0.3 (0.67/2.23). In the same way, we iterate through all the values of the cells. Note that we do not make an aligment between the headers of the tables when we calculate the similarity score.

2) Numeric Similarity: For each cell with a numeric value we used the following formula to measure the similarity score:

$$S(n_1,n_2) = \begin{cases} 0, & if \quad |n_1 - n_2| > Min\{RanOf(n_1), RanOf(n_2)\} \\ 1 - \dfrac{|n_1 - n_2|}{Min\{RanOf(n_1), RanOf(n_2)\}}, & \end{cases}$$

[5] http://data.dws.informatik.uni-mannheim.de/lodcloud/2014/ISWC-RDB/
[6] http://wiki.dbpedia.org/services-resources/downloads/dbpedia-tables



Instance URI	rdf:type	geo:alternateName	geo:name	owl:sameAs	geo:lat	geo:long	**…**	geo:countryCode
http://sws.geonames.org/3450554/	State	Salvador de Bahia|Sao Salvador|	Salvador	http://dbpedia.org/resource/Salvador,_Bahia	-12.97111111	-38.51083333	…	BR
http://sws.geonames.org/3144096/	State	Norvegija|Noarwegen	Norway	http://dbpedia.org/resource/Norway	62.0	10	…	null
…	…	…	…	…	…	…	…	…

Fig. 2. Linked Data as Tables

where n_1 is the numerical value of the cell in one table, and n_2 is the numerical value of the cell in the other table. The *RanOf* gives the value range of the numerical values in that column. This formula helps us to compare different numerical values. As we are not aware about the properties that are being compared, sometimes the comparison is not straightforward. For instance, comparing only numerical values can be ambiguous, e.g. the coordinates with population. Suppose to find in a table a cell with the value 34.6458201 and a cell in the other table containing the value 346,458,201. Using the formula (2) for numeric similarity we can deduce that the similarity is 0 because n_1 - n_2 is 346458166,3541799 which is greater than the minimum value range of both columns. Therefore, these two cells cannot be compared. The similarity score between –12.97111111 from GeoNames and –12.9816356E1 from LinkedGeo using formula (2), is 0.998.

3) Aggregation: To calculate the similarity score between two instances (two different rows in tables), we consider only property values, for which the similarity score is greater than 0.9. Thus, in the example above to calculate the similarity score between the first instance of the GeoNames dataset and the first instance of the LinkedGeoData dataset, we consider only the cells with the values *Salvador*, –12.9816356E1 and –3.8482077100000005E1. Respectively for these values, the similarity score is, 1, 0.998, 0.999. To calculate the similarity score for these two instances we aggregate the similarity score of each cell weighting all values using the geometric progression of 75% increase. We use this aggregation model to reward the properties for which the similarity value is grater than 0.9. If only one cell has the similarity score greater than 0.9 then for the aggregation, this score is multiplied by 0.75. If two cells have similarity score greater than 0.9, then their score is multiplied by (0.75 + 0.75*0.25)/2. If three columns have similarity score greater than 0.9 their score is multiplied by (0.75 + 0.75*0.25 + 0.75*0.0625)/3. The similarity score of the first instance of the GeoNames dataset and the first instance of the LinkedGeoData dataset is equal to ((1+0.998+0.999)/3)*0.9843 = 0.9833. Note that the number of properties value with similarity score greater than 0.9 and the aggreegated score are tunnable parameters. The more properties with similarity score greater than 0.9 can contribute to the aggregated score, the more confident we are to categorise these instances as similar. Also, the greater the threshold for cell similarity, the more confident we are to categorise these two instance as similar.

