Un jeu de données d'évaluation de correspondances complexes entre ontologies

Elodie Thiéblin, Ollivier Haemmerlé, Nathalie Hernandez, Cassia Trojahn

IRIT-UT2J
Institut de recherche informatique de Toulouse, Toulouse, France
nom.prenom@irit.fr

Résumé : Le web de données liées se compose d'entrepôts de données décrits par des ontologies. Ces ontologies hétérogènes ont différents niveaux de disparité (différences terminologiques, de conceptualisation, de modélisation). Les alignements simples font correspondre une entité de l'ontologie source à une entité de l'ontologie cible. Les alignements complexes, complètent les alignements simples en exprimant plus finement les différents types de disparités. Des approches permettant de détecter des correspondances complexes entre ontologies émergent et il n'existe pas encore de jeu de données exhaustif pour les évaluer. Cet article présente un jeu de données d'alignements complexes entre deux paires d'ontologies du domaine de l'organisation de conférences, ainsi qu'une évaluation d'approches d'alignement permettant d'obtenir de telles correspondances.
Mots-clés : Alignements d'ontologies, correspondances complexes, évaluation.

1 Introduction

Le web de données liées (LOD) est composé de nombreux entrepôts de données. Ces données sont décrites par différents vocabulaires (ou ontologies). La diversité des ontologies utilisées sur le LOD est source d'hétérogénéité. Klein (2001) distingue plusieurs niveaux d'hétérogénéité entre ontologies : les différences de conceptualisation (sur le domaine, la portée des ontologies et leur granularité), terminologiques (termes associés aux entités d'une ontologie) et de modélisation (conventions de modélisation et paradigmes utilisés).

L'alignement d'ontologies (Euzenat & Shvaiko (2013)) est une solution à ces problèmes d'hétérogénéité. On distingue deux catégories d'alignements : les alignements simples (composés de correspondances simples) et les alignements complexes (ayant au moins une correspondance complexe). Les correspondances simples sont limitées car elles lient une à une chaque entité d'ontologies. Les correspondances complexes sont le complément des correspondances simples car elles peuvent mieux prendre en compte les différences de modélisation entre ontologies. Ces différences de modélisation peuvent être répertoriées en *patrons de correspondance*.Les approches de détection de correspondances simples (ou approches d'alignement simple) sont relativement nombreuses et de plus en plus performantes (Achichi *et al.* (2016)). Les approches de détection de correspondances complexes sont moins nombreuses même si de nouvelles propositions voient régulièrement le jour (Qin *et al.* (2007); Ritze *et al.* (2009, 2010); Parundekar *et al.* (2012, 2010); Walshe (2014)). Parmi ces approches, une des plus significatives est basée sur des patrons de correspondance (Scharffe (2009)).

Afin de connaître les atouts et de soulever des pistes d'amélioration d'une approche d'alignement, il est important de l'évaluer.C'est le but de L'OAEI (Ontology Alignment Evaluation Initiative) (Achichi *et al.* (2016)) qui évalue des approches d'alignement sur différents jeux de données. Pour l'instant, ces jeux sont centrés sur les alignements simples.

Dans ce papier, un jeu de données composé de trois ontologies et de correspondances complexes entre deux paires de ces ontologies est proposé. La méthodologie utilisée pour construire ce jeu de données est décrite. Une approche d'alignement basée sur des patrons de correspondance est évaluée sur ce jeu de données. Nous présentons le résultat de l'évaluation. Cette évaluation donne lieu à une discussion sur les types de correspondances complexes et les pistes d'amélioration du jeu de données proposé.

2 Prérequis et travaux liés

2.1 Définitions

Un alignement est un ensemble de correspondances entre une ontologie source $o1$ et une ontologie cible $o2$. On différencie deux types de correspondances :

— Les *correspondances simples* mettent en relation deux entités atomiques de deux ontologies (ici exprimées en logique du premier ordre). Par exemple $\forall x$, *Paper*(x) $\equiv$*Article*(x) est une correspondance simple, *Paper* et *Article* étant deux classes de $o1$ et $o2$, respectivement. On parle de correspondances de cardinalité 1:1 pour désigner les correspondances simples.

— Les *correspondances complexes* permettent d'exprimer des formules logiques entre entités de $o1$ et $o2$. Par exemple $\forall x$, *Accepted_Paper*(x) $\equiv$ $\exists y$,*hasDecision*(x, y) $\wedge$ *Acceptance*(y) est une correspondance complexe car au moins un des membres de l'équivalence est une construction logique d'entités. Les correspondances complexes ont pour cardinalité 1:n, n:1 ou n:m suivant le nombre d'entités et de constructeurs présents de chaque côté de la relation.

