LDM: Link Discovery Method for new Resource Integration

Nathalie Pernelle, Fatiha Sais

LDM: Link Discovery Method for new Resource Integration

Nathalie Pernelle[1] and Fatiha Saïs[1]

LRI(CNRS UMR 8623 & Paris-Sud 11 University), INRIA Saclay,
4 rue Jacques Monod, Parc Club Orsay Université, F-91893 Orsay Cedex, France
{Nathalie.Pernelle, Fatiha.Sais}@lri.fr

Abstract. In this paper we address the problem of resource discovery in the Linked Open Data cloud (LOD) where data described by different schemas is not always linked. We propose an approach that allows discovery of new links between data. These links can help to match schemas that are conceptually relevant with respect to a given application domain. Furthermore, these links can be exploited during the querying process in order to combine data coming from different sources. In this approach we exploit the semantic knowledge declared in different schemas in order to model: (i) the influences between concept similarities, (ii) the influences between data similarities, and (iii) the influences between data and concept similarities. The similarity scores are computed by an iterative resolution of two non linear equation systems that express the concept similarity computation and the data similarity computation. The proposed approach is illustrated on scientific publication data.

1 Introduction

The appearance of Web of documents (WWW) [1] has upset the way we create and share knowledge by breaking down barriers of publishing and accessing documents. Hypertext links allow users to navigate on the graph of documents and Web search engines to index the documents and answer to user queries. However, hyperlinks do not express explicit links between the various entities described in Web of documents. With the initiative of Open Linked Data cloud [3], the number of data providers on the Web is in a continuous growth leading to a global data space of billions of assertions where data and documents can be linked. However, until now the published data is very heterogeneous in the sense that it is incomplete, inconsistent, described according to different schemas and contains duplicates. In order to be able to automatically exploit this huge amount of heterogeneous data, an important work integration must be performed.

In this paper we focus our interest on the problem of resource discovery in the Linked Open Data cloud (LOD) where data described by different schemas is not always linked. We propose an approach[1] that allows discovery of new links between data. These links can help to match schemas that are conceptually relevant with respect to a given application domain.

[1] in the setting of the ANR (the French National Research Agency) project GeOnto.

Ontology alignment plays a key role for semantic interoperability of this data. Many approaches have been proposed for automatically identifying mappings between elements (concepts and relations) described in heterogeneous ontologies [18, 14]. These approaches may exploit lexical and structural information, user inputs, prior matches or external resources. When concept and relation instances are available, it is also possible to exploit them to find more mappings between ontologies. In [7], the common instances of concepts are exploited to compute mappings between concepts. Since, data is not described using the same URIs even when it describes the same entities, these common instances cannot be obtained straightforwardly. Conversely, discovering that two pieces of data refer to the same world entity is also a key issue for data integration. We propose an approach which simultaneously addresses both problems of ontology alignment and data linking. Thus, the results of data linking step is exploited to improve the results of ontology alignment step and vice versa. These two steps are performed alternatively until a fix point is reached. The two methods exploit the semantic knowledge that is declared in different schemas (ontologies) in order to model: (i) the influences between concept similarities, (ii) the influences between data similarities, and (iii) the influences between data and concept similarities. The similarity scores are computed using an iterative resolution of two non linear equation systems that express, respectively, the concept similarity computation and the data similarity computation.

Applying this approach allows one to infer mappings of equivalence between concepts of different schemas as well as to infer *owl:same-as* relations between instances that refer to the same entity. The obtained schema mappings allow discovery new resources and inferring if they are relevant with respect to a given application domain.

The paper is organized as follows: in section 2 we present the related work in data linking and ontology reconciliation fields. In section 3, we present the ontology and data model and give a short presentation of N2R method on which our work relies. Section 4 presents the proposed approach of link discovery. Finally, we conclude and give some future work in section 5.

2 Related Work

We denote by "*web data*" the network formed by the set of structured datasets described in RDF (Resource Description Framework) and linked by explicit links. Large amount of structured data have been published, including in the project Linking Open Data cloud (LOD).

Datasets are expressed in terms of one or several ontologies for establishing the vocabulary describing data. Web data requires linking together the various sources of published data. Given the big amount of published data, it is necessary to provide methods for automatic data linking. Several tools [17, 13, 10] have recently been proposed to solve partially this problem, each with its own characteristics. For instance, [10] have developed a generic framework for integrating linking methods in order to help users finding the link discovery methods that are more suitable for their relational data. They introduced LinQL, an extension of SQL that integrates querying with string matching (e.g. weighted jaccard measure) and/or semantic matching (i.e. using synonyms/hyponyms) methods. This approach takes advantage of the DBMS query engine

optimizations and it can easily be used to test elementary similarity measures. Nevertheless, this approach is not designed to propagate similarity scores between entities, i.e. their approach is not global.

Some other works address the problem of link discovery in the context semantic Web services. In [11], the authors propose to match a user request with semantic web service descriptions by using a combination of similarity measures that can be learnt on a set of labeled examples.

Our proposal in this paper can also be compared to approaches studying the reference reconciliation problem, i.e., detecting whether different data descriptions refer to the same real world entity (e.g. the same person, the same paper, the same protein). Different approaches have been proposed. [5, 19, 2, 6] have developed supervised reference reconciliation methods which use supervised learning algorithm in order to learn parameters and help the duplicate detection. Such supervised approaches cannot be used in contexts where data amount is big and data schemas are different and incomplete.