D. Linkage Discovery

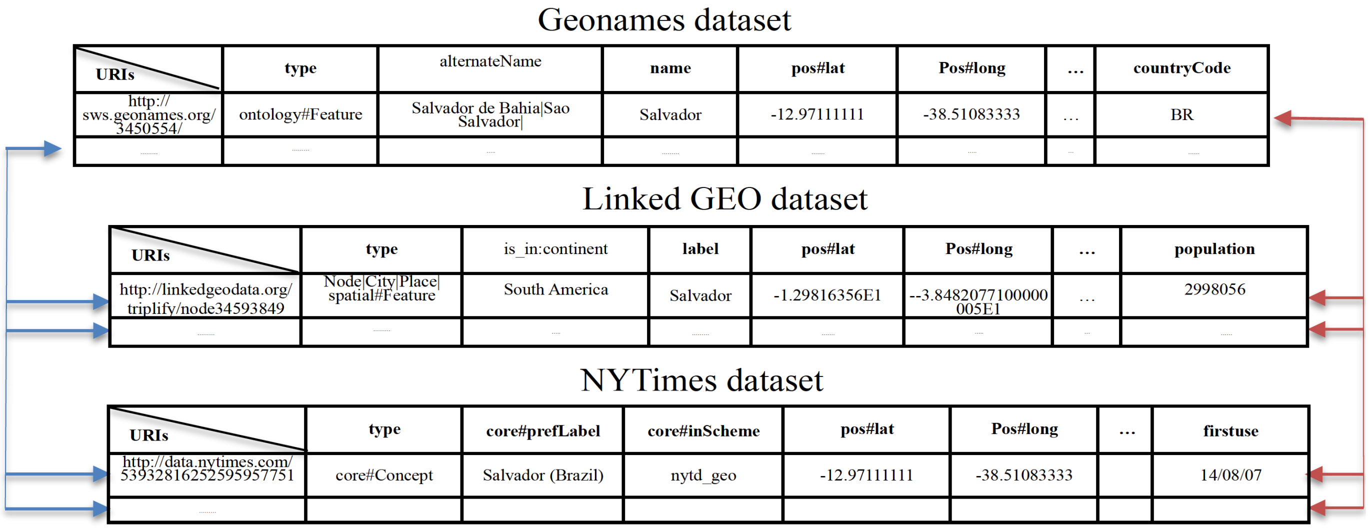
After calculating the similarity score for each instance, we consider as sameAs instances, those for which the agregated similarity score is greater than 0.9. We trained different values for this threshold as shown in Figure 4 and we can observe that for a threshold equal to 0.9 our approach reaches the best performance, where precision has the highest value with respect to recall and F-measure. Our framework is precision oriented. In this step of the approach, we discover sameAs links, filtering only those instances with agregated similarity score greater than 0.9.

III. Evaluation

A. Dataset and Gold Standard

Dataset. To evaluate our framework we used the datasets and the data interlinking information in LOD cloud 2014. For our experiments we consider GeoNames from the geographic domain as the dataset for which we want to find similar instances. The number of outdegree links from this dataset to the other datasets is 20, while the number of indegree links from other datasets to GeoNames is 134 [14]. In the LOD cloud, GeoNames has 7135 sameAs links (incoming and outgoing).

Gold Standard. We consider as Gold Standard (GS) the `owl:sameAs` links between GeoNames and other datasets



Geonames dataset

<table>
<tr><th>URIs</th><th>type</th><th>alternateName</th><th>name</th><th>pos#lat</th><th>Pos#long</th><th>…</th><th>countryCode</th></tr>
<tr><td>http://sws.geonames.org/3450554/</td><td>ontology#Feature</td><td>Salvador de Bahia|Sao Salvador|</td><td>Salvador</td><td>-12.97111111</td><td>-38.51083333</td><td>…</td><td>BR</td></tr>
<tr><td>……</td><td>……</td><td>……</td><td>……</td><td>……</td><td>……</td><td>……</td><td>……</td></tr>
</table>

Linked GEO dataset

<table>
<tr><th>URIs</th><th>type</th><th>is_in:continent</th><th>label</th><th>pos#lat</th><th>Pos#long</th><th>…</th><th>population</th></tr>
<tr><td>http://linkedgeodata.org/triplify/node34593849</td><td>Node|City|Place|spatial#Feature</td><td>South America</td><td>Salvador</td><td>-1.29816356E1</td><td>--3.8482077100000005E1</td><td>…</td><td>2998056</td></tr>
<tr><td>……</td><td>……</td><td>……</td><td>……</td><td>……</td><td>……</td><td>……</td><td>……</td></tr>
</table>

