Découverte de liens d'identité entre instances décrites dans des ontologies partiellement alignées

Nathalie Pernelle, Fatiha Sas, Brigitte Safar, Maria Koutraki, Tushar I. Ghosh

Découverte de liens d'identité entre instances décrites dans des ontologies partiellement alignées

Nathalie Pernelle[1], Fatiha Saïs[1], Brigitte Safar[1], Maria Koutraki[1], Tushar I. Ghosh[1]

LRI, (UNIVERSITÉ PARIS SUD & CNRS UMR 8623) PCRI, bât. 690, 91405 Orsay
[pernelle,sais,safar,ghosh]@lri.fr,Koutraki@ensea.fr

Résumé : Grâce au Linked Open Data, les sources RDF mises à disposition sur le Web sont de plus en plus nombreuses. Différentes approches permettent de découvrir des liens d'identités entre les données décrites dans ces sources en utilisant des règles de liage basées sur les schémas (ou ontologies) auxquelles se conforment les données. Ce type d'approche suppose de disposer, au préalable, de correspondances entre les éléments de ces ontologies. Cependant, ces correspondances peuvent être connues de manière incomplête. Pour lier deux données, nous proposons de calculer leur score de similarité en exploitant à la fois les propriétés pour lesquelles une correspondance existe et celles dont la correspondance est inconnue. Nous illustrons sur un exemple comment la prise en compte de ces dernières permet de proposer plus de liens d'identités.

Mots-clés : Web sémantique, Alignement d'ontologies, Liage de données, RDF/OWL

1 Introduction

Le nombre de sources RDF disponibles sur le Web est en constante augmentation. Un lien d'identité déclaré entre deux données RDF (owl :sameAs) permet d'indiquer que deux descriptions réfèrent à la même entité du monde réel (e.g. même film, même restaurant). L'existence de tels liens permet aux applications de combiner des informations issues de différentes sources. Des outils de liage automatique sont nécessaires quand les données à lier (instances de concepts) sont volumineuses et de nombreuses approches ont été définies dans ce but (Ferrara *et al.* (2011)). Certaines de ces approches sont fondées sur la comparaison des descrip-

tions locales de couples d'instances sans nécessiter de disposer de la description des autres instances. Ces approches (Volz *et al.* (2009); Hassanzadeh *et al.* (2009)) sont particulièrement adaptées à des environnements distribués. D'autres approches (Saïs *et al.* (2009); Herschel *et al.* (2012); Dong *et al.* (2005)) sont fondées sur l'exploitation du graphe complet des instances pour les lier collectivement : une décision de liage peut influencer les décisions de liage d'autres instances. Bien que plus coûteuses, ces approches sont plus informées et obtiennent donc a priori de meilleurs résultats.

La plupart des outils de liage de données s'appuient sur le calcul de scores de similarité. Ils peuvent exploiter les éléments communs des schémas (ou ontologies) décrivant les données ou utiliser des correspondances sémantiques établies entre les éléments de ces schémas. Ces correspondances peuvent avoir été déclarées manuellement par un expert ou découvertes par un outil d'alignement (semi) automatique (Shvaiko & Euzenat (2013)), en particulier quand les ontologies sont de grande taille.

Les outils d'alignement d'ontologies exploitent des informations de différentes natures. Pour aligner des concepts, ces outils peuvent utiliser : (i) des informations terminologiques, comme les labels des concepts, (ii) des informations structurelles induites, par exemple, par la relation de subsomption, (iii) des informations sémantiques comme la disjonction entre concepts et (iv) les similarité des instances des concepts. Pour aligner des propriétés, les outils disposent d'un ensemble d'informations moins riche. En effet, on ne dispose le plus souvent que du label d'une propriété, des types de ses domaines et co-domaines et de quelques liens de sumbsomption. Aussi, pour aligner les propriétés, certaines approches exploitent les similarités des instances des propriétés (Shvaiko & Euzenat (2013)). A l'inverse, pour lier les instances, les outils ont besoin d'exploiter les propriétés communes (ou alignées). Les deux problèmes sont interdépendants ce qui rend difficile de supposer, pour le liage d'instances, que l'on détienne toutes les mises en correspondances entre propriétés. Aussi, nous proposons dans cet article une approche qui permet de calculer des scores de similarité entre instances en exploitant à la fois les propriétés mises en correspondance et celles qui ne le sont pas. En effet, nous pensons que, même non mises en correspondance, les propriétés peuvent être utilisées pour améliorer les résultats d'un outil de liage.