In [16] we have developed an automatic method of reference reconciliation which is declarative and unsupervised reference reconciliation method. Besides, in this method we assumed that the data sets conform to the same schema, i.e. the problem of ontology reconciliation is already solved. Some ontology reconciliation approaches [7, 12] have proposed to exploit a priori reconciled instances in the ontology reconciliation process. When we aim at online reference and ontology reconciliation in the context of Linked Open Data, we cannot use these traditional reconciliation approaches, where solving the problem of reference reconciliation assumes the resolution of the ontology reconciliation and vice versa. Furthermore, up to our knowledge, there is no approach which deals with the two problems of discovering links in the ontology level and in the data level, simultaneously.

3 Preliminaries

In this section we will present the ontology and data model that we consider in this work. We will then present the Numerical method for Reference Reconciliation (N2R) [16] on which relies our link discovery approach.

3.1 Ontology and its Constraints

The considered OWL ontology consists of a set of concepts (unary relations) organized in a taxonomy and a set of typed properties (binary relations). These properties can also be organized in a taxonomy of properties. Two kinds of properties can be distinguished in OWL: the so-called relations (*owl:objectProperty*), the domain and the range of which are concepts and the so-called attributes (*owl:DatatypeProperty*), the domain of which is a concept and the range of which is a set of basic values (e.g. Integer, Date, Literal). In Figure 1, we give an extract $O1$ of the ontology that is used to describe the RDF data of the local data source of publications (see source 1 Figure 2) which we will use to illustrate our proposal.

We allow the declaration of constraints expressed in OWL-DL or in SWRL in order to enrich the domain ontology by additional and useful knowledge. The constraints that we consider are of the following types:

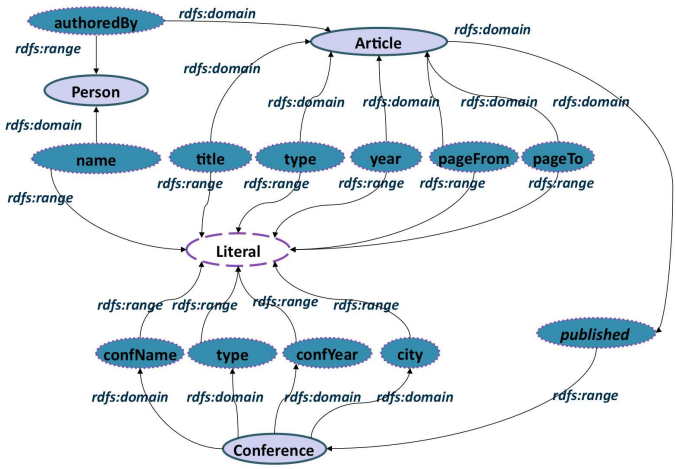

Fig. 1. An extract O1 of the local Ontology for publications

Source S1:
Article(S1_a1); title(S1_a1,“Implementing the TEA algorithm on sensors”); Person(S1_p1); Person(S1_p2); year(S1_a1, “2004”); name(S1_p1,“Olga V. Gavrylyako”); name(S1_p2,“Shuang Liu”); pageFrom(S1_a1,“64”); pageTo(S1_a1,“69”);

Conference(S1_c1); confName(S1_c1, “Proceedings of the 42nd Annual Southeast Regional Conference, 2004, Huntsville, Alabama, USA, April 2-3, 2004”); confYear(S1_c1, “2004”); city(S1_c1, “Alabama”)
authoredBy(S1_a1,S1_p1); authoredBy(S1_a1,S1_p2); published(S1_a1,S1_c1);

Article(S1_a2); title(S1_a2,“Weighted Hyper-sphere SVM for Hypertext Classification”); Person(S1_p3); Person(S1_p4); year(S1_a2, “2008”); name(S1_p3,“Shuang Liu”); name(S1_p4,“Guoyou Shi”); pageFrom(S1_a2,“733”); pageTo(S1_a2,“740”);

Conference(S1_c2); confName(S1_c2, “Advances in Neural Networks - ISNN 2008, 5th International Symposium on Neural Networks, ISNN 2008, Beijing, China, September 24-28, 2008, Proceedings, Part I”); confYear(S1_c2, “2008”) city(S1_c2, “Beijing”) authoredBy(S1_a2,S1_p3); authoredBy(S1_a2,S1_p4); published(S1_a2,S1_c2);

Source S2:
Article(S2_a1); title(S2_a1,“Implementing the TEA algorithm on sensors.’); Person(S2_p1); Person(S2_p2); year(S2_a1, “2004”); name(S2_p1,“Olga V. Gavrylyako”); name(S2_p2,“Shuang Liu”); pageFrom(S2_a1,“64”); pageTo(S2_a1,“69”);

Conference(S2_c1); confName(S2_c1, “42nd Annual Southeast Regional Conference, 2004”); confYear(S2_c1, “2004”); city(S2_c1,“Alabama”) authoredBy(S2_a1,S2_p1); authoredBy(S2_a1,S2_p2); published(S2_a1,S2_c1);

Fig. 2. an extract of RDF data

– Constraints of disjunction between concepts: DISJOINT(C,D) is used to declare that the two concepts C and D are disjoint. In the ontology O_1 we declare that all the concepts $Article$, $Conference$ and $Person$ are pairwise disjoint.

– Constraints of functionality of properties: PF(P) is used to declare that the property P (relation or attribute) is a functional property. In O_1, we declare that all the properties are functional except the relation $authoredBy$ which means that one article may have several authors.
– Constraints of inverse functionality of properties: PFI(P) is used to declare that the property P (relation or attribute) is an inverse functional property. These constraints can be generalized to a set $\{P_1, \ldots, P_n\}$ of relations or attributes to state a combined constraint of inverse functionality that we will denote PFI($P_1, \ldots, P_n$). In $O1$, we declare that the combinations ($title, year$) and ($confName$, $confYear$) are inverse functional. For example, $PFI(title, year)$ expresses that one title and one year cannot be associated to several articles (i.e. both are needed to identify an article).