NYTimes dataset

<table>
<tr><th>URIs</th><th>type</th><th>core#prefLabel</th><th>core#inScheme</th><th>pos#lat</th><th>Pos#long</th><th>…</th><th>firstuse</th></tr>
<tr><td>http://data.nytimes.com/53932816252595957751</td><td>core#Concept</td><td>Salvador (Brazil)</td><td>nytd_geo</td><td>-12.97111111</td><td>-38.51083333</td><td></td><td>14/08/07</td></tr>
<tr><td>……</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table>

Fig. 3. Instance Similarity Finding

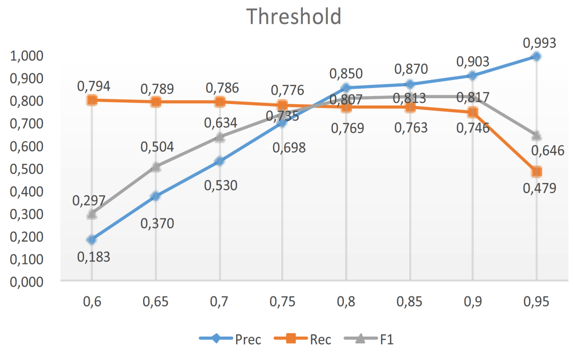


Fig. 4. Framework performance tunning similarity threshold

that already exist in LOD cloud[7]. The first column of the Table I shows the distribution of owl:sameAs links for the experimantal datasets in the current state of the LOD cloud.

B. Results

As described in Section II, we initially extract 5,890 triples having `owl:sameAs` property, where the subject is from GeoNames dataset and the object is from other datasets. For each triple we check if the object is a resource or not. If yes, we also extract the information for that resource appearing in the subject position of any triple in the cloud. The number of overall extracted triples are 587,985. For each instance in `owl:sameAs` triples we check how many properties it has. We do not consider those instances for which the number of properties is smaller than four and do not have a property from the LabelLike group. We consider this requirenment because it is very difficult even for humans to decide if two resources are the same, having only four properties and none of them being from LabelLike group. After applying these filters in our experimental data we have 1,798 `owl:sameAs` that link to 610 distinct instances from GeoNames and 1,798 instances from target datasets. Triples are then transposed into tables as described in Section 2.2. We conduct two experiments to evaluate our framework. In the first experiment we consider as target only those datasets which GeoNames has at least one `owl:sameAs` property, while in the second experiment we randomly select and add in the experimental data triples from these datasets, such that there are no *owl:sameAs* links between them and GeoNames.

a) GeoNames with other datasets where at least one `owl:sameAs` link exist: In the first experiment we evaluate the framework for finding similar instances between GeoNames and all the other datasets where at least one `owl:sameAs` link exists. In the Gold Standard there are 1,798 *owl:sameAs* instances between these datasets. Our framework generates 1,333 links as True Positive, 127 links as False Positive and 465 links as True Negative. In terms of precision, recall and F- measure the framework returns the following results: Precision (P) = 0.91, Recall (R) = 0.74 and F-measure (F) = 0.82.

b) GeoNames with other datasets where at least one or no `owl:sameAs` link exists: In the second experiment we evaluate the framework to find similar links between GeoNames with all the others datasets adding some noise in the experimental data. The noise consist of triples from 13 different datasets. We added triples from the datasets from the first experiment and also triples from three other datasets (ordnancesurvey.co.uk; fao.org and ucd.ie), where no `owl:sameAs` links exist in the Gold Standard. We add these triples to evaluate if our framework would be able to find similar links between GeoNames dataset and the triples considered to be noise. Our framework generate 1,333 links as True Positive and 277 as False Positive. Table I shows

[7] nytimes.com, europa.eu, geovocab.org, linkedmdb.org, didactalia.net, linkedgeodata.org, lexvo.org, dbpedia.org, 270a.info, lenka.no