Dans cet article, nous réutilisons un outil de liage existant appelé N2R (Saïs *et al.* (2009)) qui calcule un score de similarité entre les instances pour choisir celles qui peuvent être liées en supposant que toutes les cor-

respondances entre les éléments des deux ontologies sont connues. Quand les correspondants de certaines propriétés n'ont pas été identifiés, nous montrons comment N2R peut être étendu pour s'appuyer malgré tout sur ces propriétés afin d'affiner les scores calculés.

Nous présentons tout d'abord le problème de liage dans un contexte où les données sont conformes à des schémas distincts mais partiellement alignés. Puis nous définissons une mesure, $f_{i_{Nmap}}$, permettant d'évaluer la similarité de deux instances de concepts en exploitant leurs propriétés non mappées mais comparables. Enfin, nous proposons une façon d'intégrer cette nouvelle mesure à celle utilisée par l'outil de liage N2R sur les propriétés mappées et nous illustrons cette approche sur un exemple.

2 Liage de données dans des ontologies partiellement alignées

Soient deux sources de données s_1 et s_2 conformes à deux ontologies OWL O_1 et O_2. Une ontologie O_i est caractérisée par le n-uplet (C_i, H_i, P_i, Ax_i) où C_i est l'ensemble des concepts de O_i, H_i l'ensemble des relations de sumbsomption entre les concepts de O_i, P_i l'ensemble des propriétés qui se décompose en Po_i l'ensemble des relations définies entre les concepts de O_i et Pd_i l'ensemble des attributs des concepts, et enfin Ax_i l'ensemble des axiomes.

Soit A, le résultat d'un processus d'alignement effectué entre O_1 et O_2. On note A_C l'ensemble supposé complet des correspondances identifiées sur les concepts, et A_P, l'ensemble partiel concernant les propriétés. On suppose que les sources de données ont été saturées en utilisant les règles d'inférence du langage OWL (Patel-Schneider *et al.* (2004)). L'objectif est de calculer un score de similarité $sim(i_1, i_2)$ pour tous les couples d'instances tels que i_1 est une instance de $c_1 \in C_1$, i_2 est une instance de $c_2 \in C_2$ et c_1 et c_2 sont $comparables$ au mapping près (i.e., $c_1 \subseteq c2$, $c_1 \equiv c_2$, ...).

N2R est une approche numérique permettant de découvrir des liens d'identités entre deux instances décrites conformément à la même ontologie ou à deux ontologies pour lesquelles les correspondances sont connues. N2R est fondée sur un ensemble d'équations non linéaires qui représentent les influences entre similarités. Elle exploite en particulier les clés déclarées et identifiées comme communes aux deux ontologies, pour inférer des liens d'identité entre instances de concept. Ainsi, si la propriété $aPourCapitale$ est une clé pour le concept $Pays$, la forte similarité de deux instances de villes capitales sera propagée aux instances de pays auxquels ces villes appartiennent. Le système d'équations est résolu par une méthode itérative et les paires d'instances pour lesquelles la similarité ob-

tenue est supérieure à un seuil fixé sont liées, puis soumises à évaluation.

