N2R-Part: Identity Link Discovery using Partially Aligned Ontologies *

Nathalie Pernelle
LRI, Paris Sud University
bat. 650, Orsay, France
nathalie.pernelle@lri.fr

Fatiha Saïs
LRI, Paris Sud University
bat. 650, Orsay, France
Fatiha.sais@lri.fr

Brigitte Safar
LRI, Paris Sud University
bat. 650, Orsay, France
brigitte.safar@lri.fr

Maria Koutraki
ETIS, Cergy-Pontoise Univ.
6 av. du Ponceau, Cergy, FR
maria.koutraki@ensea.fr

Tushar Ghosh
LRI & INRIA Saclay
bat. 650, Orsay, France
Tushar.Ghosh@inria.fr

ABSTRACT

Thanks to the initiative of Linked Open Data, the RDF datasets that are published on the Web are more and more numerous. One active research field currently concerns the problem of finding links between entities. We focus in this paper on ontology-based data linking approaches which use linking rules based on the available schemas (or ontologies). This kind of systems assume to have beforehand a set of mappings between ontology elements. However, this set of mappings could be incomplete. We propose in this paper a data linking approach called N2R-Part. It is based on the computation of similarity scores by exploiting at the same time properties for which a mapping exists and those for which there is no mapping. We illustrate throughout an example how the exploitation of the unmapped properties improves the data linking results.

Keywords

Semantic Web, Ontology Alignment, Data Linkage, RDF/OWL

1. INTRODUCTION

In the Web of Data, the RDF identity links allow applications to navigate between data sources and to discover additional data describing the same real world object. When data sources provide information about large numbers of entities, these links cannot be found manually and (semi)-automated approaches are needed to generate them. Various approaches has been developed to this end [2]. Some of these approaches are instance-based and can be employed in distributed environments without having to replicate data sets locally [10, 4] while other approaches are graph-based and need to replicate data in order to generate and propagate links in the dataset [6].

Most of the linking data approaches are based on the computation of similarity scores between entities. They can exploit a shared part of the schema or semantic mappings that are declared between schema elements. These mappings can be declared manually. When ontologies are available and huge, the mappings can be proposed by an ontology alignment tool [7]. Mappings between properties are often more difficult to discover than mappings between classes. Nevertheless, we think that even if they have not been mapped, properties can be used to improve the results of a data linking tool. Indeed, there exist approaches that deal with named entity recognition in text which aim to link extracted entities to entities described in a knowledge base (a populated ontology). In this context, the properties of the extracted entities are not available, since data are not structured. However, an approach such [8] has shown that extracted entities can be successfully linked to knowledge base entities when the named entities appearing in the textual context of the extracted entities are exploited. In this paper, we extend a graph-based data linking tool named N2R [6] in order to take into account both mapped and unmapped properties. We propose a measure to estimate the similarity of two entities, based on unmapped but comparable properties.We have defined how the proposed similarity measure can be combined to a similarity that exploits only mapped properties.

We first present the data linking problem when data conform to distinct but partially aligned ontologies. Then, we present N2R-Part approach. This approach will be illustrated throughout an example. Finally, we will conclude and give some future work.

2. DATA LINKING IN PARTIALLY MAPPED ONTOLOGIES

Let s_1 and s_2 be two data sources that conform to two OWL ontologies O_1 and O_2. We consider that an ontology O_i is

defined by the tuple (C_i, H_i, P_i, Ax_i) where:
- C_i is the set of classes of O_i,
- H_i is the set of subsumption relations between classes C_i,
- P_i is the set of properties that are partitioned into two sets: Po_i is the set of relations that are defined between the classes of O_i and Pd_i is the set of datatype properties describing the classes,
- Ax_i is the set of ontology axioms, e.g., domain and range definition, keys.

Let A be the results of a mapping process that is performed on O_1 and O_2. We denote A_C and A_P the sets of mappings between classes and between properties, respectively. We assume that the data sources are already saturated using the OWL entailment rules [5].
In order to infer links between entities we compute first a similarity score $sim(i_1, i_2)$ for each pair of instances such that i_1 is an instance of $c_1 \in C_1$, i_2 is an instance of $c_2 \in C_2$ and c_1 is *comparable* to c_2, (e.g., $c_1 \subseteq c2$ or $c_2 \subseteq c_1$).

