Approche logique pour la réconciliation de références

Fatiha Saïs *, Nathalie Pernelle *
Marie-Christine Rousset**

* LRI, Université Paris-Sud 11, F-91405 Orsay Cedex,
INRIA Futurs, 2-4 rue Jacques Monod, F-91893 Orsay Cedex, France
{prenom.nom}@lri.fr
**LSR-IMAG BP 72, 38402 St MARTIN D'HERES CEDEX
Marie-Christine.Rousset@imag.fr
http://www-lsr.imag.fr/Les.Personnes/Marie-Christine.Rousset/

Résumé. Le problème de réconciliation de références consiste à décider si deux descriptions provenant de sources distinctes réfèrent ou non à la même entité du monde réel. Dans cet article, nous étudions ce problème quand le schéma des données est décrit en RDFS étendu par certaines primitives de OWL-DL. Nous décrivons et montrons l'intérêt d'une approche logique basée sur des règles de réconciliation qui peuvent être générées automatiquement à partir des axiomes du schéma. Ces règles traduisent de façon déclarative les dépendances entre réconciliations qui découlent de la sémantique du schéma. Les premiers résultats ont été obtenus sur des données réelles dans le cadre du projet PICSEL 3 en collaboration avec France Telecom R&D.

1 Introduction

Le problème de réconciliation de références est un problème majeur pour l'intégration ou la fusion de données provenant de plusieurs sources. Il consiste à décider si deux descriptions provenant de sources distinctes réfèrent ou non à la même entité du monde réel (e.g., la même personne, le même article, le même gène, le même hôtel).

Il est très difficile d'attaquer ce problème dans toute sa généralité car les causes d'hétérogénéité dans la description de données provenant de différentes sources sont variées et peuvent être de nature très différente. L'hétérogénéité des schémas est une des cause premières de la disparité de description des données entre sources. De nombreux travaux, dont on peut trouver une synthèse dans Rahm et Bernstein (2001); Shvaiko et Euzenat (2005); Noy (2004); Euzenat et Valtchev (2004), ont proposé des solutions pour réconcilier des schémas ou des ontologies par des mappings. Ces mappings peuvent ensuite être utilisés pour traduire des requêtes de l'interface de requêtes d'une source vers l'interface de requête d'une autre source.

L'homogénéité ou la réconciliation de schémas n'empêchent cependant pas les variations entre les descriptions des instances elles-mêmes. Par exemple, deux descriptions de personnes avec les mêmes attributs Nom, Prénom, Adresse peuvent différer sur certaines valeurs de ces attributs tout en référant à la même personne, par exemple, si dans l'un des tuples le prénom est en entier alors que dans l'autre tuple il n'est donné qu'en abrégé.

Les travaux en nettoyage de données qui visent la détection de doublons dans des bases de données sont confrontés exactement à ce problème. La plupart des travaux existants (e.g., Galhardas et al. (2001); Bilenko et Mooney. (2003); Ananthakrishna et al. (2002)) se fondent sur des comparaisons entre chaînes de caractères pour calculer la similarité entre valeurs d'un même attribut, puis calculent la similarité entre deux tuples en combinant les similarités trouvées entre les valeurs de chaque attribut de ces deux tuples. Dans l'approche proposé par Benjelloun et al. (2006) la comparaison de références est générique mais reste une comparaison locale deux à deux. Quelques travaux très récents (Bhattacharya et Getoor. (2004); Kalashnikov et al. (2005); Dong et al. (2005); Singa et Domingos. (2005)) ont une approche globale exploitant les dépendances qui peuvent exister entre réconciliations de références. Souvent, ces dépendances découlent de la sémantique du domaine. Par exemple, la réconciliation entre deux références à des cours décrits par leur intitulé et le nom de l'enseignant responsable peut entraîner la réconciliation entre deux références à des personnes. Cela nécessite l'explicitation de connaissances supplémentaires sur le domaine d'application, comme le fait qu'un enseignant est une personne, et qu'un cours est identifié par son intitulé et n'a qu'un enseignant responsable. Dans Dong et al. (2005), des connaissances du domaine de ce type sont prises en compte mais doivent être codées dans le poids des arcs du graphe de dépendances dont les noeuds correspondent aux paires de références potentiellement réconciliables.