Une telle approche ne se base que sur les propriétés qui ont pu être mises en correspondances. Si les correspondances sont incomplètes, elle ne permet pas de découvrir l'ensemble des données susceptibles d'être liées. De plus, même pour les propriétés mappées, les données elles-mêmes sont souvent incomplètes ou leurs valeurs peuvent être décrites en utilisant un vocabulaire très hétérogène. Aussi, il nous semble important d'essayer d'étendre l'approche afin de prendre en compte des propriétés non mises en correspondance.

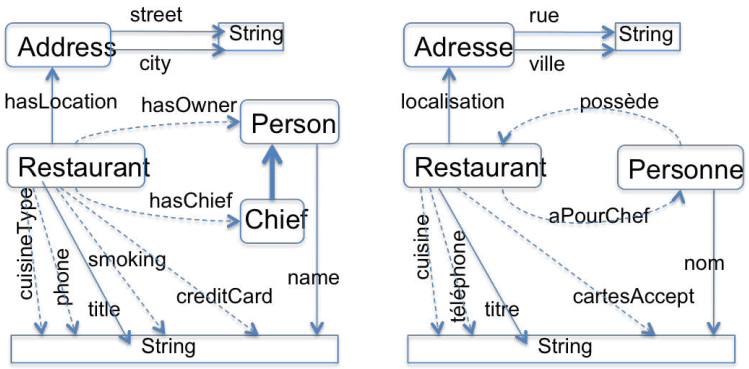

FIGURE 1 – Deux ontologies O_1 et O_2

La Figure 1 présente deux ontologies O_1 et O_2, utilisées pour illustrer notre approche. On suppose que tous les concepts sont mappés (sauf *Chief*), mais que seules les propriétés dénotées avec des traits pleins dans la figure ont des correspondants : A_P = {*hasLocation = localisation, title = titre, street = rue, city = ville, name = nom*} [1]. Les clés de O_1 sont les suivantes : pour le concept *Restaurant*, la clé {*phone*}, pour *Address*, {*street, city*} et {inverse(*hasLocation*)}, pour *Person*, {inverse(*hasOwner*)} et cette dernière se traduit intuitivement par "si deux restaurants sont identiques, ils ont le même propriétaire".

Dans le cas où les sources de données sont conformes à deux ontologies différentes, O_1 et O_2, les seules clés considérées sont celles qui sont communes aux deux ontologies. Nous supposons qu'un ensemble de clés est déclaré dans chacune des ontologies. Nous sélectionnons alors les clés pour lesquelles une correspondance d'équivalence existe pour chacune de ses propriétés. Les clés considérées sont les clés minimales sélectionnées parmi celles obtenues par le produit cartésien des clés de O_1 et O_2.

1. Pour des raisons de lisibilité, nous avons choisi pour les propriétés des labels similaires, même pour celles non mises en correspondance.

Exemple 1. Supposons que les clés déclarées dans O_2 soient $\{telephone\}$ pour le concept $Restaurant$ et $\{rue\}$ et $\{inverse(localisation)\}$ pour le concept $Adresse$. L'ensemble de clés pour lesquelles il existe une correspondance d'équivalence est : dans O_1 $\{street, city\}$, $\{inverse(hasLocation)\}$ et dans O_2, $\{rue\}$ et $\{inverse(localisation)\}$. Le produit cartésien conduit à l'ensemble de clés communes suivant pour le concept $Address$: $\{street, city\}$ et $\{inverse(hasLocation)\}$. Il n'y a pas de clés communes pour les autres concepts à cause de l'incomplétude de l'ensemble de correspondances.

3 Approche N2R-Part

Nous commencons par définir la notion de propriétés comparables, puis nous présentons le calcul des scores de similarité de deux instances exploitant les propriétés non mises en correspondances. Enfin, nous montrons comment l'outil de liage N2R est étendu pour prendre en compte ce score.

