Réconciliation de références : une approche logique adaptée aux grands volumes de données

Fatiha Saïs[1], Nathalie Pernelle[1], and Marie-Christine Rousset[2]

[1] 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
[2] LSR-IMAG BP 72, 38402 St MARTIN D'HERES CEDEX
Marie-Christine.Rousset@imag.fr

Abstract. 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. Cependant, la tâche de réconciliation de références est souvent confrontée à des grandes quantités de données mais aussi à des contraintes de performances en temps d'exécution. En vue d'un passage à l'échelle, nous présentons dans cet article, des techniques que nous intégrons dans notre approche logique L2R de réconciliation de références.

Keywords: Reconciliation de références, efficacité, performance, intégration sémantique de données, ontologies, logique.

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 [12, 14, 9], 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.

Dans cet article, nous présentons comment notre méthode logique de réconciliation de références peut s'adapter à de grands volumes de données. Cette méthode est appliquée sur des données décrites relativement à une même ontologie, vue comme un schéma sémantiquement riche, décrit en RDFS étendu par certaines primitives de OWL-DL. OWL-DL sert à poser des axiomes qui enrichissent la sémantique des classes et des propriétés déclarées en RDFS.

Avec l'emergence d'un grand nombre de sources de données sur le web et le besoin de plus en plus accru pour la conception de portails d'informations, les méthodes de réconciliations de références doivent faire face à de très grands volumes de données et à de fortes contraintes de temps d'exécution. Par exemple, les sites comparateurs de prix (e.g. www.kelkoo.com) doivent traiter des millions de références par jour. Certains portails d'information se donnent pour règle de répondre à 30 requêtes en moins de 3 secondes. Ce constat nous conduit à l'obligation de spécifier des méthodes de réconciliation les plus automatiques possible et les moins gourmandes en temps et en espace. Le passage à l'échelle du Web de telles méthodes suppose soit la possibilité de disposer d'un ensemble de connaissances permettant de filtrer les données à réconcilier avant de réaliser des traitements plus complexes pour diminuer l'espace des réconciliations possibles, ou de découper le problème en problèmes qui peuvent être résolus de manière indépendante et éventuellement parallèle.

L'article est organisé de la façon suivante. En section 2 nous définissons le modèle de données, le problème de réconciliation et nous présentons la méthode logique de réconciliation de références. En section 3, nous présentons brièvement des techniques pour améliorer l'efficacité en espace et temps de la méthode de réconciliation. En fin la section 4 conclut l'article.

2 Présentation de l'approche de réconciliation de références

Nous proposons une méthode logique de réconciliation de références qui exploite la sémantique des connaissances du domaine représentées dans le modèle de données RDFS+. Ces connaissance sont traduites en un ensemble de règles qui sont ensuite utilisées par une étape de raisonnement pour inférer des réconciliations et des non réconciliations sûres.

Nous décrivons d'abord le modèle de données que nous avons appelé RDFS+, car il étend RDFS [2] par quelques primitives de OWL-DL [1] et par des règles SWRL[3]. RDFS+ peut être considéré comme un fragment du modèle relationnel (restreint au relation unaires et binaires) enrichi par des contraintes de typage, d'inclusion et d'exclusion entre les relations et d'un ensemble de dépendances fonctionnelles. Ensuite, nous donnerons la représentation des données et enfin nous finirons par une brève présentation de la méthode de réconciliations de références.

2.1 Modèle de données

Le schéma: Un schéma RDFS est composé d'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 (Integer, date, Literal, ...), 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 (Integer, Date, Literal, ...).

Par exemple, dans le schéma RDFS présenté en figure 1 et correspondant au domaine des lieux culturels, nous avons comme relations *Located(Museum, City), Contains(CulturalPlace, Painting), PaintedBy(Painting, Artist)* et comme attributs *MuseumName(Museum, Literal), YearOfBirth(Artist, Date).*

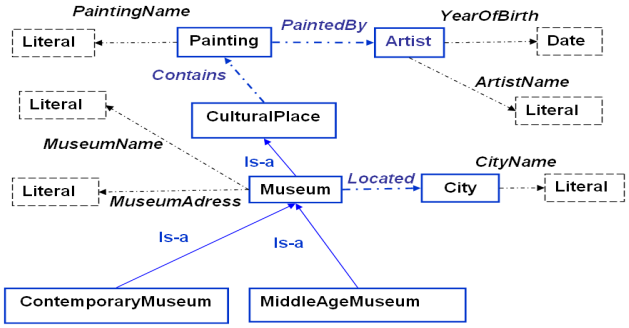

Fig. 1. Exemple de schéma RDFS

Les axiomes du schéma: Nous donnons la possibilité de déclarer des axiomes pour enrichir la sémantique d'un schéma RDFS. Les axiomes que nous considérons sont de plusieurs types.