Dans cet article, nous étudions le problème de réconciliation de références dans le cas où les données à réconcilier sont décrites relativement à une même ontologie, vue comme un schéma sémantiquement riche, décrit en RDFS(*http ://www.w3.org/TR/rdf-schema/*) étendu par certaines primitives de OWL-DL (*http ://www.w3.org/2004/OWL*). OWL-DL sert à poser des axiomes qui enrichissent la sémantique des classes et des propriétés déclarées en RDFS. On peut ainsi par exemple exprimer que deux classes sont disjointes ou que telle ou telle propriété (ou son inverse) est fonctionnelle. Nous montrons l'intérêt d'une approche logique basée sur des règles de réconciliation qui peuvent être générées automatiquement à partir des axiomes du schéma. Ces règles traduisent de façon déclarative les dépendances entre réconciliations qui découlent de la sémantique du schéma. Elles permettent d'inférer de façon sûre des réconciliations ainsi que des non réconciliations. Nous obtenons ainsi une méthode ayant une précision de 100%, et nous montrons que le rappel est augmenté de façon significative si on enrichit la sémantique du schéma en ajoutant des axiomes. Cette méthode permet d'obtenir comme produit dérivé direct un dictionnaire de synonymie entre chaînes de caractères. Ce dictionnaire peut être exploité dans une phase ultérieure de réconciliation des paires de références non traitées par la méthode logique et pour lesquelles nous envisageons une méthode numérique dont nous décrivons brièvement le principe à la fin de l'article.

L'article est organisé de la façon suivante. La section 2 définit le modèle de données (RDFS+) et le problème de réconciliation de références que nous considérons. La section 3 décrit la méthode logique que nous proposons qui repose sur des règles d'inférences traduisant les contraintes du schéma en des dépendances logiques entre réconciliations de référence. La section 4 fournit le résultat d'expérimentations que nous avons effectué sur deux jeux de données : celui de CORA qui sert de benchmark dans plusieurs travaux de nettoyage de données ; et un jeu de données fourni par France Telecom R&D dans le cadre du projet PICSEL3. La section 5 conclue l'article en situant notre approche par rapport à l'existant et en indiquant quelques perspectives.

2 Définition du problème

Nous décrivons d'abord le modèle des données que nous considérons, que nous appelons RDFS+ car il étend RDFS avec des primitives de OWL-DL. D'un point de vue "bases de données", RDFS+ peut être vu comme un fragment du modèle relationnel (restreint à des relations unaires et binaires) enrichi par la possibilité d'exprimer des contraintes de typage, d'inclusion ou d'exclusion entre relations, et de dépendances fonctionnelles.

2.1 Le modèle de données RDFS+

Le schéma : Nous considérons que nous disposons d'un schéma RDFS consistant en un ensemble de classes (relations unaires) structurées en une taxonomie et d'un ensemble de propriétés (relations binaires) qui peuvent être elles-mêmes structurées en une taxonomie de propriétés. Les propriétés sont typées. Dans la terminologie RDFS, on distingue les propriétés qui sont des relations, dont les domaines et co-domaines sont des classes, de celles dont le co-domaine est un ensemble de valeurs de base (numériques ou alpha-numériques), et qu'on appelle des attributs.
On notera :

- $R(C, D)$ pour indiquer que le domaine de la relation R est la classe C et que son co-domaine est la classe D, et
- $A(C, Litteral)$ pour indiquer que l'attribut A a comme domaine C et comme co-domaine un ensemble de valeurs (numériques ou alpha-numériques).

Les axiomes : Nous donnons la possibilité de déclarer des axiomes OWL-DL pour enrichir la sémantique d'un schéma RDFS. Les axiomes que nous considérons sont de plusieurs types. Nous ne donnons pas leur notation standard en XML, très verbeuse, mais nous précisons leur sémantique logique.

- **Axiomes de disjonction entre classes.** Nous notons $DISJOINT(C, D)$ l'axiome déclarant que les classes C et D sont disjointes, dont la sémantique logique est : $\forall X\, C(X) \Rightarrow \neg D(X)$.
- **Axiomes de fonctionnalité d'une propriété.** Nous notons $PF(P)$ l'axiome déclarant que la propriété P (relation ou attribut) est fonctionnelle, dont la sémantique logique est : $\forall X, Y, Z\, P(X, Y) \wedge P(X, Z) \Rightarrow Y = Z$.
- **Axiomes de fonctionnalité de l'inverse d'une propriété.** Nous notons $PFI(P)$ l'axiome déclarant que l'inverse de la propriété P (relation ou attribut) est fonctionnelle, dont la sémantique logique est : $\forall X, Y, Z\, P(Y, X) \wedge P(Z, X) \Rightarrow Y = Z$.

Les données : Une donnée a un identifiant (appelé référence) et une description qui est l'ensemble des faits RDF (*http ://www.w3.org/RDF/*) qui mentionnent cet identifiant. Un fait RDF est :

– soit un fait-classe de la forme $C(i)$ où i est un identifiant,

– soit un fait-relation de la forme $R(i_1, i_2)$ où R est une relation et i_1 et i_2 sont des identifiants,

– soit un fait-attribut de la forme $A(i, v)$ où A est un attribut, i un identifiant et v une valeur (numérique ou alpha-numérique).