3.1 Propriétés comparables

Si une propriété n'a pas de correspondant, nous nous appuyons sur la sémantique de ses arguments (domaine et co-domaine) pour limiter l'ensemble des comparaisons aux seules propriétés de l'ontologie qui auront des arguments comparables (i.e. domaines et co-domaines équivalents ou reliés par des sumbsomptions). Une relation $r_1 \in P_{o1}$ est comparable à une relation $r_2 \in P_{o2}$ (au mapping près) si : $\exists c_{d1}, \exists c_{r1} \in C_1$, $\exists c_{d2}, \exists c_{r2} \in C_2$, tels que Domaine$(r_1, c_{d1})$, Domaine$(r_2, c_{d2})$, CoDomaine$(r_1, c_{r1})$, CoDomaine$(r_2, c_{r2})$ et

(1) ($c_{d1} \subseteq c_{d2}$ ou $c_{d2} \subseteq c_{d1}$) et ($c_{r1} \subseteq c_{r2}$ ou $c_{r2} \subseteq c_{r1}$) ou bien
(2) ($c_{d1} \subseteq c_{r2}$ ou $c_{r2} \subseteq c_{d1}$) et ($c_{r1} \subseteq c_{d2}$ ou $C_{d2} \subseteq c_{r1}$)

La partie (2) de la définition ci-dessus permet de prendre en compte les cas où une propriété de O_1 a été définie de manière inverse dans O_2 tout en étant sémantiquement équivalente, comme les relations *hasOwner* et *possède* dans la figure 1 où *hasOwner* et *hasChief* sont toutes les deux comparables à la fois à *inverse(possède)* et à *aPourChef*.

La notion d'attributs comparables est définie de façon similaire et utilise la hiérarchie des types définis dans XML-Schéma mais se limite au point (1). Ainsi, les 4 attributs du *Restaurant* de O_1 qui sont sans correspondants et ont pour co-domaine des $string$, {*cuisineType, phone, creditCard,*

smoking} sont comparables aux 3 attributs du *Restaurant* de O_2 de même co-domaine {*cuisine, téléphone, cartesAccept*}.

3.2 Similarité de deux instances utilisant les propriétés comparables

A chaque itération, pour chaque couple d'instances (i, j), les valeurs de chaque propriété P_k non mise en correspondance et associée à i, (que i soit *domaine* ou *co-domaine* de P_k), sont comparées à chacune des valeurs des propriétés (ou propriétes inverses) comparables P_l existant sur j. L'objectif est d'identifier les meilleures propriétés comparables, notées $BestP_l$, dont les valeurs ont une forte similarité avec les valeurs de P_k. Puis, les similarités des $BestP_l$ sont agrégées pour calculer une similarité $Sim_{Nmap}(i, j)$ basée sur l'ensemble des propriétés non mises en correspondance.

Nous supposons qu'une mesure de similarité sim a été choisie pour chaque type du XML-Schema : une mesure permet de comparer les chaînes de caractères, une autre les décimaux, etc. La comparaison de deux attributs P_k et P_l se fait ensuite en plusieurs étapes :

(i) On calcule la similarité de chacune des valeurs v_{Pk1}, ..., v_{Pkn} associées à l'instance i par l'intermédiaire de P_k, avec les valeurs v_{Pl1}, ..., v_{Plm} associées à j par l'intermédiaire de P_l en utilisant la mesure de similarité choisie (sim). Pour chaque valeur v_{Pkr} (où $r \in [1..n]$), on retient sa meilleure similarité max_{sim} avec l'une des valeurs de P_l, $max_{sim}(v_{Pkr})$ = $Max_{s\in 1..m}$ (sim(v_{Pkr}, v_{Pls})), si cette similarité est supérieure à un seuil prédéfini.

(ii) Le résultat de la comparaison de deux attributs est représenté par un vecteur $(i, j, P_k, P_l, S_{Sim}, Nb_{VP})$, regroupant les deux instances i, j, les deux attributs P_k, P_l, la somme S_{Sim} des max_{sim} des v_{Pkr} et le nombre maximum d'occurrences des propriétés P_k et P_l (i.e. $Nb_{VP} = Max(n, m)$). Ce résultat n'est retenu parmi les $BestP_l$ que si $S_{Sim} > 0$.