- **Axiomes de disjonction entre classes.** Nous notons $DISJOINT(C, D)$ l'axiome déclarant que les classes C et D sont disjointes. Par exemple *DISJOINT(Museum,Artist)*.
- **Axiomes de fonctionnalité d'une propriété.** Nous notons $PF(P)$ l'axiome déclarant que la propriété P (relation ou attribut) est fonctionnelle. Par exemple PF(Located) et PF(MuseumName) expriment respectivement qu'un musée est localisé dans une et une seule ville et qu'un musée a un et un seul nom. Ces axiomes peuvent être généralisés à un ensemble $\{P_1, \ldots, P_n\}$ de relations ou d'attributs pour déclarer une combinaison des contraintes de fonctionnalité noté $PF(P_1, \ldots, P_n)$.
- **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. Par exemple $PFI(Contains)$ exprime qu'une peinture ne peut être contenue que dans un seul musée. Ces axiomes peuvent être généralisés à un

ensemble $\{P_1, \ldots, P_n\}$ de relations ou d'attributs pour déclarer une combinaison des contraintes de fonctionnalité inverse noté $PFI(P_1, \ldots, P_n)$. Par exemple, *PFI(PaintingName, PaintedBy)* exprime qu'un artiste et un nom de peinture ne peuvent pas être associés à plusieurs peintures (i.e. les deux sont nécessaires pour identifier la peinture).

– **Axiomes pour les propriétés discriminantes.** Nous notons DISC(A) l'axiome déclarant qu'un attribut A est discriminant. Ce qui signifie que il est connu que toute valeur possible de A a une seule forme (un nombre ou un code). Par exemple les attributs *Year* et *CountryName* peuvent être déclarés comme discriminants.

Il est important de noter que les axiomes de disjonction et de fonctionnalité simple (i.e., de la forme $PF(P)$ ou $PFI(P)$) peuvent être exprimés en OWL-DL alors que les axiomes déclarant les contraintes de fonctionnalité combinées (i.e., de la forme $PF(P_1, \ldots, P_n)$ ou $PFI(P_1, \ldots, P_n)$) et ceux déclarant les propriétés discriminantes (i.e. DISC(P)) nécessitent le pouvoir d'expression du langage SWRL.

Les données: Une donnée a un identifiant (appelé référence) de la forme d'une URI (e.g. *http://www.louvre.fr*, *NS-S1/painting243*), et a une description qui est l'ensemble des faits RDFS qui mentionnent cette référence. Un fait RDFS 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 considérons que les descriptions de données provenant de différentes sources sont conformes au schéma RDFS+. Afin de distinguer les données provenant de différentes sources, nous utilisons l'identifiant de la source comme préfixe des références. Ces sources de données sont conformes au schema RDFS+ du domaine des lieux culturels (figure 1).

Les axiomes sur les sources de données: Nous considérons deux types d'axiomes. Ils concernent l'hypothèse du nom unique (UNA–Unique Name Assumption) et l'hypothèse du nom unique locale (LUNA–Local Unique Name Assumption). L'UNA déclare que deux données provenant de la même source et ayant deux références différentes réfèrent forcément à deux entités différentes du monde réel (et ainsi elles ne peuvent pas être réconciliées). Une telle hypothèse est valide quand il s'agit d'une source propre (i.e. sans redondances). La LUNA est une hypothèse plus légère que le l'UNA, et déclare que toutes les références associées à une même référence par une relation réfèrent à des entités différentes deux–à–deux du monde réel. Par exemple, à partir des faits $authored(p, a_1)$, ..., $authored(p, a_n)$ provenant de la même source, nous pouvons inférer que les références $a_1, \ldots, a_n$ correspondent à des auteurs distincts de l'article ayant comme référence p.