Nous supposons que les données peuvent provenir de plusieurs sources et nous préfixons l'identifiant d'une donnée par l'identifiant de la source dont elle provient. Sauf mention explicite, nous posons par défaut *l'hypothèse du nom unique* sur chaque source. Cette hypothèse (notée UNA) a la sémantique suivante : deux données d'une même source ayant des identifiants distincts font référence à des entités distinctes du monde réel.

2.2 Exemple

Afin d'illustrer le modèle de données RDFS+, nous montrons ici un exemple de schéma RDFS, un ensemble d'axioms OWL-DL et enfin un exemple de données conformes à ce schéma qui porte sur le domaine des lieux culturels. Nous utiliserons pour l'exemple la notation graphique de RDFS.

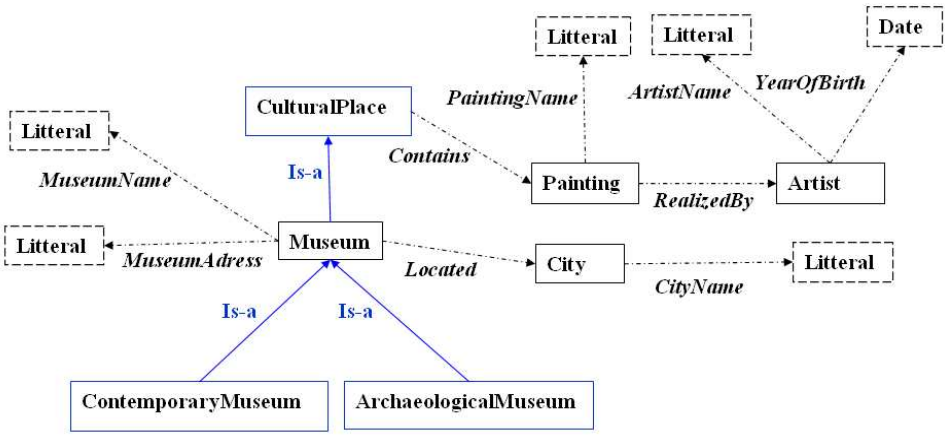

FIG. 1 – *Exemple de schéma RDFS*

Le schéma de la figure 1, contient plusieurs classes représentant des lieux culturels. Par exemple la classe *Museum* subsume les classes *ArchaeologicalMuseum* et *ContemporaryMuseum*. A cet ensemble de classes est associé un ensemble de relations et d'attributs. Par exemple la relation *Located* relie la classe *Museum* à la ville (*City*) où il est situé. Les attributs *ArtistName* et *YearOfBirth* associés à la classe *Artist* représentent le nom et la date de naissance de l'artiste.

Pour enrichir ce schéma nous déclarons l'ensemble d'axiomes suivant :

– L'ensemble des axiomes de disjonction entre classes est : {DISJOINT(*ArchaeologicalMuseum*, *ContemporaryMuseum*), DISJOINT(*Painting*,*Artist*), DISJOINT(*CulturalPlace*, *Painting*), DISJOINT(*CulturalPlace*, *City*) }.
Ces axiomes expriment, par exemple, qu'un musée ne peut pas être à la fois un musée archéologique et un musée contemporain.

– L'ensemble des axiomes de fonctionnalité de propriétés est : {PF(*Located*), PF(*PaintedBy*), PF(*ArtistName*), PF(*YearOfBirth*), PF(*PaintingName*), PF(*MuseumName*), PF(*MuseumAddress*), PF(*CityName*)}.

  Ces axiomes expriment, par exemple, qu'un musée ne peut être localisé que dans une seule ville et qu'un musée n'a qu'un seul nom.

– Axiomes de fonctionnalité inverse de propriété : { PFI(*Contains*), PFI(*ArtistName*), PFI(*PaintingName*), PFI(*MuseumName*), PFI(*MuseumAddress*), PFI(*CityName*)}.

  Ces axiomes expriment, par exemple, qu'une peinture ne peut être contenue que dans un seul musée et qu'un nom de peinture n'est associé qu'à une seule peinture.

Soient S1 et S2 deux sources de données RDF conformes au schéma RDFS de la figure 1 et vérifiant les axiomes ci-dessus. Nous présentons le contenu de ces deux sources sous forme d'un ensemble de faits RDF.

