HAL
open science

L2R: A Logical Method for Reference Reconciliation

Fatiha Saïs, Nathalie Pernelle, Marie-Christine Rousset

To cite this version:

Fatiha Saïs, Nathalie Pernelle, Marie-Christine Rousset. L2R: A Logical Method for Reference Reconciliation. Twenty-Second AAAI Conference on Artificial Intelligence, Jul 2007, Vancouver, British Columbia, Canada. pp.2007. inria-00433004

HAL Id: inria-00433004

https://inria.hal.science/inria-00433004v1

Submitted on 18 Nov 2009

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

L2R: a Logical method for Reference Reconciliation

Fatiha Saïs
LRI, Paris-Sud 11 University, and
INRIA Futurs, 2-4 rue J. Monod,
F-91893 ORSAY, FRANCE
Fatiha.Sais@lri.fr

Nathalie Pernelle
LRI, Paris-Sud 11 University, and
INRIA Futurs, 2-4 rue J. Monod,
F-91893 ORSAY, FRANCE
Nathalie.Pernelle@lri.fr

Marie-Christine Rousset
LSR-IMAG BP 72, 38402
St MARTIN D'HERES, FRANCE
Marie-Christine.Rousset@imag.fr

Abstract

The reference reconciliation problem consists in deciding whether different identifiers refer to the same data, i.e., correspond to the same world entity. The L2R system exploits the semantics of a rich data model, which extends RDFS by a fragment of OWL-DL and SWRL rules. In L2R, the semantics of the schema is translated into a set of logical rules of reconciliation, which are then used to infer correct decisions both of reconciliation and no reconciliation. In contrast with other approaches, the L2R method has a precision of 100% by construction. First experiments show promising results for recall, and most importantly significant increases when rules are added.

Introduction

The reference reconciliation problem is one of the main problems encountered when different sources have to be integrated. It consists in deciding whether different identifiers refer to the same data, i.e., correspond to the same world entity (e.g. the same person or the same publication).

Schema heterogeneity is a major cause of the mismatch of the data descriptions between sources. Extensive research work has been done recently (see (Rahm & Bernstein 2001; Shvaiko & Euzenat 2005; Noy 2004) for surveys) to reconcile schemas and ontologies through mappings.

However, the homogeneity or reconciliation of the schemas do not prevent variations between the data descriptions. For example, two descriptions of persons with the same attributes Last Name, First Name, Address can vary on the values of those attributes while referring to the same person, for instance, if the First Name is given entirely in one tuple, while it is abbreviated in the second tuple.

Data cleaning which aims at detecting duplicates in databases is faced with the same problem. Most of the existing works (e.g., (Galhardas *et al.* 2001; Bilenko & Mooney 2003; Ananthakrishna, Chaudhuri, & Ganti 2002)) do comparisons between strings for computing the similarity between the values of the same attribute, and then combine them for computing the similarity between tuples. In (Omar *et al.* 2005) the matching between data descriptions is generic but is still based on local comparisons.

Copyright © 2007, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

Some recent works (Kalashnikov, Mehrotra, & Chen. 2005; Dong, Halevy, & Madhavan 2005; Singa & Domingos. 2005; Bhattacharya & Getoor 2006) follow a global approach that exploits the dependencies possibly existing between reference reconciliations. Those dependencies often result from the semantics of the domain of interest. For example, the reconciliation between two courses described by their titles and the name of the professors in charge of them can entail the reconciliation between two descriptions of persons. This requires that some knowledge of the domain be made explicit, like the fact that a professor is a person, that a course is identified by its title and has only one professor in charge of it. In (Dong, Halevy, & Madhavan 2005), such knowledge is taken into account but must be encoded in the weights of the edges of the dependency graph.