2.2 Le problème de réconciliation références

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 REC et $NREC$ 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 $\neg Reconcile(i1, i2)$) pour $(i1, i2) \in REC$ (respectivement pour $(i1, i2) \in NREC$). Une méthode de réconciliation est totale si elle produit un résultat ($Reconcile(i1, i2)$ ou $\neg Reconcile(i1, i2)$) 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.

2.3 Méthode logique de réconciliation de références

La méthode de réconciliation logique que nous décrivons dans cette section 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 approche consiste à traduire les axiomes associés au schéma, incluant l'UNA (quand elle s'applique) et la LUNA, par des règles logiques, et à appliquer un algorithme de raisonnement pour inférer des réconciliations et des non réconciliations, ainsi que des synonymies et des non synonymies entre valeurs de base qui seront éventuellement conservées dans un dictionnaire.

Génération des règles de 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 :

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

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).

Pour toute relation R, l'hypothèse LUNA est automatiquement traduite en deux règles $R11(R)$ et $R12(R)$ suivantes:

$R11(R) : R(Z,X) \wedge R(Z,Y) \wedge (X \neq Y) \Rightarrow \neg Reconcile(X,Y)$
$R12(R) : R(X,Z) \wedge R(Y,Z) \wedge (X \neq Y) \Rightarrow \neg Reconcile(X,Y)$

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 :

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

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

  Nous générons automatiquement les règles traduisant les axiomes $PF(P_1,\ldots,P_n)$ de fonctionalité des propriétés et $PFI(P_1,\ldots,P_n)$ de fonctionalité inverse des propriétés composés de plusieurs propriétés. Par exemple, $PF(P_1,\ldots,P_n)$, quand tous les P_i sont des relations, est traduit en la règle :

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

  Si certains P_i sont des attributs, les $Reconcile(X_i,Y_i)$ correspondant doivent être remplacés par $SynVals(X_i,Y_i)$.

  De manière analogue, nous générons des règles traduisant les axiomes $PFI(P_1,\ldots,P_n)$. Quand tous les P_i sont des relations, alors l'axiome est traduit

dans par la règle :

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

– Finalement, pour tout attribut A déclaré comme étant discriminant par l'axiome DISC(A), la règle suivante $R8(A)$ est générée :
R8(A): $\neg SynVals(X, Y) \wedge A(Z, X) \wedge A(W, Y) \Rightarrow \neg Reconcile(Z, W)$

Règle de transitivité : cette règle est générée uniquement dans le cas où l'axiome de l'UNA n'est pas déclaré au niveau des sources de données.

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

Generation de l'ensemble de faits L'ensemble de faits RDF correspondant aux descriptions des données dans les sources S_1 et S_2 est enrichi par un ensemble de faits automatiquement générés :

– de nouveaux faits-classe, faits-relation et faits-attributs obtenus par héritage i.e., en exploitant la relation de subsumption entre les classes et les propriétés, déclarées dans le schéma RDFS: par exemple le fait *ContemporaryMuseum(i)* est présent dans une des sources, le fait-classe $Museum(i)$ et $CulturalPlace(i)$ sont ajoutés à la description de la source;
– des faits de la forme $src1(i)$ et $src2(j)$ pour chaque référence $i \in I_1$ et pour chaque référence $j \in I_2$;
– des faits de synonymie de la forme $SynVals(v_1, v_2)$ pour toute paire (v_1, v_2) de valeurs de base identiques (excepté la ponctuation et les variations de la casse): par exemple, le fait *SynVals("La Joconde", "la joconde")* est ajouté car ces deux valeurs diffèrent uniquement de deux majuscules ;
– des faits de non synonymie de la forme $\neg SynVals(v_1, v_2)$ pour toute paire (v_1, v_2) de valeurs de base distinctes associée a un attribut discriminant. Par exemple, $\neg SynVals(''2004'', '' 2001'')$, $\neg SynVals(''FRANCE'', '' PORTUGAL'')$ sont ajoutés si *Year* et *CountryName* ont été déclarés comme *discriminants*.

Raisonnement. Le raisonnement est appliqué sur l'union $\mathcal{R} \cup \mathcal{F}$ de l'ensemble de règles et de l'ensemble de faits générés automatiquement, comme nous l'avons expliqué précédemment. Le raisonnement doit inférer tous les faits de réconciliation, de non réconciliation, de synonymie et de non synonymie qui sont une conséquence logique de $\mathcal{R} \cup \mathcal{S}$. Le raisonnement est fondé sur la sémantique standard de la logique du premier ordre.