La source S1 :
CulturalPlace(S1_m1) ; Painting(S1_p1) ; Artist(S1_a1) ; Museum(S1_m2) ;
PaintingName(S1_p1,"La Joconde") ; PaintedBy(S1_p1, S1_a1) ; Contains(S1_m1, S1_p1),
MuseumName(S1_m1,"musée du LOUVRE") ; ArtistName(S1_a1, "Leonard De Vinci") ;
YearofBirth(S1_a1,"1452") ; Painting(S1_p2) ; PaintingName(S1_p2,"La Cene") ;
PaintedBy(S1_p2, S1_a1), Painting(S1_p3) ; PaintingName(S1_p3,"Sainte Anne") ;
PaintedBy(S1_p3, S1_a1) ; MuseumName(S1_m2, "musée des arts premiers") ;
Located(S1_m2, S1_c1) ; CityName(S1_c1,"Paris") ; MuseumAddress(S1_m2, "quai branly")

La source S2 :
Museum(S2_m1) ; Museum(S2_m2) ; Artist(S2_a1) ; MuseumName(S2_m1,"Le LOUVRE") ;
Located(S2_m1,S2_c1) ; CityName(S2_c1, "Ville de paris") ; Contains(S2_m1, S2_p1) ;
PaintingName(S2_p1, "la Joconde") ; Contains(S2_m1,S2_p2) ; PaintedBy(S2_p2,S2_a1) ;
PaintingName(S2_p2, "Vierge aux rochers") ; ArtistName(S2_a1, "De Vinci") ;
YearofBirth(S2_a1,"1452") ; Contains(S2_m1,S2_p3) ; Located(S2_m2,S2_c1) ;
MuseumName(S2_m2,"Musée du quai Branly") ;
PaintingName(S2_p3, "Sainte Anne, la vierge et l'enfant jésus "),
MuseumAddress(S2_m2, "37 quai branly, portail Debilly"),

FIG. 2 – *Exemple de données RDF*

2.3 Le problème de réconciliation

Soient S_1 et S_2 deux sources de données ayant le même schéma RDFS+. Soient I_1 et I_2 les deux ensembles d'identifiants de leurs données respectives.

Le problème de réconciliation entre S_1 et S_2 consiste à partitioner l'ensemble $I_1 \times I_2$ des paires de références en 2 sous-ensembles $Reconcile$ et $NonReconcile$ regroupant respectivement les paires de références représentant une même entité, et les paires de références représentant deux entités différentes.

Dans la suite de l'article, on utilisera la notation relationnelle plutôt que la notation ensembliste : $Reconcile(i1, i2)$ (respectivement $NonReconcile(i1, i2)$) pour $(i1, i2) \in Reconcile$ (respectivement pour $(i1, i2) \in NonReconcile$).

Une méthode de réconciliation est totale si elle produit un résultat ($Reconcile(i1, i2)$ ou $NonReconcile(i1, i2)$) pour tout couple $(i_1, i_2) \in I_1 \times I_2$.

La *précision* d'une méthode de réconciliation est la proportion, parmi les couples pour lesquels la méthode a produit un résultat (de réconciliation ou de non réconciliation), de ceux pour lesquels le résultat est correct.

Le *rappel* d'une méthode de réconciliation est la proportion, parmi tous les couples possibles de $I_1 \times I_2$, de ceux pour lesquels la méthode a produit un résultat correct.

La méthode de réconciliation que nous décrivons dans la section suivante est une méthode de réconciliation partielle qui a la caractéristique d'être globale et fondée sur la logique : elle traduit les axiomes du schéma par des règles logiques de dépendances entre réconciliations. L'intérêt d'une approche logique est qu'elle garantit une précision de 100%. Notre expérimentation se focalise donc sur l'estimation du rappel.

3 Méthode logique de réconciliation à base de règles

Notre approche consiste à traduire les axiomes associés au schéma, incluant l'UNA (quand elle s'applique), par des règles logiques (Section 3.1), et à appliquer un algorithme de raisonnement pour inférer des réconciliations et des non réconciliations (Section 3.2), ainsi que des synonymies entre valeurs de base qui seront conservées dans un dictionnaire (Section 3.3).

3.1 Génération des règles de réconciliation et de non réconciliation

Traduction de l'hypothèse du nom unique par des règles de non réconciliation : On introduit les prédicats unaires src1 et src2 pour typer chaque référence en fonction de sa source d'origine (srci(X) signifie que la référence X provient de la source Si). La contrainte de l'UNA au niveau des sources S1 et S2 se traduit par les quatre règles suivantes :

R1 : src1(X) $\wedge$ **src1(Y)** $\wedge$ **(**$X \neq Y$**)** $\Rightarrow$ **NonReconcile(X, Y),**
R2 : src2(X) $\wedge$ **src2(Y)** $\wedge$ **(**$X \neq Y$**)** $\Rightarrow$ **NonReconcile(X, Y),**
R3 : src1(X) $\wedge$ **src1(Z)** $\wedge$ **src2(Y)** $\wedge$ **Reconcile(X, Y)** $\Rightarrow$ **NonReconcile(Z, Y)**
R4 : src1(X) $\wedge$ **src2(Y)** $\wedge$ **src2(Z)** $\wedge$ **Reconcile(X, Y)** $\Rightarrow$ **NonReconcile(X, Z)**

Les deux premières règles traduisent la non réconciliation de deux références provenant d'une même source. Les deux dernières traduisent le fait qu'une référence provenant d'une source S1 (resp. S2) peut être réconciliée avec au maximum une référence de la source S2 (resp. S1).