La similarité Sim_{Nmap} des valeurs de tous les meilleurs attributs comparables est ensuite agrégée en tenant compte du nombre d'occurrences des attributs similaires : $Sim_{Nmap}\ (i, j) = \frac{\sum_{BestP_l} S_{Sim}}{\sum_{BestP_l} Nb_{VP}}$

Exemple 2. Soit le couple d'instances de *Restaurant* (i_1, i_2) dont les attributs sans correspondants et comparables ont les valeurs suivantes :

(i_1, *cuisineType*, asian),
(i_1, *cuisineType*, thai),
(i_1, *phone*, 33 68 55 51 58),
(i_1, *phone*, 33 88 82 60 36)
(i_1, *creditCard*, visaCard),
(i_1, *smoking*, only at bar)

(i_2, *cuisine*, asian)
(i_2, *cuisine*, chinese)
(i_2, *cuisine*, thai)
(i_2, *téléphone*, 33 68 55 51 58),
(i_2, *cartesAccept*, AmericanExpress)
(i_2, *cartesAccept*, visaCard)

On suppose, pour simplifier, que la mesure de similarité sim est l'égalité des chaînes de caractères. Pour identifier les $BestP_l$ à associer à *phone*, pour le couple (i_1,i_2), on doit :

(1) calculer la similarité de chacune des valeurs de v_{phone} = {33 68 55 51 58, 33 88 82 60 36} avec les valeurs associées aux différents attributs comparables de i_2. On obtient pour ces deux valeurs une max_{sim} = 0 quand on les compare aux $v_{cuisine}$ et aux $v_{cartesAccept}$ et les valeurs max_{sim}(33 68 55 51 58) = 1 et max_{sim}(33 88 82 60 36) = 0 quand on les compare aux $v_{telephone}$.

(2) Le seul attribut de i_2 ayant une $S_{Sim} > 0$ étant l'attribut *téléphone*, il est retenu comme meilleure propriété comparable à *phone*, avec le vecteur (i_1,i_2, *phone*, *téléphone*, S_{Sim} = 1, nb_{VP} = 2).

Pour les autres attributs de i_1 on retient les vecteurs suivants :

(i_1, i_2, *cuisineType*, *cuisine*, S_{Sim} = 2, Nb_{VP} = 3),
(i_1, i_2, *creditCard*, *cartesAccept*, S_{Sim} = 1, Nb_{VP} = 2).

L'attribut *smoking* n'ayant pas de meilleure propriété comparable, il n'est pas pris en compte dans le calcul. La similarité des valeurs des attributs comparables du couple (i_1, i_2) est donc : Sim_{Nmap} (i_1, i_2) = $\frac{1+2+1}{2+3+2} = \frac{4}{7}$

La similarité des valeurs de relations non mises en correspondance (instances de concepts) est calculée de la même façon. La seule particularité est que la similarité sim de deux instances de concepts évolue au fur et à mesure des propagations comme cela est fait dans N2R.

3.3 Combinaison des similarités

Dans N2R, le score de similarité d'une paire d'instances (i_1, i_2) est représenté par une variable x_i où $i \in [1..n]$ et n est le nombre de paires d'instances sur lesquelles N2R est appliquée. $X = (x_1, x_2, \ldots, x_n)$ est l'ensemble des variables qui correspondent à chacune des paires. Les similarités entre littéraux sont des constantes calculées grâce à des mesures de similarité entre chaînes de caractères (e.g. Levenstein, Jaro-Winckler, etc). Dans le système d'équations, $x_i = f_i(X)$ exprime le fait que la valeur de

x_i dépend des similarités des autres paires d'instances. Chaque équation est de la forme : $f_i(X) = max(f_{i_{Cle}}(X), f_{i_{NC}}(X))$. La fonction $f_{i_{Cle}}(X)$ renvoie le score de similarité maximum obtenu pour les propriétés clés. Cela permet de favoriser la propagation d'un bon score de similarité pour les attributs ou relations clés, à d'autres couples d'instances. La fonction $f_{i_{NC}}(X)$ est une moyenne pondérée des scores de similarité des littéraux ou des instances qui n'appartiennent pas à une contrainte de clé. (voir Saïs *et al.* (2009) pour une définition détaillée de $f_i(X)$).