Il est important de remarquer que les règles de reconciliation, bien qu'elles ont des conclusions négatives reste de la forme de clauses de Horn, pour lesquelles ils existe des méthodes de raisonnement qui sont complètes pour l'inférence des impliqués premiers, comme par exemple la resolution SLD [6].

L'algorithme de raisonnement que nous avons implémenté dans notre méthode logique, applique la résolution SLD en trois étapes sur l'ensemble de règles sous la forme de clauses de Horn et sur l'ensemble de faits qui sont des clauses (atomes) closes.

Étape de propositionalisation : nous calculons toutes les résolutions possibles des faits clos dans $\mathcal{F}$ avec des clauses de Horn correspondant au règles dans $\mathcal{R}\backslash\{R9\}$. Cela consist à propager les faits clos dans les règles sauf la règle de transitivité $R9$. Le résultat est un ensemble $\mathcal{P}$ de clauses de Horn complètement instanciées, dans lesquelles les seuls littéraux restant sont de la forme $Reconcile(i,j)$, $\neg Reconcile(i',j')$, $SynVals(u,v)$, ou $\neg SynVals(u',v')$, et chaque atome $Reconcile(i,j)$ ou $SynVals(u,v)$ est considéré comme une variable propositionnelle.

Étape d'inférence propositionnelle : la résolution SLD en propositionel est appliquée sur l'ensemble $\mathcal{P}$ des clauses de Horn *propositionnnelles* obtenues à la première étape.

Étape de transitivité: Cette étape est appliqué seulement si la règle $R9$ est dans $\mathcal{R}$, i.e., sauf si l'axiome de l'UNA n'est pas déclaré au niveau des sources de données. La résolution SLD en logique du premier ordre est appliquée sur les faits clos obtenus à l'étape précédente et sur la forme clausale de la règle de transitivité $R9$.

Il est facile de montrer que tout literal l de la forme $Reconcile(i,j)$, $\neg Reconcile(i',j')$, $SynVals(u,v)$, ou $\neg SynVals(u',v')$: $\mathcal{R} \cup \mathcal{F} \models l$ ssi $\mathcal{P} \models l$

Puisque la résolution SLD appliquée sur des clauses de Horn est complète, et puisque $\mathcal{R} \cup \mathcal{F}$ et $\mathcal{P}$ sont équivalents pour la dérivation des littéraux clos, alors cet algorithme garantit la dérivation de tous les faits de la forme $Reconcile(i,j)$, $\neg Reconcile(i',j')$, $SynVals(u,v)$, ou $\neg SynVals(u',v')$ qui peuvent être logiquement inférés à partir de l'ensemble de règles et de l'ensemble de faits.

D'autres raisonneurs, comme par exemple les raisonneurs de la logique de description, aurait pu être utilisés pour la dérivation des faits de réconciliation. Cependant, les logiques de description ne sont particulièrement pas appropriées pour exprimer certaines règles de reconciliation que nous considérons, qui nécessitent des liaisons explicites de variables. De plus, les raisonneurs des logiques de description existants ne garantissent pas la complétude du calcul d'impliqués premiers.

3 Améliorer l'efficacité de L2R

Dans cette partie nous nous intéresserons à l'efficacité de la tâche de réconciliation de références. Il s'agit d'étudier l'efficacité en espace et en temps de la méthode de réconciliation. Une technique élémentaire pour faire de la réconciliation de référence entre deux ensemble I1 et I2 est d'exécuter une boucle imbriquée de comparaisons de toute référence de I1 avec toutes les références de I2. Malheureusement, une telle stratégie nécessite au total |I1|*|I2| comparaisons. Un autre facteur qui pourrait venir augmenter le coût du calcul de la réconciliation est le coût nécessaire pour une réconciliation.

Nous présentons dans cette section certaines techniques permettant d'améliorer à la fois l'espace et le temps nécessaires pour la réconciliation de références et leurs utilisations possible dans L2R.

3.1 Une meilleure gestion de l'espace

L'espace de réconciliation représente l'ensemble des couples de références qui sont candidats à la réconciliation.