La relation d'une correspondance peut être une équivalence ($\equiv$), une relation de subsomption ($\geq$, $\leq$). Un alignement est dit complexe quand au moins une de ses correspondances est complexes. Dans la suite de cet article, les ontologies sont représentées comme suit : les rectangles sont des classes ; une flèche marquée de ($\geq$) entre deux classes marque une relation de subsomption ; un arc marqué ($\perp$) entre deux classes représente leur disjonction ; une flèche verte en trait continu entre deux classes est une propriété sur les objets et les classes qu'elle lie son domaine et co-domaine ; une flèche verte en trait pointillés est une propriété sur les données, elle lie son domaine à son type de données. Entre les entités de deux ontologies, les correspondances simples sont représentées en rouge et la relation de la correspondance est indiquée sur le lien. Les correspondances complexes ne sont pas représentées graphiquement. La figure 1 représente les fragments d'ontologies utilisés dans les correspondances ci-dessus.

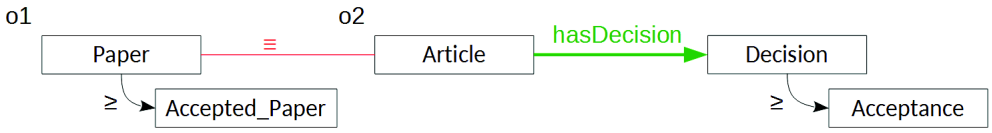


FIGURE 1 – Deux fragments d'ontologies

2.2 Patrons de correspondance

Scharffe (2009) définit les patrons de correspondance comme des solutions pour identifier des types de disparités récurrentes au niveau de la modélisation de deux ontologies. Il propose une librairie de 36 patrons de correspondances qui peuvent être assemblés en patrons composés. Par soucis de place, nous ne présentons que deux patrons représentatifs de ceux utilisés par les

approches. Le patron *Class by Attribute Type* (CAT) décrit les correspondances de la forme $\forall x, A(x) \equiv \exists y, b(x, y) \wedge C(y)$. Autrement dit, la classe A est représentée par une restriction de type C sur l'attribut (propriété sur les objets) b. Par exemple, $\forall x$, *Accepted_Paper(*x*)* $\equiv \exists y$, *hasDecision(*x, y*)* $\wedge$ *Acceptance(*y*)* est une correspondance CAT. Le patron *Class by Attribute Value* (CAV) définit qu'une classe A peut être équivalente à un attribut b dont l'objet est restreint à une valeur donnée v. Une correspondance CAV a pour forme $\forall x, A(x) \equiv b(x, v)$. Par exemple, $\forall x$, *Accepted_Paper(*x*)* $\equiv$ *is_accepted(*x*,true)* est une correspondance CAV.

2.3 Approches d'alignement complexe entre ontologies

Les approches d'alignement complexe entre ontologies émergent. Bien qu'il existe plusieurs approches d'alignement complexe entre schémas, nous faisons la distinction entre ontologie et schéma (XML, base de données, etc.). Les approches d'alignement de schémas sortent du contexte de l'étude. Certaines approches d'alignement d'ontologies se basent sur des patrons de correspondances (Ritze *et al.* (2009, 2010); Parundekar *et al.* (2010, 2012); Walshe (2014)) tandis que d'autres ne les exploitent pas (Qin *et al.* (2007); Nunes *et al.* (2011)).

Ritze *et al.* (2009, 2010) définissent les conditions nécessaires à la détection de correspondances sur la base de certains patrons. Les conditions définies dans ces approches exploitent un alignement simple fourni par l'utilisateur en entrée du processus. Ces conditions portent notamment sur les labels des entités et la structure (taxonomique et propriétés sur les objets) des ontologies. Ces deux approches diffèrent par les indices linguistiques qu'elles considèrent : Ritze *et al.* (2010) utilise des méthodes linguistiques (pré-traitement, exploitation de relations lexicales,etc.) tandis que Ritze *et al.* (2009) utilise uniquement des similarités entre chaînes de caractères. Aucune des deux approches n'utilise ni ne nécessite des instances.

Les autres approches d'alignement requièrent et utilisent des instances communes à deux bases de connaissance décrites par les ontologies à aligner. Parundekar *et al.* (2010, 2012) proposent des approches fondées sur des patrons et des instances communes. Les correspondances détectées sont de type "attribut-valeur = conjonction d'attribut-valeur" ou "attribut-valeur = attribut-ensemble de valeurs". L'espace de recherche est représenté par un arbre puis "élagué" pour trouver la meilleure correspondance. Walshe (2014) propose une approche également fondée sur les patrons et les instances communes. Son approche se concentre sur la détection de correspondances CAV en utilisant des méthodes de sélection d'attributs. Nunes *et al.* (2011) n'utilise pas de patron de correspondance mais utilise les instances pour chercher des combinaisons de propriétés utilisant des fonctions de transformation (concaténation de chaînes de caractères, etc.) par programmation génétique. Qin *et al.* (2007) ne se fonde pas sur les patrons de correspondance. Son approche explore les instances communes à deux bases de connaissances pour en extraire des ensembles de prédicats qui ont un nombre d'occurrences supérieur à un certain seuil.

Une des caractéristiques récurrente dans les approches d'alignement est l'utilisation de patrons de correspondance, ce qui sera exploité dans la suite de ce papier.