In this paper, we study the problem of reference reconciliation in the case where the data are described relatively to a rich schema expressed in RDFS (w3c Rec. b) extended by some primitives of OWL-DL (w3c Rec. a) and SWRL (w3c Sub.). OWL-DL and SWRL are used to state axioms that enrich the semantics of the classes and properties declared in RDFS. It is then possible to express that two classes are disjoint or that some properties (or their inverse) are functional. We describe the L2R system which implements a logical method for reference reconciliation, based on rules of reconciliation that are automatically generated from the axioms of the schema. Those rules are a declarative translation of the dependencies between reconciliations resulting from the semantics of the schema. They enable to infer both sure reconciliations and no reconciliations. We therefore obtain a method with a precision of 100% and we show that the recall is significantly increased if the schema is enriched by adding axioms. L2R is based on the most recent proposals for the Semantic Web (RDF, OWL-DL and SWRL). Therefore, it can be used for reconciling data in most of the applications based on the Semantic Web technologies.

The paper is organized as follows. In Section 2, we define the data model and the problem of reference reconciliation that we consider. In Section 3, we describe the logical method that we have implemented in L2R. In Section 4, we summarize the results that we have obtained for the experimental evaluation of L2R on two data sets. Conclusions and perspectives are presented in Section 5.

Problem definition

We first describe the data model, that we have called RDFS+ because it extends RDFS with some OWL-DL primitives and SWRL rules. RDFS+ can be viewed as a fragment of the relational model (restricted to unary or binary relations) enriched with typing constraints, inclusion and exclusion between relations and functional dependencies.

The RDFS+ data model

The schema: A RDFS schema consists of a set of classes (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 are distinguished in RDFS: the so-called *relations* the domain and the range of which are classes and the so-called *attributes* the domain of which is a class and the range of which is a set of basic values (Integer, date, String,...).
We will use the following notation :
– $R(C, D)$ indicates that the domain of the relation R is the class C and that its range is the class D,
– $A(C, Literal)$ indicates that the domain of the attribute A is the class C and that its range is a set of alpha-numeric values.

For example, in the RDFS schema presented in figure 1 and corresponding to a cultural application, we have as relations *located(Museum, City), contains(CulturalPlace, Painting), paintedBy(Painting, Artist)*
and as attributes *museumName(Museum, Literal), yearOfBirth(Artist, Date)*.

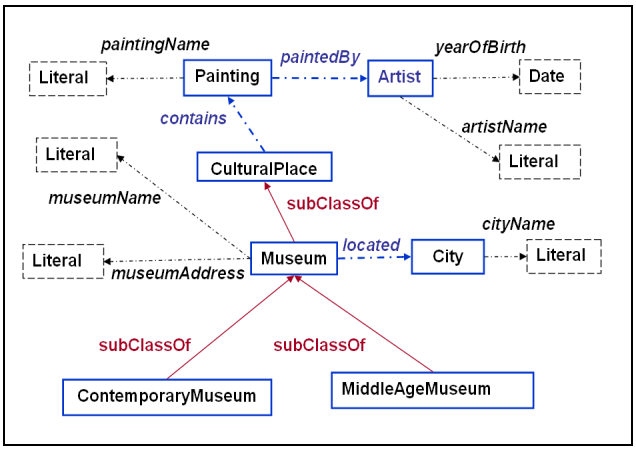

Figure 1: Example of a RDFS schema

The schema axioms: In order to enrich the RDFS schema, we allow axioms of the following types.

- **Axioms of disjunction between classes** $DISJOINT(C, D)$ is used to declare that the two classes C and D are disjoint, for example: $DISJOINT(Museum, Artist)$.
- **Axioms of functionality of properties** $PF(P)$ is used to declare that the property P (relation or attribute) is a functional property. For example, $PF(located)$ and $PF(museumName)$ express respectively that a museum is located in one and only one city and that a museum has only one name. These axioms can be generalized to a set $\{P_1, \ldots, P_n\}$ of relations or attributes to state a combined constraint of functionality that we will denote $PF(P_1, \ldots, P_n)$.
- **Axioms of inverse functionality of properties** $PFI(P)$ is used to declare that the property P (relation or attribute) is an inverse functional property. For example, $PFI(Contains)$ expresses that a painting cannot belong to several cultural places. These axioms 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 $PFI(P_1, \ldots, P_n)$. For example, $PFI(paintingName, paintedBy)$ expresses that one artist and one painting name cannot be associated to several paintings (i.e. both are needed to identify a painting).
- **Axioms for discriminant properties** DISC(A) is used to declare that the attribute A is discriminant if it is known that each possible value of this attribute has a single form (a number or a code). For instance, the attributes *yearOfBirth* and *countryName* can be declared as discriminant.