Filtrage. Pour diminuer la taille de l'espace des réconciliations, les méthodes de réconciliations de références peuvent utiliser en pré-traitement des techniques de filtrage. Ces techniques exploitent des connaissances du domaine pour limiter le nombre de couples candidats. Il peut s'agir de :

Méthodes dites de *blocking*. on ne considère que les paires de références qui possèdent une (ou plusieurs) caractéristiques communes, caractéristiques telle que le numéro de ISBN pour les livres, ou encore le nom de famille pour les personnes. Ces techniques ont été introduites par [8] et sont utilisées dans des travaux récents tels que [5]. Si il existe une telle caractéristique qui comporte m valeurs, l'espace des réconciliation est divisé par m.

L'exploitation de connaissances du domaine telles que les disjonctions entre classes. Dans [13] deux références appartenant à deux classes disjointes ne sont pas réconciliables. France telecom nous a fournit un corpus décrivant 562368 hotels. Les disjonctions entre classes d'hôtels de pays différents permettent de réduire l'espace des réconciliations de 67,8%.

L'exploitation de propriétés sur les sources telles que l'Unique Name Assumption. deux références issues d'une source de données qui vérifie l'hypothèse de l'UNA sont forcément distinctes.

Notre méthode L2R [13] de réconciliation a été expérimentée sur le corpus CORA (corpus de citations de publications scientifiques qui a servi de benchmark) qui comporte 6000 références d'articles, de conférences et d'auteurs. L'espace de réconciliation concernant les articles et les conférences contient 2587 références. Nous avons obtenu sur ce corpus un rappel de 64,6% sur l'ensemble REC et de 94,9% sur l'ensemble NREC ce qui nous donne un rappel global de 94,5%. L'espace des réconciliations contient donc 6692569 couples de références à comparer. Utiliser l'année comme propriété discriminante, sachant que les publications existent sur 6 années différentes, permet d'avoir des décisions de non réconciliations pour 21,8% de l'espace de réconciliation. L2R a été également expérimentée sur un jeu de données, fourni par France Télécom R&D, qui comporte sept sources de données contenant des références d'hotels. Dans chacune des sources nous avons l'UNA qui est vérifiée. Nous avons obtenu sur ce corpus un rappel de 54% sur l'ensemble REC et de 75,9% sur l'ensemble NREC ce qui nous donne un rappel global de 75,9%. L'axiom de l'UNA, nous a permis d'inférer des non réconciliations pour 9% de l'espace de réconciliation.

Ces non réconciliations peuvent également être propagées en cours de traitement. Ainsi, si une référence $r1$ d'une source $S1$ a été réconciliée à une référence $r2$ d'une source $S2$ qui possède l'UNA, toutes les autres possibilités de réconciliation de $r1$ avec une référence de $S2$ peuvent être éliminées.

Afin de réduire l'espace de réconciliation, le filtrage peut être effectué à une étape de pré-traitement. Cependant, cela peut empêcher la detection des inconsistances : inférer à la fois $Reconcile(i1, i2)$ et $\neg Reconcile(i1, i2)$, car les paires filtrées ne sont plus considérées par la méthode de réconciliation de références. De plus, nous ne pouvons plus garantir la complétude de notre méthode L2R pour l'inférence des non réconciliations.

Partitionnement des données pour partitioner l'espace de réconciliation. Le partitionnement consiste à diviser *l'espace de réconciliation* en plusieurs sous-ensembles (parties) de taille plus petites.

L'espace de réconciliation est partitionné en plusieurs sous-ensembles de paires de références de taille plus petites de manière à ce que la couverture (le rappel) de la méthode de réconciliation ne soit pas diminuée.

Pour partitionner l'espace des réconciliation, on peut d'abord partitionner les références. Pour assurer la non redondance et la non perte d'information, le partionnement doit satisfaire trois critères (utilisés par [11]):

Complétude: pour toute référence il existe une partition P_i contenant cette référence.
Reconstruction: pour toute source S partitionnée en un ensemble de parties P_i, il existe une opération de reconstruction telle que $S = \Box P_i$ pour tout P_i appartenant à l'ensemble des partitions. Cette opération $\Box$ de reconstruction est à définir en fonction du partitionnement effectué. Par exemple, dans le cas où les partitions sont des composantes connexes d'un graphe, l'opération de reconstruction est la fusion de graphes.
Disjonction: une référence n'est présente que dans une seule partition.