N2R [6] is a numerical approach that allows to infer identity links between pairs of instances that are described according to the same ontology or to two different ontologies for which a complete set of mappings is already computed. N2R is based on a set of non linear equations to express the influences between similarities. To distinguish the different impacts of the properties on the similarity of the instance pairs, N2R exploits the semantics of keys that are declared and identified as common in the ontologies to infer identity links between class instances. Thus, if the property *hasAsCapital* is a key for the class *Country*, a strong similarity of two city instances that are capital is propagated to the country instances, these two cities belong to. The obtained equation system is solved thanks to an iterative method based on Jacobi. The instance pairs for which the similarity is greater than a fixed threshold are linked. A such data linking approach exploits only the properties that are mapped. If the set of mappings is incomplete, the approach looses information and do not exploit all the information that is available on entities. Therefore, it may miss identity links or compute erroneous links, in particular when information is incomplete and heterogenous. This is the reason why we have extended the N2R approach to be able to exploit, in addition to the mapped properties the unmapped ones.

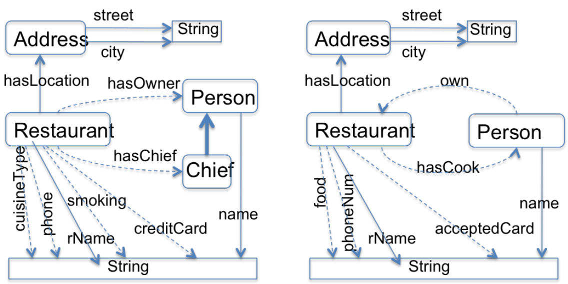

Figure 1: The two ontologies O_1 and O_2

In Figure 1, we show an example of two ontologies O_1 and O_2 that will be used to illustrate our approach. We assume that we have only the following set of equivalence mappings between the properties of O_1 and O_2:

A_P={*(hasLocation = hasLocation), (rName = rName), (street = street), (city = city), (name = name)*}.

In O_1, we consider the following keys: for the class *Restaurant*, the key is {*phone*}, for *Address* the keys are {*street, city*} and {inverse(*hasLocation*)}, and for *Person* the key is {inverse(*hasOwner*)}. This last expresses intuitively that "if two restaurants are the same then they have the same owner".

In case of data sources that conform to two different ontologies O_1 and O_2, with distinct sets of keys, the considered keys are only the common keys. We first select the keys for which there is an equivalence mapping for each of its properties. Then, the keys of O_1 and O_2 that are considered as common are computed by selecting the minimal keys of the Cartesian product of the keys of O_1 and O_2. For example, assume that the keys declared in O_2 are the following: for the class *Restaurant*, the key set is {phoneNum}, for the class *Address*, the keys are {street} and {inverse(*hasLocation*)}. Then the keys for which it exits equivalence mappings are: in O_1 {*street, city*},{inverse(*hasLocation*)} and in O_2 {*street*}, and {inverse(*hasLocation*)} since the properties *hasOwner* and *phone* do not belong to the mapping set. The Cartesian product leads to the common keys for the class *Address*: {*street, city*} and {inverse(*hasLocation*)}. Keys cannot be found for the other classes because of the incompleteness of mappings.

3. N2R-PART APPROACH

We first define the notion of comparable properties. Then we present the computation of similarity score of two instances by exploiting unmapped properties. Finally, we will show how the data linking tool N2R is extended to take into account this similarity score.

3.1 Comparable Properties

If a property of one of the two ontologies has not been mapped, we exploit the semantics of its domain and range to find comparable properties in the other ontology. More precisely, two properties are said comparable if one of their domains and one of their ranges are equivalent or more specific.

A relation $r_1 \in P_{o1}$ is comparable to another relation $r_2 \in P_{o2}$ if: $\exists C_{d1}$, $\exists C_{d2}$, $\exists C_{r1}$, $\exists C_{r2}$ such that Domain(r_1, C_{d1}), Domain(r_2, C_{d2}), Range(r_1, C_{r1}), Range(r_2, C_{r2}) and
(1) ($C_{d1} \subseteq C_{d2}$ or $C_{d2} \subseteq C_{d1}$) and ($C_{r1} \subseteq C_{r2}$ or $C_{r2} \subseteq C_{r1}$)
or
(2) ($C_{d1} \subseteq C_{r2}$ or $C_{r2} \subseteq C_{d1}$) and ($C_{r1} \subseteq C_{d2}$ or $C_{d2} \subseteq C_{r1}$)

The part (2) of the above definition allows to take into account the case where a property of O_1 has been defined as an inverse of a property in O_2. As an example, the relations *hasOwner* and *own* of the Figure 1 where *hasOwner* and *hasChief* are both comparable at the same time to *inverse(own)* and to *hasCook*.