It is important to note that the axioms of disjunction and of simple functionality (i.e., of the form $PF(P)$ or $PFI(P)$) can be expressed in OWL-DL while the axioms stating combined constraints of functionality (i.e., of the form $PF(P_1, \ldots, P_n)$ or $PFI(P_1, \ldots, P_n)$) and those stating discriminant properties (i.e. DISC(P)) require the expressive power of SWRL.

The data: A datum has a reference which has the form of a URI (e.g. *http://www.louvre.fr, NS-S1/painting243*), and a description which is a set of RDF facts involving its reference. An RDF fact can be:
– either a class-fact $C(i)$, where C is a class and i is a reference,
– or a relation-fact $R(i_1, i_2)$, where R is a relation and i_1 and i_2 are references,
– or an attribute-fact $A(i, v)$, where A is an attribute, i a reference and v a basic value (integer, string, date, ...).

We consider that the descriptions of data coming from different sources conform to the same RDFS+ schema. In order to distinguish the data coming from different sources, we use the source identifier as the prefix of the reference of the data coming from that source. For example, figure 2 provides examples of data coming from two RDF data sources S1 and S2 which conform to a same RDFS+ schema describing the cultural application previously mentioned.

The axioms on the data sources: We consider two kinds of axioms accounting for the Unique Name Assumption (UNA) and the Local Unique Name Assumption (denoted LUNA). The UNA states that two data of the same data source having distinct references refer to two different real world entities (and thus cannot be reconciled). Such an assumption is valid when a data source is clean.

The LUNA is weaker than the UNA, and states that all the references related to a same reference by a relation refer

Source S1 :
CulturalPlace(S1_m1); MiddleAgeMuseum(S1_m3); Painting(S1_p1); Painting(S1_p2);Artist(S1_a1); museumName(S1_m1,*"musee du LOUVRE"*); museumName(S1_m2, *"musee des arts premiers"*); . . .

Source S2 :
Museum(S2_m1); Museum(S2_m2);Painting(S2_p1); Painting(S2_p2); Artist(S2_a1); located(S2_m1,S2_c1); museumName(S2_m1,*"Le LOUVRE"*); contains(S2_m1,S2_p2); . . .

Figure 2: *Example of RDF data*

to real world entities that are pairwise distinct. For example, from the facts $authored(p, a_1)$, ..., $authored(p, a_n)$ coming from the same data source, we can infer that the references $a_1, \ldots, a_n$ correspond to distinct authors of the paper referred to by p.

The reference reconciliation problem

Let S_1 and S_2 be two data sources which conform to the same RDFS+ schema. Let I_1 and I_2 be the two reference sets that correspond respectively to the data of S_1 and S_2. Let *Reconcile*[1] be a binary predicate. *Reconcile*(X, Y) means that the two references denoted by X and Y refer to the same world entity.

The reference reconciliation problem between S_1 and S_2 consists in partitioning the set $I_1 \times I_2$ of reference pairs into two subsets REC and NREC such that for each reference pair $(i_1, i_2) \in REC$ $Reconcile(i_1, i_2)$ and such that for each pair $(i_1, i_2) \in NREC$ $\neg Reconcile(i_1, i_2)$.

A reconciliation method is complete if it provides a result (*Reconcile*(i_1, i_2) or $\neg Reconcile(i_1, i_2)$) for each pair $(i_1, i_2) \in I_1 \times I_2$.

The *precision* of a reconciliation method is the ratio of correct reconciliations and no reconciliations among those found by the method. The *recall* of a reconciliation method is the ratio of correct reconciliations and no reconciliations found by the method among the whole expected set of correct reconciliations and no reconciliations.

The reconciliation method L2R described in the next section is not complete since it does not guarantee to infer reconciliations or no reconciliations for all the pairs of $I_1 \times I_2$. Its distinguishing features are that it is global and logic-based: every schema axiom is automatically translated into logical rules that express dependencies between reconciliations. The advantage of such a logical approach is that it guarantees a 100% precision. Therefore our experiments are focused on estimating its recall.