Le graphe G d'un ensemble de références I d'une source de données peut être représenté sous la forme d'un multi-graphe orienté étiqueté dont les sommets V_G sont des références et les arcs E_G sont des relations entre références.

$G = \langle V_G, E_G, R_G \rangle$ où
$V_G = I$, $E_G \subseteq (V_G \times R_G \times V_G)$ et $\quad \langle i1, r, i2 \rangle \in E_G \; ssi \; \exists r, r(i1, i2)$

Considérons que l'ensemble de références est représenté dans un multi-graphe G. Ainsi, l'ensemble des partitions dans l'ensemble de références correspond exactement à l'ensemble de composantes fortement connexes CFC que l'on peut former dans le graphe G. Une composante connexe pour une référence i représente le graphe contenant l'ensemble des références atteignables à partir de i par les relations instanciées.

Une fois que ces CFC ont été définies, elles sont ensuite enrichies par l'ensemble des valeurs atomiques associées aux différentes références par les attributs. Ces derniers sont très importants pour la mesure de la similarité entre les descriptions des références afin de pouvoir décider de leur réconciliation ou de leur non réconciliation. Ainsi, l'espace de réconciliation de référence est réduit à un ensemble d'espaces plus petits correspondant aux paires de composantes fortement connexes du graphe de références G, enrichies par les valeurs des attributs associés à chaque référence des CFCs. Lors de la réconciliation d'une paire de CFCs, l'ensemble des paires de références est le produits cartésien des deux ensembles de références contenues dans les deux CFCs . Bien sûr, certaines paires pourront ne pas être considérées par la suite compte tenu des connaissances du domaine et des informations renseignées.

Nous notons que la taille des composantes connexes est dépendant du niveau de redondance dans les sources (présence de l'UNA). Plus précisément, plus la source est redondante plus les composantes connexes de cette source sont petites. Ainsi, si une source qui décrit un ensemble de références bibliographiques possède l'UNA, chaque

publication est reliée à un ensemble d'auteurs (3 en moyenne) auxquels sont également associées d'autres publications qui font donc parties de la même composante connexe. En revanche, une source telle que CORA ne possède par l'UNA et donc chaque composante connexe se limite à la description d'une publication. CORA qui comporte 6000 références donc peut être découpées en 1295 CFC et donc en 1677025 espaces de 25 références en moyenne.

3.2 Une meilleure gestion du temps

Parallélisation de l'exécution de la réconciliation. Afin d'améliorer les performances en terme de temps d'exécution de la réconciliation, une des solutions est de concevoir un algorithme parallèle. Il s'agit de distribuer la tâche de réconciliation sur un ensemble de processus tout en assurant la non perte d'informations. Pour cela, un ensemble de partitions disjointes peuvent être définies comme présenté précédemment. Dans notre méthode de réconciliation le partitionnement satisfait les contraintes de complétude, de reconstruction et de disjonction, donc il est facile de réaliser le calcul parallèle de la réconciliation de références.

Dans le cas où le partionnement ne peut pas retourner des partitions qui satisfont les trois contraintes, [10] propose d'utiliser deux fonctions qui sont importantes pour la distribution des tâches : la fonction "*scope*" qui permet de distribuer les différentes partitions sur différents processus et la fonction "*responsible*" qui permet d'assurer la non redondance, cela en décidant quel processus est responsable de quelle réconciliation. Une définition plus formelle de ces deux fonctions est donnée dans [10].

Améliorer l'éfficacité d'une réconciliation élémentaire. Traditionnellement, chaque comparison d'une paire de références nécessite la comparaison de l'ensemble des valeurs d'attributs communs et éventuellement la prise en compte des références qui leur sont associées par les relations communes. Avec notre méthode logique de réconciliation nous nous ne occupons pas de cette tâche de calcul de similarité entre les références qui peut être parfois coûteux (méthodes globales). Nous avons uniquement un test d'égalité de valeurs de base des attributs impliqués dans les axiomes du schéma (PF, PFI, DISC). Dans une méthode numérique de réconciliation de références, il est possible de sélectionner un ensemble d'attributs et de relations pour lesquels nous avons une connaissance du domaine exprimant leur pertinence pour le calcul de similarité. Ces connaissances peuvent correspondre aux axiomes de fonctionalité et de fonctionnalité inverse (PF, PFI) des relations et des attributs du schéma RDFS+. Dans [15] on propose un ensemble de techniques pour la réduction de la complexité de la réconciliation d'une paire de références. Une première technique consist à sélectionner un ensemble d'attributs assez pertinents afin de diminuer la dimension en terme d'attributs et de relations des paires de références. Ainsi, ils réduisent le temps de génération du modèle induit qui est un arbre de de décision, utilisé pour classifier les paires de références comme réconciliées ou non réconciliées.