Traduction des disjonctions entre classes par des règles de non réconciliation : Pour chaque paire de classes C et D déclarées disjointes dans le schéma (DISJOINT(C,D)) ou inférées comme telles par héritage, la règle suivante est générée :

R5(C,D) : C(X) $\wedge$ **D(Y)** $\Rightarrow$ **NonReconcile(X, Y)**

Traduction des axiomes de fonctionnalité par des règles de réconciliation de références et de synonymies de valeurs :

– Pour toute relation R déclarée comme fonctionnelle par un axiome PF(R), la règle R6.1(R) est générée, qui traduit le fait que pour une instance de la classe du domaine de R il existe au plus une instance du co-domaine.

R6.1(R) : Reconcile (X, Y) $\wedge$ R(X, Z) $\wedge$ R(Y, W) $\Rightarrow$ Reconcile(Z,W)

– Pour tout attribut A déclaré comme fonctionnel par un axiome PF(A), la règle R6.2(A) est générée, qui exprime que pour une instance de la classe du domaine de A il existe au plus une valeur de base appartenant au co-domaine. Le prédicat binaire EquiVals permet d'exprimer que deux valeurs de base sont synonymes. Il est l'équivalent sur les valeurs de base du prédicat Reconcile.

R6.2(A) : Reconcile (X, Y) $\wedge$ A(X, Z) $\wedge$ A(Y, W) $\Rightarrow$ EquiVals(Z,W)

– Pour toute relation R déclarée comme fonctionnelle inverse par un axiome PFI(R), la règle règle R7.1(R) est générée, qui exprime que pour une instance de la classe du co-domaine de R il existe au plus une instance du domaine.

R7.1(R) : Reconcile (X, Y) $\wedge$ R(Z, X) $\wedge$ R(W, Y) $\Rightarrow$ Reconcile(Z,W)

– Pour tout attribut A déclaré comme fonctionnel inverse par un axiome PFI(A), la règle R7.2(A) est générée, qui traduit le fait que pour deux valeurs de base synonymes appartenant au co-domaine il existe au plus une instance de la classe du domaine.

R7.2(A) : EquiVals (X, Y) $\wedge$ A(Z, X) $\wedge$ A(W, Y) $\Rightarrow$ Reconcile(Z,W)

Remarque. On peut traduire par des règles analogues aux règles précédentes des dépendances fonctionnelles impliquant plusieurs relations ou attributs.

3.2 Inférence de réconciliations et de non réconciliations

L'algorithme du chaînage avant est appliqué sur la base de connaissances composée de la base des règles présentées ci-dessus et de la base de faits contenant :

– l'ensemble de faits-classe, faits-relation et de faits-attribut représentant les descriptions de l'ensemble de références des deux sources étendue par les propriétés obtenues par héritage ;

– un fait de type src1(X) ou src2(X) pour toute référence X permettant de représenter sa provenance ;

– un ensemble de faits instance du prédicat EquiVals(X,Y) qui expriment l'égalité à une normalisation près (élimination des ponctuations, des mots vides) des valeurs de base. Ainsi, le fait EquiVals("La Joconde", "la joconde") est posé car les deux chaines de caractères ne diffèrent que par deux majuscules.

En nous appuyant sur les données de la figure 2 et sur le schéma de la figure 1, nous allons montrer comment les décisions de réconciliations et de non réconciliation se propagent grâce à l'enchainement des règles.

Les règles R1 et R2 (traduisant l'UNA) puis les règles R5(CulturalPlace, Painting) et R5(Artist, Painting), traduisant des axiomes de disjonction entre classes, permettent d'inférer, entre autres, les non réconciliations suivantes :

NonReconcile(S1_m1, S1_m2), NonReconcile (S1_p1, S1_p2),
NonReconcile(S2_m1, S2_p1), NonReconcile (S2_p1, S2_p2),
NonReconcile(S1_m1, S2_p1), NonReconcile (S1_a1, S2_p1).

La règle R7.2(PaintingName) traduisant la propriété de fonctionnalité inverse de l'attribut PaintingName, et les faits PaintingName(S1_p1, "La joconde"), PaintingName(S2_p1, "La Joconde") et EquiVals("La Joconde", "la joconde") permettent d'inférer Reconcile(S2_p1, S1_p1), c'est-à-dire que S2_p1 et S1_p1 réfèrent à la même peinture.