Reference Reconciliation method

The method is based on the inference of facts of reconciliation ($Reconcile(i, j)$) and of no reconciliation ($\neg Reconcile(i', j')$) from a set of facts and a set of rules which transpose the semantics of the data sources and of the schema into logical dependencies between reference reconciliations. Facts of synonymy ($SynVals(v_1, v_2)$) and of no synonymy ($\neg SynVals(u_1, u_2)$) between basic values (strings, dates) are also inferred. The binary predicate *SynVals* is analogous to the predicate *Reconcile* but applied on basic values.

We first describe the generation of the reconciliation rules, then the generation of the facts and finally the reasoning which is performed on the set of rules and facts to infer reconciliation decisions.

Generation of the set of reconciliation rules

They are automatically generated from the axioms that are declared on the data sources and on their common schema.

Translation of the axioms on the data sources We introduce the unary predicates src1 and src2 in order to label each reference according to its original source ($srci(X)$ means that the reference X is coming from the source Si).

The UNA assumption, if it is stated on the sources $S1$ and $S2$, is translated automatically by the following four rules :

$$R1 : src1(X) \wedge src1(Y) \wedge (X \neq Y) \Rightarrow \neg Reconcile(X, Y)$$

$$R2 : src2(X) \wedge src2(Y) \wedge (X \neq Y) \Rightarrow \neg Reconcile(X, Y)$$

$$R3 : src1(X) \wedge src1(Z) \wedge src2(Y) \wedge Reconcile(X, Y) \Rightarrow \neg Reconcile(Z, Y)$$

$$R4 : src1(X) \wedge src2(Y) \wedge src2(Z) \wedge Reconcile(X, Y) \Rightarrow \neg Reconcile(X, Z)$$

The first two rules express the fact that two references coming from the same source cannot be reconciled. The last ones mean that one reference coming from a source $S2$ (resp. $S1$) can be reconciled with at most one reference coming from a source $S1$(resp. $S2$).

For each relation R, the LUNA assumption is translated automatically by the following rules denoted respectively $R11(R)$ and $R12(R)$:

$$R(Z, X) \wedge R(Z, Y) \wedge (X \neq Y) \Rightarrow \neg Reconcile(X, Y)$$

$$R(X, Z) \wedge R(Y, Z) \wedge (X \neq Y) \Rightarrow \neg Reconcile(X, Y)$$

Translation of the schema axioms For each pair of classes C and D involved in a $DISJOINT(C, D)$ statement declared in the schema, or such that their disjunction is inferred by inheritance, the following rule is generated:

$$R5(C, D) : C(X) \wedge D(Y) \Rightarrow \neg Reconcile(X, Y)$$

For each relation R declared as functional by the axiom $PF(R)$, the following rule $R6.1(R)$ is generated :

$$Reconcile(X, Y) \wedge R(X, Z) \wedge R(Y, W) \Rightarrow Reconcile(Z, W)$$

For each attribute A declared as functional by the axiom $PF(A)$, the following rule $R6.2(A)$ is generated :

$$Reconcile(X, Y) \wedge A(X, Z) \wedge A(Y, W) \Rightarrow SynVals(Z, W)$$

[1] Reconcile and not Reconcile can also be expressed in OWL by using *sameAs* and *differentFrom* predicates.

The binary predicate *SynVals* replaces the predicate *Reconcile*, applied on basic values.
For each relation R declared as inverse functional by the axiom $PFI(R)$, the following rule $R7.1(R)$ is generated :

$$Reconcile(X,Y) \wedge R(Z,X) \wedge R(W,Y) \Rightarrow Reconcile(Z,W)$$

For each attribute A declared as inverse functional by the axiom $PFI(A)$, the following rule $R7.2(A)$ is generated :

$$SynVals(X,Y) \wedge A(Z,X) \wedge A(W,Y) \Rightarrow Reconcile(Z,W)$$

Likewise, analogous rules are generated for translating axioms $PF(P_1,\ldots,P_n)$ of combined functionality and $PFI(P_1,\ldots,P_n)$ of combined inverse functionality.
For instance, $PF(P_1,\ldots,P_n)$, where all the P_i's are relations, is translated into the rule:

$$R7.1(P_1,\ldots,P_n) : \bigwedge_{i\in[1..n]} [P_i(Z,X_i) \wedge P_i(W,Y_i) \wedge Reconcile(X_i,Y_i)] \Rightarrow Reconcile(Z,W)$$