3.2 Data description and its constraints.

A piece of data has a reference, which has the form of a URI (e.g. `http://dblp.l3s.de/d2r/resource/authors/A._Joe_Turner`), and a description, which is a set of RDF facts involving its reference. An RDF fact can be either: (i) a concept-fact $C(i)$, where C is a concept and i is a reference, (ii) a relation-fact $R(i1, i2)$, where R is a relation and $i1$ and $i2$ are references, or (iii) an attribute-fact $A(i, v)$, where A is an attribute, i a reference and v a basic value (e.g. integer, string, date). We consider the Unique Name Assumption (UNA) which can be declared or not on a data source. Declaring UNA on a data source means that two different data descriptions having two different references, then we infer that they refer to distinct entities.

The data description that we consider is composed of RDF facts coming from the data sources which are enriched by applying the OWL entailment rules. Figure 2, provides examples of data coming from two RDF data sources S1 and S2, which conform to the same ontology describing the scientific publication domain previously mentioned.

In the N2R method which we will present in section 3.3, we consider that the descriptions of data coming from different sources conform to the same OWL ontology (possibly after ontology reconciliation). In the link discovery method, that we will present in section 4, the assumption of prior ontology reconciliation is not fulfilled, i.e. the considered data source do not conform to the same ontology.

3.3 N2R: a Numerical method for Reference Reconciliation

N2R is a numerical method which allows inferring reconciliation decisions between reference coming from different sources that conform to the same ontology, i.e. the problem on ontology reconciliation is already solved.

N2R [16] has two main distinguishing characteristics. First, it is fully unsupervised: it does not require any training phase from manually labeled data to set up coefficients or parameters. Secondly, it is based on equations that model the influences between similarities. In the equations, each variable represents the (unknown) similarity between two references while the similarities between values of attributes are expressed by constants. These constants are obtained, either by (i) exploiting a dictionnary of synonyms (e.g.

WordNet thesaurus, the dictionnary of synonyms generated by L2R method [15]); or (ii) using standard similarity measures on strings or on sets of strings [4]. Furthermore, ontology and data knowledge (disjunctions and UNA) is exploited by N2R in a filtering step to reduce the number of reference pairs that are considered in the equation system. The functions modeling the influence between similarities are a combination of maximum and average functions in order to take into account the constraints of functionality and inverse functionality declared in the OWL ontology in an appropriate way.

N2R can also take as input a set of reference pairs that are reconciled (sim =1) by another method (e.g. L2R [15] in the LN2R approach) or given by a user like the *owl:same-as* links available in the Open Linked Data cloud.

The equations modeling the dependencies between similarities. For each pair of references, its similarity score is modeled by a variable x_i and the way it depends on other similarity scores, is modeled by an equation: $x_i = f_i(X)$, where $i \in [1..n]$ and n is the number of reference pairs for which we apply N2R, and $X = (x_1, x_2, \ldots, x_n)$ is the set of their corresponding variables. Each equation $x_i = f_i(X)$ is of the form:

$$f_i(X) = max(f_{i-df}(X), f_{i-ndf}(X))$$

The function $f_{i-df}(X)$ is the maximum of the similarity scores of the value pairs and the reference pairs of attributes and relations with which the i-th reference pair is functionally dependent. The maximum function allows propagating the similarity scores of the values and the references having a strong impact. The function $f_{i-ndf}(X)$ is defined by a weighted average of the similarity scores of the value pairs (and sets) and the reference pairs (and sets) of attributes and relations with which the i-th reference pair is not functionally dependent. See [16] for the detailed definition of $f_{i-df}(X)$ and $f_{i-ndf}(X)$.

Iterative algorithm for reference pairs similarity computation. Solving this equation system is done by an iterative method inspired from the Jacobi method [8], which is fast converging on linear equation systems. To compute the similarity scores, we have implemented an iterative resolution method. At each iteration, the method computes the variable values by using those computed in the precedent iteration. Starting from an initial vector $X^0 = (x_1^0, x_2^0, ..., x_n^0)$, the value of the vector X at the k-th iteration is obtained by the expression: $X^k = F(X^{k-1})$. At each iteration k we compute the value of each x_i^k: $x_i^k = f_i(x_1^{k-1}, x_2^{k-1}, ...x_n^{k-1})$ until a fix-point with a precision ϵ is reached. The fix-point is reached when: $\forall i, |x_i^k - x_i^{k-1}| <= \epsilon$.

In order to illustrate the iterative resolution of the equation system, we consider an extract of RDF data given in Figure 2 corresponding to the set of RDF facts where the references $S1\_a1$, $S1\_c1$, $S2\_a1$ and $S2\_c1$ are involved. By considering the disjunctions between concepts of $O1$ and the UNA in $S1$ and $S2$, we obtain an equation system of six variables:

x_1 = Sim$_r$(S1_a1, S2_a1) ; x_2 = Sim$_r$(S1_c1, S2_c1) ;
x_3 = Sim$_r$(S1_p1, S2_p1) ; x_4 = Sim$_r$(S1_p1, S2_p2) ;
x_5 = Sim$_r$(S1_p2, S2_p1) ; x_6 = Sim$_r$(S1_p2, S2_p2).

We give bellow, the similarity scores of basic values obtained by using the Jaccard similarity measure. For clarity reasons, we denote the value of an attribute A associated to a reference i as: $A.val(i)$. For example, the $confYear$ value associate to the

reference $S2\_c2$ is denoted $confYear.val(S2\_c2)$ which equals to "2008". The similarity score of the two conference names that are needed in the equation system and that belong to $]0,1[$ is:
$Sim_v(confName.val(S1\_c1), confName.val(S2\_c1)) = 0.43$. All the similarity scores of basic values, that are needed in the computation, are either equal to 1 or equal to 0.