TABLE I
DISTRIBUTION OF SAMEAS LINKS IN GOLD STANDARD AND FRAMEWORK RESULTS

Dataset	GS	TP	FP	TP*	FP*
nytimes.com	497	460	4	462	2
europa.eu	719	679	97	770	6
geovocab.org	16	0	91	65	25
linkedmdb.org	11	9	0	9	0
didactalia.net	10	0	0	0	0
linkedgeodata.org	45	31	18	46	3
lexvo.org	97	18	1	19	0
dbpedia.org	227	127	23	131	19
270a.info	175	9	0	9	0
lenka.no	1	0	1	1	0
ordnancesurvey.co.uk	0	0	30	14	16
fao.org	0	0	10	10	0
ucd.ie	0	0	2	2	0

Gold Standard (GS), True Positive (TP), False Positive (FP), Verified True Positive (TP*), Verified False Positive (FP*).

the distribution of the links generated by the framework for each dataset. In terms of precision, recall and F- measure the framework returns the following results: Precision (P) = 0.81, Recall (R) = 0.74 and F-measure (F) = 0.77. As an observation, in the second experiment the performance of our framework decreases as a result of an increasing number of False Positive.

C. Discussion

In the following we will analyse in more detail the results from our framework, focusing in the False Positive. In order to evaluate the performance of our approach we manually check if the links generated as False Positive were correct or not. As a checking result, from 127 as False Positive from the first experiment, 99 were correct and 28 were incorrect mappings, meaning that the total number of True Positive is 1,432 and the number of False Positive is 28. Manually checking from 277 False Positive mappings from the second experiment, 206 links were correct so the number of real True Positive found by the framework is 1,539, while the number of real False Positive is 71. From this verification we prove that our framework could find 14 similar links between GeoNames and ordnancesurvey.co.uk, among which there are no links in the LOD cloud, thus improving the linkage information. We found that the resource e.g *gn:2110425/nauru.html* should be linked to *gv:0–170*[8] as both refer to the island of Nauru and *gn:2652355/cornwall.html* should be linked to *ords:7000000000043750*[9] as both refer to the county of Cornwell in England. This information currently is missing in the LOD cloud. While if we check for two resources classified as similar *gn:6324733/st_john_s.html* from GeoNames dataset and *ords:7000000000019514* from OrdnanceSurvey we see that these two recources refer to different places eventhough they share the same name. In the information that we have in the cloud, these two resources share three properties for the name (LabelLike group) and one of the coordinates is similar. These four properties contribute to the similarity score categorising these two resources as similar. Another misclassification is between *gn:2618425/copenhagen.html* and *dbpedia:Copenhagen_Municipality*. Because these resources share many properties our framework classifies them as similar. While we observe some true classification errors, many of the mistakes made by our framework point to fact that many resources are described with similar properties so it is difficult also for humans without prior knowledge to classify them as similar. Our framework can be used also to check the quality of URIs in a dataset. In the dataset ordnancesurvey.co.uk, the resource *Isle of Wight* is described with two URIs, *ords:7000000000025469* and *ords:7000000000025195*. Also in LinkedGeoData we find that the resource for the city of Vienna has two different URIs, *lgdo:node240034347* and *lgdo:node17328659*.

IV. RELATED WORKS

As we mentioned in the introduction we focus on data linking at instance level and thus in the related work we present only those tools or techniques that are close to our approach.

Similarity is usually performed on string bases. Similar to [9], we adopt the Edit Distance (Levenshtein) similarity function and the numerical similarity. Often semi-automated approaches, which must be preconfigured by the user may select from a wide range of similarity functions those suitable for the task at hand such as Silk [9]. The Silk system [9] assumes a supervised matching scenario where the user specifies entities to link in a configuration file and selects an aggregation approach (weighted average, max(min), Euclidean distance, or weighted product) for her task. Similar to Silk, the LIMES system [18] is a semi-automated approach that needs a configuration file to be setted up. In contrast, our approach implements an automated workflow which can be applied to a wide range of domains and is considered totally unsupervised. Our approach that is considered complementary to Silk and LIMES, take as input not only two datasets but one against all datasets in the LOD cloud.