If some P_i's are attributes, the corresponding $Reconcile(X_i,Y_i)$ must be replaced by $SynVals(X_i,Y_i)$.
Similarly, $PFI(P_1,\ldots,P_n)$, where all the P_i's are relations, is translated into the rule:

$$R7.2(P_1,\ldots,P_n) : \bigwedge_{i\in[1..n]} [P_i(X_i,Z) \wedge P_i(Y_i,W) \wedge Reconcile(X_i,Y_i)] \Rightarrow Reconcile(Z,W)$$

Finally, for each attribute A declared as discriminant by the axiom DISC(A) the following rule $R8(A)$ is generated :

$$\neg SynVals(X,Y) \wedge A(Z,X) \wedge A(W,Y) \Rightarrow \neg Reconcile(Z,W)$$

Transitivity rule : this rule is generated only if the UNA axiom is not stated on the data sources.

$$R9 : Reconcile(X,Y) \wedge Reconcile(Y,Z) \Rightarrow Reconcile(X,Z)$$

Generation of the set of facts

The set of RDF facts corresponding to the description of the data in the two sources S_1 and S_2 is augmented with the generation of:

- new class-facts, relation-facts and attribute-facts obtained by inheritance, i.e., by exploiting the subsumption statements between classes and properties that are stated into the RDFS schema: for example if the fact *ContemporaryMuseum(i)* is present in one of the sources, the class-facts $Museum(i)$ and $CulturalPlace(i)$ are added to the description of that source;
- facts of the form $src1(i)$ and $src2(j)$ for each reference $i \in I_1$ and each reference $j \in I_2$,
- synonymy facts of the form $SynVals(v_1,v_2)$ for each pair (v_1,v_2) of basic values that are identical (up to some punctuation or case variations): for instance, the fact *SynVals("La Joconde", "la joconde")* is added because these two values differ only by two capital letters,
- non synonymy facts of the form $\neg SynVals(v_1,v_2)$ for each pair (v_1,v_2) of distinct basic values associated with so-called discriminant attributes. For instance, $\neg SynVals("2004","2001")$, $\neg SynVals("FRANCE","PORTUGAL")$ are added if *yearOfBirth* and *countryName* are declared *discriminant*.

Reasoning

The reasoning applies to the union $\mathcal{R} \cup \mathcal{F}$ of the set of rules and the set of facts generated as explained before. It must infer all the facts of reconciliation, no reconciliation, synonymy, no synonymy that are logically entailed by $\mathcal{R} \cup \mathcal{S}$, based on the standard first-order logical semantics.

It is important to notice that the reconciliation rules though having negative conclusions still correspond to Horn clauses, for which there exists reasoning methods that are complete for the inference of prime implicates, like for instance SLD resolution (Chang & Lee 1997).

The reasoning algorithm that we have implemented in L2R is a three-steps application of SLD resolution (Chang & Lee 1997) to the Horn clausal form of the rules and to the facts that are unit ground clauses.

Propositionalization step: all the possible resolutions of the ground facts in $\mathcal{F}$ with the Horn clauses corresponding to the rules in $\mathcal{R}\backslash\{R9\}$ are computed. It consists in propagating the ground facts into the rules except the transitivity rule $R9$. The result is a set $\mathcal{P}$ of fully instantiated Horn clauses in which the only remaining literals are of the form $Reconcile(i,j)$, $\neg Reconcile(i',j')$, $SynVals(u,v)$, or $\neg SynVals(u',v')$, and each atom $Reconcile(i,j)$ or $SynVals(u,v)$ is seen as a propositional variable.