The weights that are used in the weighted average of equations are computed in function of the number of common attributes and common relations of the reference pairs. The similarity computation is illustrated by the equation system (see Table 1) obtained from the data descriptions shown in Figure 2 which conforms to the ontology $O1$. The detailed equations expressing the similarity computation of two articles and two conferences are as follows:
$x_1 = \max(\frac{1}{2}(Sim_v(title.val(S1\_a1), title.val(S2\_a1)) + Sim_v(year.val(S1\_a1),$
$year.val(S2\_a1))), \frac{1}{6}(x_2 + SJ(\{S1\_p1, S1\_p2\}, \{S2\_p1, S2\_p2\}),$
$Sim_v(pageFrom.val(S1\_a1), pageTo.val(S2\_a1)))$
with SJ is the $SoftJaccard_o$ similarity measure between sets of objects (see section 4.2)

$x_2 = \max(x_1, \max(\frac{1}{2}(Sim_v(confName.val(S1\_c1), confName.val(S2\_c1))+$
$Sim_v(confYear.val(S1\_c1), confYear.val(S2\_c1))), \frac{1}{4}(Sim_v(city.val(S1\_c1),$
$city(S2\_c1)))$

$x_3 = \frac{1}{2} * x_1 + \frac{1}{2} * Sim_v(name.val(S1\_p1), name.val(S2\_p1))$

The equation system and the different iterations of the resulting similarity computation are provided in Table 1. We assume that fix-point precision ϵ equals to 0.005.

Iterations	**0**	**1**	**2**	**3**
$x_1 = \max(\frac{1}{2}(1+1), \frac{1}{6}(x_2 + XS_1 + 1 + 1))$	0	1	1	1
$x_2 = \max(x_1, \max(\frac{1}{2}(0.43+1), \frac{1}{4}(1)$	0	0.71	1	1
$x_3 = \frac{1}{2}(x_1 + 1)$	0	0.5	1	1
$x_4 = \frac{1}{2}(x_1 + 1)$	0	0.5	1	1
$x_5 = \frac{1}{2}(x_1 + 0)$	0	0	0.5	0.5
$x_6 = \frac{1}{2}(x_1 + 0)$	0	0	0.5	0.5

Table 1. *Example of iterative similarity computation*

The solution of the equation system is $X = (1, 1, 1, 1, 0.5, 0.5)$. This corresponds to the similarity scores of the six reference pairs. The fix-point has been reached after three iterations. If we fix the reconciliation threshold T_{rec} at 0.80, then we obtain four reconciliation decisions: two articles, two conferences and two pairs of persons.

4 Link Discovery Method (LDM)

We present in this section our LDM approach which aims to discover a LOD source that shares concepts and data with a data source described dy a domain ontology. Our

approach compares a local dataset on which domain knowledge can be declared and a LOD dataset by using a combined ontology reconciliation and reference reconciliation method. Since data that is provided by the LOD source and by the domain application source is not described using the same ontology, we have adapted N2R method in order to be able to compute data similarities when data do not belongs to non disjoint concepts but to similar concepts. Furthermore, we have defined how similarities between concepts of two ontologies can be computed when some of their references are common (i.e. same URI or *owl:same-as* links that have been previously asserted) or similar. The main steps of our link discovering approach are as follows:

1. application of an ontology mapping tool to obtain: (i) the set of equivalent/ comparable properties and (i) initial similarity scores for some concept pairs;
2. building of the two equation systems: the *conceptual equation system* which expresses the similarity computation between pairs of concepts in function of their labels, their structural similarity and their references; and the *instance level equation system* one which expresses the similarity computation between pairs of references in function of their common description and the similarity of the concepts they are instance of;
3. iterative resolution of the conceptual equation system until a fix point is reached;
4. iterative resolution of the instance level equation system until a fix point is reached.

The two steps (3) and (4) are iterated until a global fix point is reached, i.e., neither the resolution of the conceptual equation system nor the resolution of the instance level equation system does update the similarity scores.

In the following subsections, we will first describe the elementary similarity measures that are used to compute similarities. Then, we present the two equation systems that have been defined to compute concept similarities and data similarities. Finally, we illustrate our LDM approach on data and ontologies of publication domain.

4.1 Initialization

We first use an alignment tool which exploits lexical and structural information to find similarity scores between ontology elements (concepts and properties). Given a local ontology O1 and a LOD ontology O2, the used alignment tool finds a set of mappings and each mapping is described by the tuple $\{e_1, e_2, co, rel\}$ where e_1 is aligned with the confidence co to the element e_2 using the type of correspondence rel (e.g. equivalence, subsumption, overlap, closeness, etc.). These scores are used to initialize the similarity score sim_{Init} of each pair of concepts and to find a set of properties (relations or attributes) that are very similar ($rel = equivalence$ or $subsumption$, $co \geq th$ and th is a high threshold). These properties are then considered as equivalent.

4.2 Elementary similarity measures

We present in this section the elementary measures used to compute similarity scores between pairs of concepts of two ontologies. These elementary similarity measures take into account the lexical and the structural knowledge declared in the two ontologies.

Most of these elementary similarity measures are based on the *SoftJaccard* similarity measure which computes similarity between sets of basic values or between sets of objects (e.g., references, concepts).

SoftJaccard: a similarity measure for sets of objects. In [16], we have defined the *SoftJaccard* similarity measure which is an adaptation of the *Jaccard* similarity measure in the sense that: (i) instead of considering only basic values we consider sets of basic values and (ii) instead of considering the equality between values we consider a similarity score with respect to a threshold θ.

Let S_1 and S_2 be two sets of elements which can be basic values or objects. To compute the similarity score between S_1 and S_2 we compute, first, the set $CLOSE_T(S_1, S_2, \theta_k)$ which represents the set of element pairs of $S_1 \times S_2$ having a similarity score $sim_T \geq \theta$.