LINDA [19] is a system used to compute the similarity between two entities based on their neighbours. Two kinds of similarities are computed; apriori similarity and contextual similarity. Apriori similarity is based on literals and constraints and contextual similarity is computed in each iteration and considers the current state of similarity matrix. In contrast from our approach LINDA assumes each dataset to be already disambiguated while in our approach we do not make such an assumption thus addressing a more widely application.

A statistical and qualitative analysis of instance level equivalence in the LOD cloud to automatically compute alignments at the conceptual level could be found in [6]. Adopting classical Jaccard methods to the ontology alignment task allow to improve the level of integration between datasets as this will help to resolve semantic heterogeneity. The authors used the Jaccard coefficient to measure the similarity between two concepts when interpreted as sets of instances. They considered DBpedia as the source datasets and 6 target datasets, and extracted the **sameAs** links and also the concepts hierarchy where the behaviour of classical Jaccard similarity

[8] gv : http://gadm.geovocab.org/id/

[9] http://data.ordnancesurvey.co.uk/doc/

measure was analysed by studying the influence of hierarchical information in producing the alignments.

Authors in [13], introduced an approach to automatically detect redundant identifiers solely by matching the URIs of information resources. They used two techniques to match URIs. The first is to tokenize the URI in all special characters and calculate the cosine similarity of all TF-IDF vectors and the second technique is to use exact string matching techniques after dividing the URI into prefix, infix and suffix to detect duplicates. Their approach is limited only for string similarity and do not cover cases when the URI contains numerical information and blank nodes. In contrast our approach covers both cases.

The LiQuate framework [11] combines Bayesian Networks and rule-based systems to analyze the quality of data and links in the LOD cloud. The Bayesian Networks models dependencies among resources, while queries among these models, represent the probability that different resources have redundant labels or that a link between two resources is missing while a probabilistic rule-based system is used to infer new links that associate equivalent resources. LiQuate framework can be used to suggest ambiguities or possible incompleteness in the data or links and to resolve the ambiguities and incompleteness identified during the exploration of the Bayesian Network. The LiQuate framework deals with two incompleteness problems; link incompleteness and ambiguities between labels of resources and between sets of links. In difference with our approach, Liquate is a semi automatic approach for which the last update was in 2013.

Authors in [12], have proposed a framework for iterative blocking where the entity resolution results of blocks are reflected to other blocks, in order to generate new record matches. Our approach is orthoganal with the proposed one, as the former can be applied to any system, while the latter can be used with any core ER algorithm that processes a block of records.

V. Conclusion

In this paper we proposed a framework which can automatically find similar instances in the LOD cloud without a prior knowledge about the type they belong to and the properties they share. The results show that this framework is very useful to find similar pairs between datasets not only in the same category but also with other datasets despite the category they belong to.

The analysis of the limitations of our framework, i.e., the cases where the similar pairs found were wrong, point to the current information in LOD, where usually instances even though describing different things, their property values are similar. As a future work we plan to run the framework in the whole LOD cloud, considering not only the instances connected by the `owl:sameAs` property but all the instances and also verify the True Negative links generated to verify for quality problems between the instances already connected with the `owl:sameAs` property in the LOD cloud.

Acknowledgment

The paper have been supported in part by National Natural Science Foundation of China under contract number 71171132, and 61373030.

References

[1] G. Jeh and J. Widom. *SimRank: a measure of structural-context similarity.* In Proceedings of the eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2002): 538-543.

[2] Jerome Euzenat and Pavel Shvaiko, *Ontology Matching, Second Edition*, Springer, 2013

[3] Alexander Maedche, and Steffen Staab. ”Measuring similarity between ontologies.” Knowledge engineering and knowledge management: Ontologies and the semantic web. Springer Berlin Heidelberg, 2002. 251-263.

[4] Tim Berners-Lee, J. Hollenbach, Kanghao Lu, J. Presbrey, Eric Prud'hommeaux and Monica M. C. Schraefel, *Tabulator Redux: Browsing and Writing Linked Data*, Proceedings of the WWW2008 Workshop on Linked Data on the Web, LDOW, Beijing, China, April 22, 2008.