Nous proposons de combiner le calcul de N2R avec celui réalisé sur les propriétés sans correspondant. Cette combinaison devrait permettre :

- de conserver le fait qu'une forte similarité entre des propriétés clés mises en correspondance entrainent une forte similarité sur les instances. Pour cela, nous gardons l'idée d'utiliser une fonction maximum entre ces similarités ($f_{i_{Cle}}(X)$) et le reste du calcul.

- de donner une influence plus forte à la similarité des propriétés mises en correspondance par rapport à celles qui ne le sont pas (pondération par un coefficient $\alpha \in [0..1]$).

Chaque équation $x_i = f_i(X)$ devient :

$$f_i(X) = max(f_{i_{Cle}}(X), f_{i_{All}}(X) + \alpha \times f_{i_{Nmap}}(X))$$

où la fonction $f_{i_{All}}(X)$ est une moyenne pondérée de tous les scores de similarité des propriétés mises en correspondance et $f_{i_{Nmap}}(X)$ celle des autres. Ces deux fonctions doivent tenir compte du nombre de propriétés existantes dans les schémas et susceptibles d'être comparées. Ce nombre de propriétés (nb_P) est le minimum du nombre de propriétés associables aux concepts les plus spécifiques dont relèvent les deux instances comparées. Dans notre exemple, pour des instances de personne, nb_P vaut 2, pour des restaurants, nb_P vaut 7, et pour les adresses, il vaut 3.

Exemple 3. Soient s_1 et s_2 deux sources contenant les descriptions suivantes en plus de celles présentées dans l'exemple 2.

(i_1, *hasLocation*, a_1)	(i_2, *localisation*, a_2)
(a_1, *street*, 17 rue blainville 75013)	
(a_1, *city*, Paris)	(a_2, *ville*, Paris)
(i_1, *title*, le lotus bleu)	(i_2, *titre*, le lotus bleu)
(i_1, *hasOwner*, p_1)	(p_2, *possède*, i_2)
(p_1, *name*, Chang Lee)	(p_2, *nom*, Chang Lee)

Les trois variables x_A, x_R, x_P représentent respectivement les scores des paires d'instances d'adresses (a_1, a_2), de restaurants (i_1, i_2) et de per-

sonnes (p_1, p_2). Leur score est initialisé à 0 et évolue à chaque itération en fonction de la valeur des variables qui leur sont liées.

Les similarités entre littéraux une fois évaluées sont considérées comme des constantes. Ainsi les constantes a, b et c, toutes les 3 égales à 1, représentent respectivement la similarité des noms de personnes sim(Chang Lee, Chang Lee), de villes, sim(Paris, Paris) et de restaurants, sim(le lotus bleu, le lotus bleu). La constante d = 4/7, représente la similarité des attributs sans correspondants des restaurants calculée précédemment. Avec un coefficient $\alpha = \frac{4}{5}$, les influences de similarité entre les 3 variables x_A, x_R, x_P se représentent par les équations suivantes :

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

Ainsi, la similarité x_A des adresses (a_1, a_2) prend la valeur maximum entre (i) la similarité des deux restaurants x_R (seule clé utilisable) et (ii) la similarité pondérée des deux propriétés mises en correspondance (les villes b et les restaurants x_R). La pondération ($\frac{1}{3}$) correspond au nombre de propriétés susceptibles d'être comparées, $\frac{1}{nb_P}$. Notons que cette équation ne fait pas intervenir de propriété sans correspondant.