Datatype properties are defined as comparable in an analogous way but limited to the point (1). The hierarchy of data types defined in XML Schema[1] is exploited. Thus, the four datatype properties of the class *Restaurant* of O_1 that have not been mapped and that have as range the data

[1] http://www.w3.org/TR/xmlschema-2/

type xsd : $string$, {*cuisineType, phone, creditCard, smoking*} are comparable to the three datatype properties of the class *Restaurant* of O_2 having the same range {*food, phoneNum, acceptedCard*}.

3.2 Similarity of two Instances using Comparable Properties

At each iteration, for each pair of instances (i, j), the values of each unmapped property P_k that describes i, (whatever i appears in the *domain* or in the *range* of P_k) are compared to each value of comparable (inverse) property P_l that describes j. This is performed, in order to identify the best comparable properties, denoted $BestP_l$, that have a strong similarity with the values of P_k. Then, the similarity scores of the values of the properties of $BestP_l$ are then aggregated to compute the similarity $Sim_{NAP}(i, j)$ based on the set of unmapped properties.

We assume that a similarity measure sim is chosen and declared for each *xsd:dataType*: a measure to compare string values, one to compare integers, another to compare dates and so on. The comparaison of two datatype properties P_k and P_l is achieved in several steps:

(i) we compute the similarity between each value v_{Pk1}, ..., v_{Pkn} that describe the instance i through the property P_k, and the values v_{Pl1}, ..., v_{Plm} that describe the instance j through the property P_l using the similarity measure sim. For each value v_{Pkr} (with $r \in [1..n]$) we keep the best similarity max_{sim} with one of the values of P_l, $max_{sim}(v_{Pkr})$ = $Max_{s\in[1..m]}$ (sim(v_{Pkr}, v_{Pls})), if the similarity is greater than a fixed threshold θ.

(ii) the result of the comparison of the two datatype properties is expressed using a vector $(i, j, P_k, P_l, S_{Sim}, Nb_{VP})$, where: i and j are the two instances to be compared, S_{Sim} is the sum of the values of max_{sim} of v_{Pkr} and Nb_{VP} is the maximum number of the instances of the properties P_k and P_l, i.e., $Nb_{VP} = Max(n, m)$. This vector is built only if $S_{Sim} > 0$.

The similarity function Sim_{NAP} expresses the similarity of Not Aligned Properties (NAP). It corresponds to the aggregation of the similarity scores of all the best comparable properties, by taking into account the number of instances of similar properties nb_{VP}:

$$Sim_{NAP}(i, j) = \frac{\sum_{BestP_l} S_{Sim}}{\sum_{BestP_l} Nb_{VP}}$$

Example 1. Let (i_1, i_2) be a pair of instances of *Restaurant*. We consider the set of datatype properties that are unmapped and comparable, have the following values:

(i_1, *cuisineType*, "asian"),	(i_2, *food*, "asian")
(i_1, *cuisineType*, "thai"),	(i_2, *food*, "chinese")
(i_1, *phone*, "33 68 55 51 58"),	(i_2, *food*, "thai"),
(i_1, *phone*, "33 88 82 60 36"),	(i_2, *phoneNum*, "33 68 55 51 58"),
(i_1, *creditCard*, "visaCard"),	(i_2, *acceptedCard*, "MasterCard")
(i_1, *smoking*, "only at bar")	(i_2, *acceptedCard*, "visaCard"),

For sake of simplicity, we assume that the similarity sim is the equality of string values. In order to identify the $BestP_l$ to assign to *phone*, for the instance pair (i_1,i_2), we should :

(1) compute the similarity of each value of v_{phone} = {3368555158, 3388826036} with the values that describe the different comparable datatype properties of i_2. We obtain for these two values a $max_{sim} = 0$ where we compare them to v_{food} and to $v_{acceptedCard}$. We obtain also $max_{sim}(3368555158) = 1$ and $max_{sim}(3388826036) = 0$ where the v_{phone} values are compared to $v_{phoneNum}$.

(2) the only datatype property that has a $S_{Sim} > 0$ is *phoneNum*. It is retained as $BestP_l$ to *phone*, with the vector (i_1,i_2, *phone*, *phoneNum*, S_{Sim} = 1, nb_{VP} = 2).