[5] Lifang Gu, Rohan Baxter, Deanne Vickers, and Chris Rainsford *Record linkage: Current practice and future directions*. CSIRO Mathematical and Information Sciences Technical Report, Volume 3, Pages 83, 2003.

[6] Gianluca Correndo, Antonio Penta, Nicholas Gibbins and Nigel Shadbolt *Statistical Analysis of the owl: sameAs Network for Aligning Concepts in the Linking Open Data Cloud*, 23rd International Conference, DEXA 2012, Vienna, Austria, Proceedings, Part II, pages 215 - 230.

[7] William W. Cohen, Pradeep D. Ravikumar and Stephen E. Fienberg, *A Comparison of String Distance Metrics for Name-Matching Tasks*, Proceedings of IJCAI-03 Workshop on Information Integration on the Web (IIWeb-03), August 9-10, 2003, Acapulco, Mexico, pages 73–78.

[8] Harry Halpin, Patrick J. Hayes, James P. McCusker, Deborah L. McGuinness and Henry S. Thompson, *When owl: sameAs Isn't the Same: An Analysis of Identity in Linked Data*, 9th International Semantic Web Conference, China, November 7-11, 2010, Part I, pages 305-320.

[9] Volz, Julius, Christian Bizer, Martin Gaedke, and Georgi Kobilarov. *Silk-A Link Discovery Framework for the Web of Data.* LDOW 2009.

[10] Li Ding, Joshua Shinavier, Zhenning Shangguan and Deborah L. McGuinness, *SameAs Networks and Beyond: Analyzing Deployment Status and Implications of owl: sameAs in Linked Data*, 9th International Semantic Web Conference, China, November 7-11, 2010, pages 145-160.

[11] Edna Ruckhaus, Maria-Esther Vidal, Simón Castillo, Oscar Burguillos and Oriana Baldizan, *Analyzing Linked Data Quality with LiQuate*, The Semantic Web: ESWC 2014 Satellite Events - Anissaras, Crete, Greece, May 25-29, 2014, pages 488–493.

[12] Steven Euijong Whang and David Menestrina, Georgia Koutrika, Theobald Martin, and Hector Garcia-Molina, *Entity resolution with iterative blocking*, Proceedings of the 2009 ACM SIGMOD International Conference on Management of data, pages 219–232.

[13] George Papadakis, Gianluca Demartini, Peter Fankhauser and Philipp Kärger, *The missing links: discovering hidden same-as links among a billion of triples*, iiWAS'2010, 8-10 November 2010, Paris, France, pages 453–460.

[14] Max Schmachtenberg, Christian Bizer and Heiko Paulheim, *Adoption of the Linked Data Best Practices in Different Topical Domains*, The Semantic Web - ISWC 2014 - 13th International Semantic Web Conference, Riva del Garda, Italy, October 19-23, 2014, pages 245–260.

[15] Sean Bechhofer, Van Harmelen Frank, Jim Hendler, Ian Horrocks, Deborah L. McGuinness, Peter F. Patel-Schneider and Lynn Andrea Stein, W3C Recommendation, World Wide Web Consortium

[16] Alfio Ferrara, Andriy Nikolov and François Scharffe, *Data Linking for the Semantic Web*, International Journal in Semantic Web Inf. Syst., volume 7, number 3, pages 46–76, 2011.

[17] Enayat Rajabi, Miguel-Ángel Sicilia and Salvador Sánchez Alonso, *An empirical study on the evaluation of interlinking tools on the Web of Data*, J. of Information Science, volume 40, number 5, pages 637–648, 2014.

[18] Axel-Cyrille Ngonga Ngomo and Sören Auer, *LIMES - A Time-Efficient Approach for Large-Scale Link Discovery on the Web of Data*, IJCAI 2011, Barcelona, Catalonia, Spain, July 16-22, 2011, pages 2312–2317.

[19] Christoph Böhm, Gerard de Melo, Felix Naumann and Gerhard Weikum *LINDA: distributed web-of-data-scale entity matching*, Proceedings of the 21st ACM international conference on Information and knowledge management, pages 2104–2108, 2012.