4 Conclusion

Nous avons présenté comment notre méthode logique de réconciliation de références peut s'adapter aux grands volumes de données et à des contraintes strictes de temps.

Nous avons proposé quelques stratégies permettant de limiter la taille des données et le temps de calcul, en vue d'un passage à l'échelle de la réconciliation de références.

Le passage à l'échelle du Web peut s'agir également du fait d'envisager l'application des méthodes de réconciliation dans un contexte pair-à-pair. En d'autres termes, en plus de la distribution et de l'autonomie des sources données, les schémas (ontologies) sont aussi autonome et distribués. Certaines approches travaillent dans ce cadre et estiment que l'utilisation de mappings entre ontologies locales qui se connaissent est plus réaliste que la création d'une ontologie commune importante. Un système tel que *SomeWhere* [4] est une approche complètement pair à pair dans laquelle chaque pair stocke localement ses propres axiomes et un ensemble de mappings. Il exploite les mappings pour répondre de façon correcte et complète à une requête. Une telle approche peut être complétée par une méthode permettant de découvrir et d'exploiter des réconciliations de références afin d'intégrer les données provenant de différents pairs lors d'une requête. [7] proposent une définition des tables de réconciliation (mapping) dans lesquelles sont stockées les références réconciliées entre les différents pairs.

References

1. Owl-dl, http://www.w3.org/2004/owl.
2. Rdfs, http://www.w3.org/tr/rdf-schema/.
3. Swrl, http://www.w3.org/submission/swrl/.
4. Philippe Adjiman, Philippe Chatalic, François Goasdoué, Marie-Christine Rousset, and Laurent Simon. Distributed reasoning in a peer-to-peer setting: Application to the semantic web. *Journal of Artificial Intelligence Research*, 25:269–314, 2006.
5. Rohan Baxter, Peter Christen, and Tim Churches. A comparison of fast blocking methods for record linkage. In *ACM workshop on Data cleaning Record Linkage and Object identification*, 2003.
6. Chin-Liang Chang and Richard Char-Tung Lee. *Symbolic Logic and Mechanical Theorem Proving*. Academic Press, Inc., Orlando, FL, USA, 1997.
7. A. Kementsietsidis, M. Arenas, and R. Miller. Mapping data in peer-topeer systems: Semantics and algorithmic issues, 2003.
8. Howard B. Newcombe and James M. Kennedy. Record linkage: making maximum use of the discriminating power of identifying information. *Commun. ACM*, 5(11):563–566, 1962.
9. Natalya Noy. Semantic integration: a survey on ontology-based approaches. *SIGMOD Record, Special Issue on Semantic Integration*, 2004.
10. Benjelloun Omar, Garcia-Molina Hector, Kawai Hideki, Larson Tait, Menestrina David, and Thavisomboon Sutthipong. D-swoosh: A family of algorithms for generic, distributed entity resolution. In *Technical Report, Stanford InfoLab*, 2006.
11. M. Tamer Ozsu and Patrick Valduriez. *Principles of distributed database systems (2nd ed.)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1999.
12. Erhard Rahm and Philip Bernstein. A survey of approaches to automatic schema matching. *VLDB Journal*, 10:334–350, 2001.
13. Fatiha Sais, Nathalie Pernelle, and Marie-Christine Rousset. Approche logique pour la réconciliation de références. In *Extraction et Gestion des Connaissances conference (EGC 2007)*, 2007.
14. Pavel Shvaiko and Jerome Euzenat. A survey of schema-based matching approaches. *Journal on Data semantics*, 2005.
15. Vassilios S. Verykios, Ahmed K. Elmagarmid, and Elias N. Houstis. Automating the approximate record-matching process. *Information Sciences*, 126(1–4):83–98, 2000.