We also obtain the following best comparable properties:
(i_1, i_2, *cuisineType*, *food*, S_{Sim} = 2, Nb_{VP} = 3),
(i_1, i_2, *creditCard*, *acceptedCard*, S_{Sim} = 1, Nb_{VP} = 2).
The property *smoking* has no $BestP_l$ and will then not be considered in similarity computation. $Sim_{NAP}(i_1, i_2) = \frac{1+2+1}{2+3+2} = \frac{4}{7}$. The similarity score of the values of unmapped object properties (instances) is computed using an analogous way. The only particularity is that sim of two instances evolve as propagations are performed in N2R-Part.

3.3 Similarity Combination

In N2R, the similarity score of an instance pair (i_1, i_2) is represented by a variable x_i with $i \in [1..n]$ and n is the number of instance pairs for which N2R is performed. $X = (x_1, x_2, \ldots, x_n)$ is the set of variables that correspond to each instance pair. The similarity scores between literals are expressed using constants obtained thanks to similarity measures (e.g. Levenstein, Jaro-Winckler, ...). In the equation system, $x_i = f_i(X)$ expresses the fact that the value x_i depends on the similarities of the other instance pairs. Each equation is in the form:

$$f_i(X) = max(f_{i_{Key}}(X), f_{i_{NKey}}(X))$$

The function $f_{i_{Key}}(X)$ returns the maximum similarity score obtained for the properties that are involved in keys. Thus, allows to boost up the propagation of a high similarity score for the datatype properties or the object properties that are involved in keys to other instance pairs. The function $f_{i_{NKey}}(X)$ is a weighted average of the similarity scores of literals or instances that are not involved in a key (see [6] for a detailed presentation of $f_i(X)$).

The proposal here consists in aggregating the similarity score obtained by N2R and the one obtained on the unmapped properties. This aggregation should allow to:

- ensure that a high similarity score of the values of mapped properties, that are involved in keys, leads to a high similarity score of the instances that are described using these properties. For this reason, we keep the solution of using a maximum function between these similarity scores ($f_{i_{Key}}(X)$) and the other ones.

- give a bigger importance to similarity scores of the mapped property values in comparison to the unmapped property values (use of a weight $\alpha \in [0..1]$).

Each equation $x_i = f_i(X)$ becomes:
$f_i(X) = max(f_{i_{Key}}(X), f_{i_{AllMap}}(X) + \alpha \times f_{i_{NAP}}(X))$

with the function $f_{i_{AllMap}}(X)$ is a weighted average of all the similarity scores of the mapped property values and $f_{i_{NAP}}(X)$ is the similarity score of the others.

These two functions should take into account the number of properties (nb_P) that exist in the different schemas and that are likely to be compared. To do so, we consider c_1

(resp. c_2) the most specific class instantiated by i_1 (resp. i_2) and n_1 (resp. n_2) the number of (inherited) properties that describe c_1 (resp. c_2). This property number nb_P is: $min(n_1, n_2)$. In our example, an instance of person is described by two properties (*hasOwner* and *name*) in O_1 and three properties (*own*, *hasCook* and *name*) in O_2. Then, for two person instances, nb_P is 2. For restaurants, nb_P is 7, and for addresses, it is 3.

Example 2. Let s_1 and s_2 be two data sources that contain the following descriptions in addition to the ones given in Example 1.

(i_1, *hasLocation*, a_1)	(i_2, *localisation*, a_2)
(a_1, *street*, "17 rue polar")	
(a_1, *city*, "Paris")	(a_2, *ville*, "Paris")
(i_1, *rName*, "le lotus bleu")	(i_2, *rName*, "le lotus bleu")
(i_1, *hasOwner*, p_1)	(p_2, *own*, i_2)
(p_1, *name*, "Chang Lee")	(p_2, *name*, "Chang Lee")

The three variables x_A, x_R, x_P represent, respectively, the similarity scores of the instance pairs of Address (a_1, a_2), of Restaurant (i_1, i_2) and of Person(p_1, p_2). They are initialized to 0 and change at each iteration in function of the variable values that appear in their equations.

The similarity scores of literals are expressed using constants. Thus, the constants a, b and c, are all equal to 1, and represent respectively the similarity score of person names sim("Chang Lee", "Chang Lee"), the one of cities, sim("Paris", "Paris") and the one of restaurant names, sim("le lotus bleu", "le lotus bleu"). The constant $d = 4/7$, represents the similarity of unmapped datatype properties computed above.