La similarité x_R des restaurants (i_1, i_2) est l'aggrégation pondérée par le coefficient $\frac{1}{nb_P}$ ici égal à $\frac{1}{7}$, de : (i) la somme des similarités des valeurs des propriétés mises en correspondance, c pour $name$ et x_A pour $hasLocation$, et (ii) les similarités issues des propriétés comparables non mises en correspondance, i.e. celle des 3 attributs calculée dans l'exemple 2 et celle identifiée entre les relations (own, $hasOwner$).

Le tableau ci-dessous présente les valeurs de similarité des variables après 7 itérations, (au point fixe à 0.001), suivant qu'on utilise N2R-Part uniquement sur les relations avec correspondants (sans $f_{i_{Nmap}}$) ou à la fois sur les relations avec et sans correspondant (avec $f_{i_{Nmap}}$).

Variable	x_R	x_A	x_P
sans $f_{i_{Nmap}}$	0.199	0.399	0.5
avec $f_{i_{Nmap}}$	0.489	0.496	0.695

TABLE 1 – Tests sur les 6 instances

Sans $f_{i_{Nmap}}$, le système ne se base que sur 4 propriétés dont une seule est clé (une partie de l'adresse étant non renseignée pour a_2). De plus la propagation est impossible entre les restaurants et les personnes (propriété sans correspondant). Avec $f_{i_{Nmap}}$, on exploite 4 propriétés supplémentaires

et on rend possible certaines propagations, ce qui ne peut, par construction, qu'augmenter les scores de similarité. Il est clair, qu'à ce jour, toutes les propriétés rapprochées sont utilisées et mises sur le même plan, ce qui est coûteux et pas toujours pertinent (e.g. *creditCard*).

4 Conclusion et perspectives

Nous avons montré sur l'exemple de N2R comment étendre une méthode de liage de données pour prendre en compte des propriétés sans correspondant identifié. Ce premier travail va être expérimenté sur des données réelles et affiné en différenciant les propriétés sans correspondant pour favoriser les propriétés clés ou à fort pouvoir discriminant (i) déclarées dans l'une ou l'autre des ontologies ou (ii) découvertes automatiquement dans un jeu de données.

Références

DONG X., HALEVY A. Y. & MADHAVAN J. (2005). Reference reconciliation in complex information spaces. In *Proceedings of the ACM SIGMOD International Conference on Management of Data, Baltimore, Maryland, USA, June 14-16*, p. 85–96.

FERRARA A., NIKOLOV A. & SCHARFFE F. (2011). Data linking for the semantic web. *Int. J. Semantic Web Inf. Syst.*, **7**(3), 46–76.

HASSANZADEH O., KEMENTSIETSIDIS A., LIM L., MILLER R. J. & WANG M. (2009). A framework for semantic link discovery over relational data. In *Proceedings of the 18th ACM conference on Information and knowledge management*, CIKM '09, p. 1027–1036, New York, NY, USA : ACM.

HERSCHEL M., NAUMANN F., SZOTT S. & TAUBERT M. (2012). Scalable iterative graph duplicate detection. *IEEE Trans. Knowl. Data Eng.*, **24**(11), 2094–2108.

PATEL-SCHNEIDER P. F., HAYES P. & HORROCKS I. (2004). *OWL Web Ontology Language Semantics and Abstract Syntax Section 5. RDF-Compatible Model-Theoretic Semantics*. Rapport interne, W3C.

SAÏS F., PERNELLE N. & ROUSSET M.-C. (2009). Combining a logical and a numerical method for data reconciliation. *J. on Data Semantics*, **12**, 66–94.

SHVAIKO P. & EUZENAT J. (2013). Ontology matching : State of the art and future challenges. *IEEE Trans. Knowl. Data Eng.*, **25**(1), 158–176.

VOLZ J., BIZER C., GAEDKE M. & KOBILAROV G. (2009). Discovering and maintaining links on the web of data. In *Proceedings of the 8th International Semantic Web Conference*, ISWC '09, p. 650–665, Berlin, Heidelberg : Springer-Verlag.