Les musées S1_m1 et S2_m2 contenant ces peintures sont eux mêmes réconciliés par déclenchement de la règle R7.1(Contains), qui permet d'inférer Reconcile(S1_m1,S2_m2).

La propagation de ces nouvelles réconciliations permettra grâce à la règle R6.2(MuseumName) d'inférer Equivals("Le LOUVRE", "musée du LOUVRE"), établissant la synonymie entre les valeurs de base "Le LOUVRE" et "musée du LOUVRE", et grâce à la règle R6.1(Located) d'inférer la réconciliation des deux références sur les villes S1_c1 et S2_c1. Le fait inféré Reconcile(S1_c1 , S2_c1) permet ensuite d'inférer, par le déclenchement de la règle R6.2(CityName), le nouveau fait EquiVals("ville de Paris","Paris") qui établit la synonymie entre les deux valeurs de base "ville de Paris" et "Paris".

Les règles R3 et R4 de traduction de l'UNA permettent d'éliminer toute autre possibilité de réconciliation. La règle R4 permet ainsi d'inférer NonReconcile(S2_m2, S1_m1) à partir des faits Reconcile(S1_m1, S2_m1) src1(S1_m1), src2(S2_m1) et scr2(S2_m2) : par UNA, on sait que les deux musées S2_m1 et S2_m2 sont différents, et comme on a réconcilié les deux références S1_m1 et S2_m2 au musée du Louvre, on est sûr que la référence S1_m1 de la source S1 au musée du Louvre n'est pas réconciliable avec la référence S2_m2 de la source S2 (qui est une référence au musée du quai Branly).

3.3 Génération et exploitation du dictionnaire

L'ensemble des synonymies qui sont inférées entre valeurs de base (prédicat Equivals) sont conservées dans un dictionnaire qui est alimenté au fur et à mesure des réconciliations de références provenant de différentes sources.

Le dictionnaire contient différents types de synonymies :

– Des codifications telles que *1* pour *oui*, *75* pour *Paris* et *(*)* pour *étoile*

– Des abréviations telles que *apt* pour *appartement*, ou acronymes tels que *ACM* pour *Association for Computing Machinery*

– Des vrais synonymes tels que *bon* pour *confortable*

– Des traductions telles que *Milano* pour *Milan*

Nous avons vu que ces équivalences sont exploitées durant la phase de réconciliation elle-même. Le fait de conserver ces valeurs dans un dictionnaire permet également d'utiliser ces connaissances lorsque l'approche logique est appliquée à d'autres sources. Le dictionnaire peut permettre d'alimenter la base de faits intiale par tous les faits EquiVals qu'il contient. Ce dictionnaire peut aussi être exploité dans une méthode ou une étape de réconciliation numérique fondée sur le calcul de similarité entre chaînes de caractères.

4 Expérimentation

Nous présentons dans cette section les premiers résultats de l'expérimentation de notre approche logique de réconciliation de références. Cette méthode a été testée sur des données de deux domaines différents : des données du domaine du tourisme et des données d'un portail

de publications scientifiques en informatique. Dans le premier jeu de données, en présence de l'UNA au niveau de chaque source, la réconciliation de références a pour objectif l'intégration de données entre différentes sources. Dans le second jeu de données, l'objectif de la réconciliation est de nettoyer une source (ie. éliminer les doublons) pour laquelle l'UNA n'est pas posée.

4.1 Présentation des données de test (FT_HOTELS et CORA)

Notre premier jeu de données FT_HOTELS, fournit par France Telecom R&D dans le cadre du projet PICSEL3, représente un ensemble de sept sources de données contenant 28934 références d'hotels situés en Europe. Ces données sont rendues conformes au schéma RDFS+ par le biais de wrappers. L'UNA est vérifiée au niveau de chaque source. Nous donnons ci-après un extrait du schéma RDFS+ dont nous disposons :

- les classes : *Établissement*, *Hotel*, *Restaurant*, *Séjour*, *Service*, ...
- les propriété (attributs) : *NomHotelAssocie*, *AdresseAssociee*, *PaysAssocie*, *Etoiles*, ...
- les axiomes :
  - de disjonction : des références d'hotels sont forcément différentes de références de séjours et différentes de celles de services, ... ;
  - de fonctionnalité : toutes les propriétés sont fonctionnelles à l'exception de *AutreService*, *AutreDescription* et *URLEtablissement* ;
  - de fonctionnalité inverse : un seul axiome mais qui porte sur les deux attributs *NomHotelAssocie* et *AdresseAssociee*, et qui exprime qu'un hôtel est identifié de manière unique si son nom et son adresse ont des valeurs identiques.