Using the weight $\alpha = \frac{4}{5}$, the similarity influences between the three variables x_A, x_R, x_P are expressed by the following equations:

$$x_A = \max(\ x_R,\ \tfrac{1}{3}\ b + \tfrac{1}{3}\ x_R),\ x_P = \tfrac{1}{2}\ a + \tfrac{4}{5}\ (\tfrac{1}{2}\ x_R)$$
$$x_R = \tfrac{1}{7}\ c + \tfrac{1}{7}\ x_A + \tfrac{4}{5}\ (\tfrac{3}{7}\ d + \tfrac{1}{7}\ x_P),$$

The similarity x_A of addresses (a_1, a_2) is the maximum value of: (i) restaurant similarity score x_R (the common key *hasLocation*) and (ii) the weighted similarity score of the litterals or instances that instantiate mapped properties (the cities b and the restaurants x_R). The weight ($\frac{1}{3}$) corresponds to $\frac{1}{nb_P}$. Note that, this equation does not involve unmapped properties.

For instance, the similarity x_R of restaurants (i_1, i_2) is the aggregation of: (i) the similarity score of the mapped datatype property $rName$ and the mapped object property *hasLocation* and (ii) the similarity score of the unmapped datatype properties computed above and the best comparable object property (*own*, *hasOwner*).

The table 1 presents the similarity values of the variables after five iterations, (a fix-point at 0.001), depending on whether we use N2R-Part on the mapped object properties only (without NAP) or on the mapped and unmapped object properties. Without NAP, the system uses only four properties and one key (since a part of the address description is not given for a_2). Furthermore, the similarity propagation is not possible between the restaurants and the persons (because the properties *own* and *hasOwner* are not mapped). With NAP, we exploit four additional properties and we allow some similarity propagations, as between restaurants x_R and persons x_P. By construction, the way unmapped properties are taken into account can only increase the similarity scores of the ones obtained without exploiting unmapped properties. In this first definition of the approach, all the mapped properties are considered at the same level of importance. Nevertheless, similarity computations are time consuming and not always relevant. For instance the accepted credit cards are not relevant to be taken into account when comparing restaurants.

Variables	x_R	x_A	x_P
without NAP	0.199	0.399	0.5
with NAP	0.489	0.496	0.695

Table 1: Tests on instances of Example 2

4. CONCLUSION AND FUTURE WORK

In this paper, we have shown how an existing data linking tool can be extended to take into account unmapped properties. This approach allows to augment the considered information when comparing to entities. The proposed approach generates more candidate links between entities, since similarity scores can only increase. Indeed, it can be too restrictive to assume that all the relevant links are found by using only keys and mapped properties, especially, when data are incomplete and heterogenous.

We now plan to test the approach on real datasets of different domains. Furthermore, this approach can be refined to select only unmapped properties that are highly discriminative: keys or (inverse) functional properties that have been declared in the ontologies or discriminative sets of properties automatically discovered in the RDF dataset [9]. Finally, we aim to study how the results of a such approach can be exploited to learn new possible mappings between properties.

5. REFERENCES

[1] A. Ferrara, A. Nikolov, and F. Scharffe. Data linking for the semantic web. *Int. J. Semantic Web Inf. Syst.*, 7(3):46–76, 2011.

[2] O. Hassanzadeh, A. Kementsietsidis, L. Lim, R. J. Miller, and M. Wang. A framework for semantic link discovery over relational data. In *Proceedings of ACM CIKM'2009*, pages 1027–1036. ACM, 2009.

[3] P. F. Patel-Schneider, P. Hayes, and I. Horrocks. OWL Web Ontology Language Semantics and Abstract Syntax Section 5. RDF-Compatible Model-Theoretic Semantics. Technical report, W3C, Dec. 2004.

[4] F. Saïs, N. Pernelle, and M.-C. Rousset. Combining a logical and a numerical method for data reconciliation. *J. on Data Semantics*, 12:66–94, 2009.

[5] P. Shvaiko and J. Euzenat. Ontology matching: State of the art and future challenges. *IEEE Trans. Knowl. Data Eng.*, 25(1):158–176, 2013.

[6] F. Suchanek, M. Sozio, and G. Weikum. Sofie: A self-organizing framework for information extraction. In *WWW conference*, pages 631– 640, 2009.

[7] D. Symeonidou, N. Pernelle, and F. Saïs. Kd2r: A key discovery method for semantic reference reconciliation. In *OTM Workshops SWWS*, pages 392–401, 2011.

[8] J. Volz, C. Bizer, M. Gaedke, and G. Kobilarov. Discovering and maintaining links on the web of data. In *Proceedings of ISWC '09*, pages 650–665, 2009.