$CLOSE_T(S_1, S_2, \theta) = \{e_j \mid e_j \in S1 \text{ and } \exists e_k \in S_2 \text{ s.t. } Sim_T(e_j, e_k) > \theta\}$,

with T a parameter which indicates if the sets S_1 and S_2 contain basic values, then $T = v$ or contain objects, then $T = o$. When $T = v$, the function Sim_v corresponds to a similarity measure between basic values like $Jaccard$, $Jaro - Winkler$, and so on [4]. When $T = o$, the function Sim_o corresponds to a similarity score that can be provided by a tool dedicated to object comparison like N2R tool [16] for references or TaxoMap [9] tool for concepts.

$$SoftJaccard_T(S_1, S_2, \theta) = \frac{\mid CLOSE(S_1, S_2, \theta) \mid}{\mid S_1 \mid}, \; with \; \mid S_1 \mid \geq \mid S_2 \mid$$

Similarity measures used to compare concepts. To compute the similarity scores between concepts we exploit both the conceptual content which means the sets of ancestors and the sets of descendants but also the sets of shared properties with respect to a given equivalence relation. The similarity score between concepts is also function of the similarity scores of their references, i.e. instance level content.

Similarity of concept labels. In OWL ontologies sets of labels are usually associated to the concepts. In case of concepts where the labels are not given, we consider their corresponding URIs. Let L_1 be the set of labels of a concept c_1 and L_2 be the set of labels of the concept c_2. The label similarity sim_{label} is computed by applying the $SoftJaccard$ similarity measure on the two sets of basic values L_1 and L_2: $sim_{label}(c_1, c_2) = SoftJaccard_v(L_1, L_2, \theta_1)$.

Similarity of concept ancestors. For two concepts, we also compute the similarity of their ancestor sets in the two ontologies. Let A_1 be the set of ancestors of the concept c_1 and A_2 be the set of ancestors of c_2. The ancestor similarity sim_{anc} is computed by applying $SoftJaccard$ similarity measure on the two sets of concept ancestors (i.e. objects) which is defined as follows: $sim_{anc}(c_1, c_2) = SoftJaccard_o(A_1, A_2, \theta_2)$

Similarity of concept descendants. The similarity score of two concepts also depends on the similarity scores of their descendants in the two ontologies. Let D_1 be the set of descendants of the concept c_1 and D_2 be the set of descendants of c_2. The descendant similarity sim_{desc} is computed by applying $SoftJaccard$ similarity measure on the two sets of concept descendants which is defined as follows:

$sim_{desc}(c_1, c_2) = SoftJaccard_o(D_1, D_2, \theta_3)$

Similarity of shared properties of concepts. The similarity of two concepts depends on the proportion of equivalent properties compared to the full number of properties defined for both concepts. Let $R1_d$ (resp. $R2_d$) be the set of properties such that the concept c_1 (resp. c_2) is subsumed by the (equivalent) property domain and let $R1_r$ (resp. $R2_r$) the set of properties such that the c_1 (resp. c_2) is subsumed by one of the range of the (equivalent) property. The relation similarity sim_{rel} is defined as follows :

$$sim_{rel}(c_1, c_2) = \frac{\mid (R1_d \cap R2_d) \cup (R1_r \cap R2_r) \mid}{\mid (R1_d \cup R2_d \cup R1_r \cup R2_r) \mid}$$

Similarity of concept references. The similarity score of two concepts also depends on the set of their references. Let I_1 (resp. I_2) be the set of instances of c_1 (resp. c_2), the similarity of c_1 and c_2 depends on the similarity scores obtained for the pairs of references of $I_1 \times I_2$ and it is computed by applying the $SoftJaccard$ similarity measure on the sets I_1 and I_2 of references (i.e. objects). $sim_{ref}(c_1, c_2)$ is defined as follows: $sim_{ref}(c_1, c_2) = SoftJaccard_o(I_1, I_2, \theta_4)$.

4.3 Equation modeling the dependencies between similarities in LDM approach

In LDM approach the similarity of each pair of references is expressed by a variable x_i in the instance level equation system. Its value depends on the common description of the pair of references w.r.t the equivalent/ comparable properties (cf. N2R). It depends also on the similarity scores of the concepts sc_i that are instantiated by the pair of references. An equation of the instance level equation system$x_i = g_i(X)$, where $i \in [1..n]$ and n is the number of reference pairs and $X = (x_1, \ldots, x_n)$, is of the form:

$$g_i(X) = \frac{1}{2}(sc_i, f_i(X))$$

with sc_i is the similarity score computed by the resolution of the conceptual equation system presented in the following. The function $f_i(X)$ is expressed as in N2R method and we consider that knowledge on the (inverse) functionality of the shared properties declared in the local ontology is also fulfilled in the LOD ontology.

The similarity of each pair of concepts (c, c') is expressed by a variable xc_j in the conceptual equation system. Its value depends on the initial similarity score provided by the alignment tool, the similarity of their labels, the set of their equivalent / comparable properties and the similarity of their references represented respectively by the constants sim_{j-init}, $sim_{j-label}$, sim_{j-rel} and sim_{j-ref}. It depends also on the similarity of their ancestors and their descendants represented by the variables XSC_{j-anc} and XSC_{j-desc} computed using SotfJaccard function.

An equation $xc_j = h_j(XC)$, where $j \in [1..m]$ and m is the number of concept pairs and $XC = (xc_1, \ldots, xc_m)$, is of the form:

$$h_j(XC) = max(sim_{j-init},$$

$$\frac{1}{5}(XSC_{j-anc} + XSC_{j-desc} + sim_{j-rel} + sim_{j-label} + sim_{j-ref}))$$

The values of the constants sim_{j-init}, $sim_{j-label}$, sim_{j-rel} and sim_{j-ref} are computed using the similarity functions described in the above subsection.