Propositional inference step: the propositional SLD resolution is applied to the set $\mathcal{P}$ of the *propositional* Horn clauses resulting from the first step.

Transitivity step: This step applies only if the rule $R9$ is in $\mathcal{R}$, i.e., only if the UNA axiom is not stated on the data sources. The first-order SLD resolution is applied to the ground facts obtained at the previous step and to the clausal form of $R9$.

It is easy to show that for any literal l of the form $Reconcile(i,j)$, $\neg Reconcile(i',j')$, $SynVals(u,v)$, or $\neg SynVals(u',v')$: $\mathcal{R} \cup \mathcal{F} \models l$ iff $\mathcal{P} \models l$

Since the SLD resolution on Horn clauses is complete, and since $\mathcal{R} \cup \mathcal{F}$ and $\mathcal{P}$ are equivalent for the derivation of ground literals, this algorithm guarantees to derive all the facts of the form $Reconcile(i,j)$, $\neg Reconcile(i',j')$, $SynVals(u,v)$, or $\neg SynVals(u',v')$ that can be logically entailed from the set of rules and the set of facts.

Other reasoners, like for instance description logic reasoners, could be used for the derivation of reconciliation facts. However, description logics are not specially appropriate to express some of the reconciliation rules that we consider, which require explicit variable bindings. In addition, up to our knowledge, the existing description logic reasoners are not guaranteed to be complete for the computation of prime implicates.

Experiments

The L2R rule-based method has been implemented and tested on data sets related to two different domains: the tourism domain and the scientific publications.

Presentation of the data sets(HOTELS and CORA)

The first real data set HOTELS, provided by an industrial partner, corresponds to a set of seven data sources which leads to a pairwise data integration problem of 21 pairs of data sources. These data sources contain 28,934 references to hotels located in Europe. The UNA is stated for each source.

The hotel descriptions in the different sources are very heterogeneous. First, the properties that are instantiated are different from one to another. Second, the basic values are multilingual, contain abbreviations, and so on.

The second data set CORA[2] (used by (Dong, Halevy, & Madhavan 2005)) is a collection of 1295 citations of 112 different research papers in computer science. In this data set, the objective of the reference reconciliation is the cleaning of a given data source (i.e. duplicates elimination). The reference reconciliation problem applies then to $I \times I$ where I is the set of references of the data source S to be cleaned. For this data set, the UNA is not stated and the RDF facts describe references which belong to three different classes (*Article*, *Conference*, *Person*).

The RDFS+ schemas : HOTELS conforms to a RDFS schema of tourism domain, which is provided by the industrial partner. We have added a set of disjunction axioms (e.g. DISJOINT(Hotel, Service)), a set of functional property axioms (e.g. *PF(establishmentName)*) and a set of inverse functional property axioms (e.g. *PFI(establishmentName, associatedAddress)*).

For the CORA data set, we have designed a simple RDFS schema on the scientific publication domain, which we have enriched with disjunction axioms (e.g. DISJOINT(Article, Conference)), a set of functional property axioms (e.g. *PF(published)*, *PF(confName)*) and a set of inverse functional property axioms (e.g. *PFI(title, year,type)*, *PFI(confName, confYear)*).

Results

Since the set of reconciliations and the set of no reconciliations are obtained by a logical rule-based algorithm the precision is of 100% by construction. Then, the measure that it is meaningful to evaluate in our experiments is the recall. For the CORA data set, the expected results for reference reconciliation are provided. Therefore, the recall can be easily obtained by computing the ratio of the reconciliations and no reconciliations obtained by L2R among those that are provided.

For the HOTELS data set, we have manually detected the correct reconciliations and no reconciliations between the references of two data sources containing respectively 404 and 1392 references to hotels. For the other pairs of data sources, we provide quantitative results, i.e. the number of reconciliations and no reconciliations.

In the following, we summarize the results obtained on the HOTELS data set and then those obtained on the CORA data set. We emphasize the impact on the recall of increasing the expressiveness of the schema by adding axioms.