L'ensemble de références d'hotels dont on dispose sont décrites de manière très variable dans les différentes sources. En particulier, la propriété *AdresseAssociee* n'est pas toujours renseignée. De plus, les valeurs de base sont décrites de manière très hétérogène : elles sont multilingues, contiennent des abbréviations, etc.

Le second jeu de données[1] est une collection de 1879 citations d'articles de 112 articles de recherche en informatique différents. Ces articles ont été récupérés à partir du moteur de recherche d'articles scientifiques en informatique CORA. A chaque citation est associée un identifiant et donc une référence. Les données extraites sont rendues conformes au schéma RDFS+ dont un extrait est donné ci-après :

- les classes : *Publication*, *JournalPaper*, *ProceedingsPaper*, *TechnicalReportPaper* ;
- les propriétés : *HasTitle*, *HasAuthors*, *PublishedIn*, *HasPages*, *HasDate*, *hasType* ;
- les axiomes de :
  - de disjonction : les articles publiés dans des journaux sont forcément différents de ceux publiés dans des proceeding ou en technical report. Cela est traduit par DISJOINT(*JournalPaper*, *ProceedingsPaper*) et DISJOINT(*JournalPaper*, *TechnicalReportPaper*) ;
  - de fonctionnalité : toutes les propriétés sont fonctionnelles y compris la propriété *HasAuthors* représentant la liste des auteurs non distingués ;

[1] CORA rendu disponible par McCallum à (http ://www.cs.umass.edu/ mccallum/data/cora-refs.tar.gz)

– de fonctionnalité inverse : un seul axiome mais qui porte sur deux attributs *hasTitle* et *hasType* (*hasType* prend des valeurs dans : {proceedings, journal, technical-report})

4.2 Résultats et validation

Comme l'ensemble des réconciliations et des non réconciliations est obtenu par un algorithme d'inférence à base de règles logiques, nous avons donc une précision à 100%. Il nous reste donc à évaluer le rappel. Pour calculer le rappel sur les données de CORA, nous avons comparé le nombre de réconciliations et de non réconciliations par rapport au nombre de celles qu'il fallait effectivement trouver. L'information concernant des réconciliations et des non réconciliations effectives est représentée dans les données de CORA sous forme d'annotations sur les références. En revanche, nous ne disposons pas de cette information pour le jeu de données FT_HOTELS. Nous avons donc effectué la validation à la main sur un échantillon de 1796 références représentant les références de deux sources de données de taille (404 x 1392). Pour nous faciliter la tâche de recherche des réconciliations oubliées, nous avons examiné l'ensemble des réconciliations possibles, après filtrage, par des recherches mots-clés.

Nous présentons dans le tableau 3 le rappel global concernant les décisions de réconciliations (paires réconciliées ou non réconciliées). De plus, nous donnons le détail du rappel concernant l'ensemble des paires de références réconciliées et l'ensemble des paires de références non réconciliées.

Dans le but de montrer l'intérêt de la richesse du schéma sur le rappel obtenu par notre algorithme, nous présentons les résultats sur le jeu de données FT_HOTELS dans deux cas : (1) pas d'axiomes de disjonctions entre classes d'hotels et (2) ajout au schéma d'un ensemble d'axiomes de disjonctions traduisant le fait que deux hotels situés dans deux pays différents sont forcément différents.

	FT_HOTELS sans Disj	**FT_HOTELS avec Disj**	**CORA**
Rappel (global)	8.3 %	75.9 %	36 %
Rappel (Reconcile.)	54 %	54 %	79 %
Rappel (NonReconcile.)	8.2 %	75.9 %	33 %

FIG. 3 – *Résultats de l'approche logique de réconciliation de références*

Comme le montre la figure 3, sur le jeu de données FT_HOTELS, nous avons obtenu un rappel global de 8.3 % avec un rappel correspondant à celui du sous-ensemble des *NonReconcile* 8.2 % qui représente uniquement les inférences réalisées à partir des règles de l'UNA. Le rappel correspondant au sous-ensemble des *Reconcile* est de 54 % malgré les irrégularités dans les valeurs. Il s'agit essentiellement d'absences d'information (eg. adresse non renseignée) ou de variabilités dans les valeurs, en particulier dans les adresses : "*parc des fées*" vs. "*parc des fées, (près de Royan)*", ou encore "*11, place d'arme*" vs. "*place d'arme*". De plus, certaines données de FT_HOTELS sont décrites en plusieurs langues : "*Chatatoa*" vs. en basque "*Chahatoenia*". Ces résultats montrent également l'apport de l'enrichissement du schéma par une connaissance telle que la disjonction entre hotels situés dans des pays différents. Cet ajout est peu coûteux et pourtant il permet d'augmenter le rappel du sous-ensemble *NonReconcile* de 8.2 % à 75.9 %.