The size m of the conceptual equation system is $\mid C_1 \times C_2 \mid$, where C_1 (resp. C_2) is the set of concepts of the ontology O_1 (resp. O_2). The size of the instance level equation system depends on the number k of comparable relations and on the size of their corresponding domain instances and range instances. Let r_{i1} and r_{i2} be two comparable relations. Let E_{i1} (resp. E_{i2}) be the set of domain instances of r_{i1} (resp. of r_{i2}) and E_{i3} (resp. E_{i4}) be the set of range instances of r_{i3} (resp. r_{i4}). It also depends on the number of comparable attributes k' and on the size of their corresponding domain instances. Let a_{j1} and a_{j2} be two comparable attributes. Let E_{j1} (resp. E_{j2}) be the set of domain instances of a_{j1} (resp. of a_{j2}). The number n of variables of the instance level equation system is:

$$n = \mid \bigcup_{i=1}^{i=k}((E_{i1} \times E_{i2}) \cup (E_{i3} \times E_{i4})) \cup (\bigcup_{j=1}^{j=k'}(E_{j1} \times E_{j2})) \mid$$

The computation complexity of the LDM method is O($(n^2 * it_{ref}) + (m^2 * it_c)$), with it_{ref} is the number of iterations of the instance level equation system and it_c is the number of iterations of the conceptual equation system.

One of the most distinguishing characteristic of LDM is its ability to propagate similarities at different levels: (i) between pairs of concepts, (ii) between pairs of references and (iii) between sets of references and sets of concepts. By using two separated equation systems we avoid the propagation between references when we compute the concept similarity scores and we avoid also the propagation between concepts when we compute the reference similarity scores. Thus, we decrease the size of the equation system and we allow a user to visualize and validate the intermediate equation system results.

4.4 Illustrative example

We present in Figure 3 an extract of the DBLP ontology which is used to describe the DBLP data published in the LOD. The considered data set only contains a collection of conference proceedings and the collection of their corresponding research papers in computer science. In order to illustrate our approach of link discovery, we will compare the local RDF data of the source S1 given in Figure 2 with the extract of DBLP dataset of the LOD given in Figure 4.

The initialization step provides the following initial similarity scores for the concept pairs:
$sim_{init}(Article, InProceedings) = 0.3$; $sim_{init}(Article, Proceedings) = 0.1$;
$sim_{init}(Article, Agent) = 0.1$; $sim_{init}(Person, InProceedings) = 0.0$;
$sim_{init}(Person, Proceedings) = 0.0$; $sim_{init}(Person, Agent) = 0.3$;
$sim_{init}(Conference, InProceedings) = 0.2$;
$sim_{init}(Conference, Proceedings) = 0.2$; $sim_{init}(Conference, Agent) = 0.1$

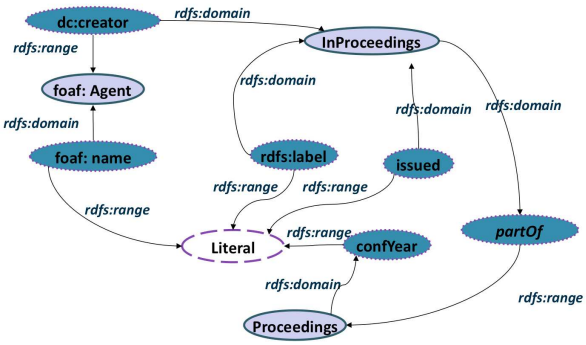

Fig. 3. An extract O2 of LOD DBLP ontology

LOD source S2:
InProceedings(S2_a1); label(S2_a1,"Implementing the TEA algorithm on sensors'); Agent(S2_p1); Agent(S2_p2); issued(S2_a1, "2004"); name(S2_p1,"Olga V. Gavrylyako"); name(S2_p2,"Shuang Liu");

Proceedings(S2_c1); label(S2_c1, "42nd Annual Southeast Regional Conference, 2004"); creator(S2_a1,S2_p1); creator(S2_a1,S2_p2); partOf(S2_a1,S2_c1);
InProceedings(S2_a2); label(S2_a2,"New Chaos Produced from Synchronization of Chaotic Neural Networks"); Agent(S2_p3); issued(S2_a2, "2008");
name(S2_p3,"Zunshui Cheng");

Proceedings(S2_c2); label(S2_c2, "Advances in Neural Networks - ISNN 2008, 5th International Symposium on Neural Networks"); creator(S2_a2,S2_p3); partOf(S2_a2,S2_c2);

Fig. 4. An extract of DBLP data set on the LOD

Since, there are no subsumption relations in $O1$ and in $O2$ the conceptual equations do not take into account the similarity scores of the ancestors and of the descendants. For example, the equation expressing the similarity of the two concepts $Article$ and $InProceedings$ is: $xc_1 = \max(0.3, \frac{1}{3}(\frac{2}{3} + 0 + sim_{1-ref}))$. In this example, the conceptual equation system consists of nine variables $(xc_1, \ldots, xc_9)$.