[2]another version of CORA is provided by McCallum, (http://www.cs.umass.edu/ mccallum/data/cora-refs.tar.gz)

	RDFS+		**RDFS+ & {DA or DP}**	
	HOTELS	CORA	HOTELS	CORA
Recall (REC)	54 %	52.7 %	54 %	52.7 %
Recall (NREC)	8.2 %	50.6 %	75.9 %	94.9 %
Recall	8.3 %	50.7 %	75.9 %	94.4 %
Precision	100 %	100 %	100 %	100 %

Figure 3: *L2R results on HOTELS and CORA data sets*

Results on HOTELS data set For the quantitative results, the application of L2R on the 21 pairs of data sources leads to 1063 reconciliations and 251,523,187 no reconciliations.

In the figure 3, we show the recall that we have obtained on the two sources on which we have manually detected the reconciliation and no reconcilation pairs. We distinguish the recall computed only on the set of reconciled references (REC) and only on not reconciled references (NREC).

As it is shown in the column named "RDFS+ (HOTELS)" of the figure 3, we have obtained a recall of 8.3%. If we only consider the reconciliations subset (REC) the recall is 54%. The REC subset corresponds to the reconciliations inferred by exploiting the inverse functional axiom *PFI(establishmentName, associatedAddress)*. It is important to emphasize that those reconciliations are inferred in spite of the irregularities in the data descriptions: not valued addresses and a lot of variability in the values, in particular in the addresses : "*parc des fees*" vs. "*parc des fees, (nearby Royan)*". In addition, in one of the data sources, several languages are used for the basic values: "*Chatatoa*" versus "*Chahatoenia*" in Basque language.

If we only consider the no reconciliations subset (NREC) the recall is 8.2%. Actually, the only rules that are likely to infer no reconciliations are those translating the UNA assumption. Now, if we enrich the schema just by declaring pairwise disjoint specializations of the $Hotel$ class (by distinguishing hotels by their countries), we obtain an impressive increasing of the recall on NREC, from 8.2% to 75.9%, as it is shown in the "RDFS+ (HOTELS) & DA" column.

Results on CORA data set We focus on the results obtained for the *Article* and *Conference* classes, which contain respectively 1295 references and 1292 references.

As presented in the column named "RDFS+ (CORA)" of the figure 3, the recall obtained on the CORA data set is 50.7%. This can be refined in a recall of 52.7% computed on the REC subset and a recall of 50.6% computed on NREC subset. The set of inferred reconciliations (REC subset) for references to articles is obtained by exploiting the axiom PFI(*Title*, *Year*) of combined inverse functionality on the properties $title$ and $year$. For the conferences, 35.8% of the reconciliations are obtained by exploiting the axiom PFI(*confName*, *confYear*) of combined inverse functionality on the attributes $confName$ and $confYear$, and 64.1% are obtained by propagating the reconciliations of references to articles, using the axiom PF(*published*) of functionality of the relation $published$.

The set of inferred no reconciliations (NREC subset) are

obtained by exploiting the axiom of disjunction between the *Article* and *Conference* classes.

For this data set, the RDFS+ schema can be easily enriched by the declaration that the property *confYear* is discriminant. When this discriminant property is exploited, the recall on the REC subset remains unchanged (52.7%) but the recall on NREC subset grows to 94.9%, as it is shown in the "RDFS+ (CORA) & DP" column. This significant improvement is due to chaining of different rules of reconciliations: the no reconciliations on references to conferences for which the values of the *confYear* are different entail in turn no reconciliations of the associated articles by exploiting the axiom PF(*published*).

This recall is comparable to (while a little bit lower than) the recall on the same data set obtained by *supervised* methods like e.g., (Dong, Halevy, & Madhavan 2005). The point is that L2R is *not supervised* and guarantees a 100% precision.

For the references of the class *Person*, since there is no axiom concerning the properties associated to that class, only the LUNA assumption can infer no reconciliations. For 3521 Person references, 4298 no reconciliations have been inferred by using the rules corresponding to LUNA application to the *author* relation.