En ce qui concerne le jeu de données CORA, nous avons obtenu un bon rappel sur le sous-ensemble *Reconcile* 79 %. Dong et al. (2005) obtiennent un meilleur rappel (97%)

mais n'ont pas une précision de 100%. Le rappel sur le sous-ensemble *NonReconcile* est seulement de 33 %. En effet, ce dernier résultat est obtenu en exploitant uniquement deux axiomes de disjonction du schéma et sans hypothèse d'UNA.

Nous avons également inféré un ensemble de synonymes que nous avons stockées dans un dictionnaire. Par exemple, pour le jeu de données FT_HOTELS, pour lequel nous avons obtenu 1063 réconciliations (au total), le dictionnaire généré contient 3671 synonymies. Ces dernières représentent essentiellement des codifications d'information (eg.*EquiVals("1","Y"))*, des traductions telles que *EquiVals("Florence","FIRENZE")* et des descriptions syntaxiquement proches telles que *EquiVals("2100","DK-2100")* qui sont des codes postaux au Danemark mais aussi *EquiVals("Avignon - Le Pontet","LE PONTET")*.

5 Conclusion

L'approche logique que nous venons de présenter pour la réconciliation de références présente l'intérêt de ne produire que des réconciliations et des non réconciliations sûres, ce qui la distingue des autres travaux existants. La sûreté de cet ensemble de réconciliations est un atout dans un domaine où il est difficile d'estimer à l'avance le taux d'erreurs d'une approche non supervisée Winkler (2006). Cette approche permet également de découvrir des synonymies entre valeurs de base, conservées dans un dictionnaire qui s'enrichit au fur et à mesure que de nouvelles réconciliations sont inférées, et dont l'enrichissement entraîne de nouvelles réconciliations. Tout en garantissant une précision de 100%, les premières expérimentations montrent que notre méthode obtient un taux de rappel très satisfaisant, qui augmente significativement quand on rajoute des connaissances sur le schéma des données.

D'autres travaux - appelés *blocking method* - utilisent des connaissances du domaine pour réduire le nombre de paires de références à considérer Baxter R. (2003). Ces travaux considèrent seulement les paires qui possèdent une caractéristique donnée commune (exemple : le nom de famille pour des personnes). Ce type de connaissance peut être déclaré comme disjonction dans le schema afin d'être exploité comme le sont les autres connaissances du domaine. C'est ce que nous avons fait pour tirer parti des valeurs de pays des hotels. Nous avons mentionné dans l'introduction que des connaissances du domaine pouvait être également traduites par des poids dans approche numérique Dong et al. (2005).

La méthode que nous avons proposé est partielle car elle ne produit pas de résultat pour toutes les paires de références possibles. Nous envisageons d'augmenter le rappel des non réconciliations en exploitant les contraposées de règles. Cela permettra également d'enrichir le dictionnaire par de nouvelles connaissances sur les valeurs de base : l'inférence de non Equivals(v1,v2) nous permettra de sélectionner les valeurs dont la similarité syntaxique est bonne et qui, pourtant, ne signifient pas la même chose (exemples : *musée du prado* vs *musée du lido*, *chatillon sur marne* vs *chatillon sur seine*).

Pour obtenir une méthode totale de réconciliation de références, nous prévoyons d'appliquer une méthode numérique de réconciliation à l'ensemble des paires de référence pour lesquelles l'étape logique n'a pas produit de résultat. Pour cette étape numérique, nous avons plusieurs pistes pour exploiter les résultats de réconciliation et non réconciliation produit par

l'étape logique, en particulier pour apprendre les poids à associer aux différentes similarités calculées dans les descriptions de deux références.

Références

Ananthakrishna, R., S. Chaudhuri, et V. G. . (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., C. T. (2003). A comparison of fast blocking methods for record linkage. In *ACM workshop on Data cleaning Record Linkage and Object identification*.

Benjelloun, O., H. Garcia-Molina, et H. Kawai (2006). Generic entity resolution in the serf project. *IEEE Data Eng. Bull. 29*(2), 13–20.

Bhattacharya, I. et L. Getoor. (2004). Iterative record linkage for cleaning and integration. In *DMKD'04*.

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

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

Euzenat, J. et P. Valtchev (2004). Similarity-based ontology alignment in owl-lite. In *ECAI*, pp. 333–337.

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

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

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

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

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

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

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

Summary

The reference reconciliation problem consists in deciding whether different identifiers refer to the same data, i.e., correspond to the same world entity. In this work, we exploit the semantics of the RDFS+ (RFDS extended by a fragment of OWL-DL) data model, by generating a set of rules used to infer sure decisions both of reconciliation and no reconciliation. We also present the first results which are obtained on two different real data sets.