The instance level equation system consists of twenty-five equations representing all the reference pairs where the common description is not empty. For example, the equations expressing:

- The similarity of the two references $S1\_a1$ (Article) and $S2\_a1$ (InProceedings) is: $x_1 = \frac{1}{2}(sc_1, \max(\frac{1}{2}(Sim_v(label.val(S2\_a1), title.val(S1\_a1)) + Sim_v(issued.val(S2\_a1), year.val(S1\_a1)), \frac{1}{4}(SJ(\{S1\_p1, S1\_p2\}, \{S2\_p1, S2\_p2\}) + x_{14})$
- The similarity of the two references $S1\_c1$ (Conference) and $S2\_c1$(Proceedings) is : $x_{14} = \frac{1}{2}(sc_{14}, \max(x_1, \frac{1}{2}(Sim_v(label.val(S2\_c1), confName.val(S1\_c1)))$
- The similarity of the two references $S1\_p1$ (Person) and $S2\_p1$ (Agent) is:

  $x_5 = \frac{1}{2}((sc_5, \frac{1}{2}(x_1 + Sim_v(name.val(S1\_p1), name.val(S2\_p1))$

– The similarity of the two references $S1\_p1$ (Conference) and $S2\_p1$ (InProceedings) is: $x_{18} = \frac{1}{2}(sc_{18}, \max(Sim_v(confName.val(S1\_c1), label.val(S2\_a1))$

In Table 2 we show the iterative resolution of the conceptual equation system ES_1 modeling the similarity of all the pairs of concepts of the ontologies $O1$ and $O2$. The column sim_{init} represents the initial similarity score computed by an external concept alignment tool, like TaxoMap [9]. The Table 3 shows the results of the iterative resolution of the instance level equation system ES_2 of the pairs of references coming from the local source S1 of Figure 2 which conforms to the local ontology O1 and the S2 LOD source which conforms to the LOD DBLP ontology O2.

Variables of ES_1	**sim$_{init}$**	**Resolution1– iteration1**	**Resolution2– it1**
xc_1 = (Article, InProceedings)	0.3	$\max(0.3, \frac{1}{3}(\frac{4}{6})) = 0.3$	xc_1= 0.33
xc_2 = (Article, Proceedings)	0.1	$\max(0.1, \frac{1}{3}(\frac{1}{6})) = 0.1$	xc_2= 0.1
xc_3 = (Article, Agent)	0.1	$\max(0.1, \frac{1}{3}(0)) = 0.1$	xc_3= 0.1
xc_4 = (Person, InProceedings)	0.0	$\max(0, \frac{1}{3}(0)) = 0.0$	xc_4= 0.0
xc_5 = (Person, Proceedings)	0.0	$\max(0, \frac{1}{3}(0)) = 0.0$	xc_5= 0.0
xc_6 = (Person, Agent)	0.3	$\max(0.3, \frac{1}{3}(1)) = 0.33$	xc_6= 0.427
xc_7 = (Conference, InProceedings)	0.2	$\max(0.2, \frac{1}{3}(\frac{1}{4})) = 0.2$	xc_7= 0.2
xc_8 = (Conference, Proceedings)	0.2	$\max(0.2, \frac{1}{3}(\frac{2}{4})) = 0.25$	xc_8= 0.33
xc_9 = (Conference, Agent)	0.1	$\max(0.1, \frac{1}{3}(0)) = 0.1$	xc_9= 0.1

Table 2. *The two resolutions of the conceptual equation system ES_1*

The $Resolution1$ step of ES_1 corresponds to the first iterative resolution of ES_1 where sim_{ref} of all the concepts equals to 0. The fix-point of $\epsilon = 0.05$ is reached in two iterations. The $Resolution1$ step of ES_2 corresponds to the first iterative resolution of ES_2 where sc_i of all the references equals to sim_{init} (c.f. Table 2) . The fix-point of $\epsilon = 0.05$ is reached in three iterations. The $Resolution2$ step of ES_1 corresponds to the second iterative resolution of ES_1 where sim_{ref} of all the concepts equals to the similarity scores computed by ES_2 at the last iteration of $Resolution1$. The fix-point of $\epsilon = 0.05$ is also reached in two iterations. The $Resolution2$ step of ES_2 corresponds to the second iterative resolution of ES_2 where sc_i of all the references equals to the similarity scores computed by ES_1 at the last iteration of $Resolution1$. The fix point of $\epsilon = 0.05$ is reached in two iterations.

The global fix-point is reached after three resolutions. At $Resolution3$ [2] of the two systems ES_1 and ES_2 we obtain the same similarity scores than the last iteration of their corresponding $Resolution2$ step. The results obtained by ES_1 show that the method obtains the best similarity scores for the most possible equivalent concepts: $(Article, InProceedings)$, $(Person, Agent)$ and $(Conference, Proceedings)$. In an analogous way, the results obtained by ES_2 show that the best similarity scores are obtained for the most possible *owl:same-as* references. If we fix the reconciliation

[2] The scores are not shown here, they are equal to those obtained in the $Resolution2$ of ES_1 and ES_2.

Variables of ES_2	Resolution1– iteration 1	Res1–it2	Res1–it3	Res2–it1
x_1 = ($S1\_a1$,$S2\_a1$)	$\frac{1}{2}(0.3 + \max(\frac{1}{2}(2), \frac{1}{4}(0)) = 0.66$	0.66	0.66	0.665
x_2 = ($S1\_a2$,$S2\_a1$)	$\frac{1}{2}(0.3 + \max(\frac{1}{2}(0), \frac{1}{4}(0)) = 0.15$	0.165	0.175	0.19
...	...	...	...	...
x_5 =($S1\_p1$,$S2\_p1$)	$\frac{1}{2}(0.33 + \frac{1}{2}(1)) = 0.415$	0.58	0.58	0.62
x_6 = ($S1\_p2$,$S2\_p1$)	$\frac{1}{2}(0.33 + \frac{1}{2}(0)) = 0.165$	0.33	0.33	0.379
...	...	...	...	...
x_{14} = ($S1\_c1$,$S2\_c1$)	$\frac{1}{2}(0.25 + \max(0, \frac{1}{2}(0.438)) = 0.219$	0.455	0.455	0.477
...	...	...	...	...
x_{18} = ($S1\_c1$,$S2\_a1$)	$\frac{1}{2}(0.2 + \max(0)) = 0.1$	0.1	0.1	0.1
...	...	...	...	...