Conclusion

L2R is a logical method for the reference reconciliation problem. One of the advantages of such an approach is that it provides reconciliations and no reconciliations that are sure. This distinguishes L2R from other existing works. This is an important point since, as it has been emphasized in (Winkler 2006), unsupervised approaches which deal with the reference reconciliation problem have a lot of difficulties to estimate in advance the precision of their system when it is applied to a new set of data. The experiments show promising results for recall, and most importantly its significant increasing when rules are added. This shows the interest and the power of the generic and flexible approach of L2R since it is quite easy to add rules to express constraints on the domain of interest.

Inferring no reconciliation can be related to the so-called *blocking* methods introduced in (Newcombe & Kennedy 1962) and used in recent approaches such as (Baxter, Christen, & Churches 2003) for reducing the reconcilation space. Note also that our definition of LUNA is used by (Dong, Halevy, & Madhavan 2005) to eliminate inconsistent reconciliations after the reconciliation step. In our approach, the different kinds of knowledge used for reducing the reconciliation space are handled in a declarative way.

The main perspective that we plan is to complement the L2R logical method (which does not guarantee to provide a result for every pair of references) with a numerical method. This numerical step will apply to pairs of references for which the logical step has not produced any result of reconciliation or of no reconciliation. It will compute scores of similarities between references based on the combination of similarities between related sets of values and references. This step will take into account the L2R results by assigning a maximum similarity score to the inferred reconciliations and synonyms. We plan to exploit the results of the logical step to learn the weighting coefficients involved in the combination of the different similarity scores.

References

Ananthakrishna, R.; Chaudhuri, S.; and Ganti, V. 2002. Eliminating fuzzy duplicates in data warehouses. In *SIGMOD '02: Proceedings of the 2002 ACM SIGMOD international conference on Management of data*. ACM Press.

Baxter, R.; Christen, P.; and Churches, T. 2003. A comparison of fast blocking methods for record linkage. In *ACM workshop on Data cleaning Record Linkage and Object identification*.

Bhattacharya, I., and Getoor, L. 2006. *Entity Resolution in Graphs*. Wiley. chapter Entity Resolution in Graphs.

Bilenko, M., and Mooney, R. 2003. Adaptive duplicate detection using learnable string similarity measures. In *SIGKDD'03*.

Chang, C.-L., and Lee, R. C.-T. 1997. *Symbolic Logic and Mechanical Theorem Proving*. Orlando, FL, USA: Academic Press, Inc.

Dong, X.; Halevy, A.; and Madhavan, J. 2005. Reference reconciliation in complex information spaces. In *SIGMOD '05: Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, 85–96. New York, NY, USA: ACM Press.

Galhardas, H.; Florescu, D.; Shasha, D.; Simon, E.; and Saita., C. 2001. Declarative data cleaning: Language, model and algorithms. In *VLDB '01*.

Kalashnikov, D.; Mehrotra, S.; and Chen., Z. 2005. Exploiting relationships for domain-independent data cleaning. In *SIAM Data Mining '05*.

Newcombe, H. B., and Kennedy, J. M. 1962. Record linkage: making maximum use of the discriminating power of identifying information. *Commun. ACM* 5(11):563–566.

Noy, N. 2004. Semantic integration: a survey on ontology-based approaches. *SIGMOD Record, Special Issue on Semantic Integration*.

Omar, B.; Hector, G.-M.; Jeff, J.; Euijong, W. S.; Qi, S.; and Jennifer, W. 2005. Swoosh: A generic approach to entity resolution. Technical repport, Stanford infoLab.

Rahm, E., and Bernstein, P. 2001. A survey of approaches to automatic schema matching. *VLDB Journal* 10:334–350.

Shvaiko, P., and Euzenat, J. 2005. A survey of schema-based matching approaches. *Journal on Data semantics*.

Singa, P., and Domingos., P. 2005. Object identification with attribute-mediated dependences. In *PKDD '05*.

w3c Rec. http://www.w3.org/tr/owl-features/.

w3c Rec. http://www.w3.org/tr/rdf-schema/.

w3c Sub. http://www.w3.org/submission/swrl/.

Winkler, W. E. 2006. Overview of record linkage and current research directions. Technical report, Statistical Research Division U.S. Census Bureau Washington, DC 20233.