Table 3. *The resolution of the instance level equation system* ES_2

threshold at 0.45 we infer the reconciliation of the two papers ($S1\_a1$, $S2\_a1$), of the two persons ($S1\_p1$, $S2\_p1$) and of the two conferences ($S1\_c1$, $S2\_c1$).

In this example we have shown the applicability of the approach even when the considered ontologies are not syntactically close and when they have very poor structure (no subsumption relations) which means that the ancestors and the descendants are not considered.

5 Conclusion and Future Work

In this paper we have presented a Link Discovering Method (LDM) which allows discovery of new data sources that are published in the Open Linked Data cloud (LOD). Our approach is based on the idea of comparing a local dataset on which domain knowledge can be declared and a LOD dataset by using a combined ontology reconciliation and reference reconciliation method. By using our LDM method one may discover more *owl:same-as* links with datasets available on the LOD.

One of the most distinguishing characteristic of our link discovery approach resides on its ability to propagate similarities at different levels: (i) between pairs of concepts, (ii) between pairs of references and (iii) between sets of references and sets of concepts. By using two separated equation systems we avoid the propagation between references when we compute the concept similarity scores and we avoid also the propagation between concepts when we compute the reference similarity scores.

As a very short term perspective, we plan to test our LDM approach on real data sets and evaluate the quality of its results and its scalability. It will be worth to compare LDM method with those of existing link discovery methods like [10]. As future work, we plan to extend the approach to be able, in addition of the equivalent properties, take into account the other properties in oder to consider richer data descriptions. Moreover, we aim also to extend the LDM method to compute also similarities between the properties of the considered ontologies.

References

1. Berners-Lee, T., Cailliau, R., Groff, J.F., Pollermann, B.: World-wide web: The information universe. Electronic Networking: Research, Applications and Policy 1(2), 74–82 (1992)
2. Bilenko, M., Mooney, R.J.: Adaptive duplicate detection using learnable string similarity measures. In: KDD. pp. 39–48 (2003)
3. Bizer, C., Heath, T., Berners-Lee, T.: Linked data - the story so far. Int. J. Semantic Web Inf. Syst. 5(3), 1–22 (2009)
4. Cohen, W.W., Ravikumar, P.D., Fienberg, S.E.: A comparison of string distance metrics for name-matching tasks. In: IIWeb. pp. 73–78 (2003)
5. Cohn, D.A., Atlas, L.E., Ladner, R.E.: Improving generalization with active learning. Machine Learning 15(2), 201–221 (1994)
6. Dong, X., Halevy, A.Y., Madhavan, J.: Reference reconciliation in complex information spaces. In: SIGMOD Conference. pp. 85–96 (2005)
7. Euzenat, J., Loup, D., Touzani, M., Valtchev, P.: Ontology alignment with ola. In: Sure, Y., Corcho, O., Euzenat, J., Hughes, T. (eds.) Proc. 3rd ISWC2004 workshop on Evaluation of Ontology-based tools (EON), Hiroshima (JP). pp. 59–68 (2004)
8. Golub, G.H., Loan, C.F.V.: Matrix computations (3rd ed.). Johns Hopkins University Press, Baltimore, MD, USA (1996), `http://portal.acm.org/citation.cfm?id=248979`
9. Hamdi, F., Safar, B., Niraula, N.B., Reynaud, C.: Taxomap in the oaei 2009 alignment contest. In: Proceedings of the 4th International Workshop on Ontology Matching (OM-2009) collocated with the 8th International Semantic Web Conference (ISWC-2009) Chantilly, USA, October 25, 2009 (2009)
10. Hassanzadeh, O., Kementsietsidis, A., Lim, L., Miller, R.J., Wang, M.: A framework for semantic link discovery over relational data. In: Proceedings of the 18th ACM Conference on Information and Knowledge Management, CIKM 2009, Hong Kong, China, November 2-6, 2009. pp. 1027–1036 (2009)
11. Kiefer, C., Bernstein, A.: The creation and evaluation of isparql strategies for matchmaking. In: The Semantic Web: Research and Applications, 5th European Semantic Web Conference, ESWC 2008, Tenerife, Canary Islands, Spain, June 1-5, 2008, Proceedings. pp. 463–477 (2008)
12. Li, J., Tang, J., Li, Y., Luo, Q.: Rimom: A dynamic multistrategy ontology alignment framework. IEEE Transactions on Knowledge and Data Engineering 21, 1218–1232 (2009)
13. Nikolov, A., Uren, V.S., Motta, E., Roeck, A.N.D.: Handling instance coreferencing in the knofuss architecture. In: Proceedings of the 1st IRSW2008 International Workshop on Identity and Reference on the Semantic Web, Tenerife, Spain, June 2, 2008 (2008)
14. Rahm, E., Bernstein, P.A.: A survey of approaches to automatic schema matching. The VLDB Journal 10(4), 334–350 (2001)
15. Saïs, F., Pernelle, N., Rousset, M.C.: L2r: A logical method for reference reconciliation. In: AAAI. pp. 329–334 (2007)
16. Saïs, F., Pernelle, N., Rousset, M.C.: Combining a logical and a numerical method for data reconciliation. J. Data Semantics 12, 66–94 (2009)
17. Scharffe, F., Liu, Y., Zhou, C.: Rdf-ai: an architecture for rdf datasets matching, fusion and interlink. In: Proc. IJCAI 2009 workshop on Identity, reference, and knowledge representation (IR-KR), Pasadena (CA US) (2009)
18. Shvaiko, P., Euzenat, J.: A survey of schema-based matching approaches pp. 146–171 (2005)
19. Tejada, S., Knoblock, C.A., Minton, S.: Learning object identification rules for information integration. Inf. Syst. 26(8), 607–633 (2001)