2.4 Evaluation des approches d'alignement complexe

Il existe de nombreux jeux de données pour évaluer les approches d'alignement simple. Ces jeux de données ont chacun une particularité suivant le type d'ontologies à aligner : ontolo-

gies volumineuses, sur un domaine particulier (e.g. médical), ontologies légères ou lourdes, etc. L'OAEI (Achichi *et al.*, 2016) utilise certains de ces jeux de données pour évaluer les approches d'alignement simple. Les approches d'alignement complexe citées précédemment ont été évaluées à la main essentiellement sur les sorties qu'elles produisent.

Les approches de Ritze *et al.* (2009, 2010) prennent en entrée les ontologies conférence de l'OAEI et les correspondances complexes générées ont été évaluées en terme de précision. Ritze *et al.* (2009) a aussi été évaluée à partir des ontologies Benchmark de l'OAEI.

Les approches d'alignement proposées par Parundekar *et al.* (2010, 2012) ont trouvé de nombreuses correspondances entre divers entrepôts de données du LOD [1]. Etant donné le grand nombre de correspondances détectées, un sous-ensemble a été évalué à la main. La précision et le rappel ont été calculés sur un sous-ensemble de *GeoNames* et *DBpedia* portant sur les pays. Parmi toutes les correspondances trouvées et correctes, on peut se demander si toutes sont nécessaires. En effet, certaines peuvent notamment être décomposées en correspondances simples. Par exemple (avec *dbo* le préfixe pour l'ontologie de *DBpedia* et *lgdo* celui de l'ontologie de *LinkedGeoData*), $\forall$ *x,y, rdf:type(x,y)* $\wedge$ *dbo:Populated_Place(y)* $\equiv$ *rdf:type(x,y)* $\wedge$ *lgdo:Place(y)* pourrait être remplacée par la correspondance simple $\forall$ *x, dbo:Populated_Place(x)* $\equiv$ *lgdo:Place(x).*

Walshe (2014) crée des ontologies synthétiques (ensembles de classes) à partir d'instances de DBpedia partageant une paire "attribut-valeur" et d'autres classes ne pouvant pas apparaître dans une correspondance CAV avec DBpedia. La précision est évaluée sur les correspondances produites suivant le nombre d'instances communes.

Qin *et al.* (2007) propose les résultats de son approche [2] évalués à la main entre deux bases de connaissances. Parmi les 9 règles proposées en référence, seules deux ne sont pas décomposables en correspondances simples.

Les jeux de données d'alignements complexes existants ne sont pas réutilisables pour toutes les approches d'alignement complexe car ils ont été créés pour (ou par) une approche donnée. À notre connaissance, il n'existe aucun jeu de données d'alignements complexes entre ontologies complet et réutilisable pour évaluer les approches d'alignement complexe.

3 Méthodologie de création de jeu de données d'alignement complexe

Nous décrivons la méthodologie employée pour constituer le jeu de données proposé. Cette méthodologie étant appliquée à la main, elle n'est pas adaptée à des ontologies de grande taille.

Le but de cette méthodologie est de traduire chaque entité de l'ontologie source $o1$ en utilisant les entités de l'ontologie cible $o2$ et d'assurer que toutes ces correspondances sont cohérentes. La méthodologie se focalise donc sur la découverte de correspondances d'équivalence 1:n permettant une certaine exhaustivité (par rapport aux découvertes de correspondances n:m). L'alignement créé se veut exhaustif pour les correspondances 1:n d'équivalence.

1. `http://www.isi.edu/integration/data/UnionAlignments/`
   `http://www.isi.edu/integration/data/LinkedData/`
2. `http://aimlab-server.cs.uoregon.edu/services/ontomap/`

3.1 Étapes

Avant toute chose, il faut déterminer le but de l'alignement. Le type d'alignement ne sera pas forcément le même qu'il serve à fusionner des ontologies ou qu'il serve de médiateur pour de la réécriture de requêtes. Dans le cas de fusion d'ontologies, l'alignement devra être cohérent ; il ne devra pas contenir de correspondances menant à des inférences inexactes. Pour de la réécriture de requêtes, la cohérence d'un alignement est moins cruciale (Euzenat & Le Duc, 2012). Dans le cadre de cet article, nous cherchons à obtenir un alignement cohérent. La méthodologie s'applique en 5 étapes :

1. Obtenir des correspondances simples entre $o1$ et $o2$. Si cet alignement n'existe pas, les approches de l'état de l'art peuvent être réutilisées.
2. Mettre en correspondance une à une chaque entité de $o1$ (classe, relation et propriété) en se basant sur ses labels, axiomes, définition, contexte, instances (s'il y en a) et l'alignement simple. On cherche si possible une équivalence pour l'entité. Si aucune équivalence n'est trouvée, on cherche une relation de subsomption.
3. Vérifier manuellement les conditions de cohérence pour chaque correspondance produite en fonction de la nature de l'entité source et des patrons impliqués dans la correspondance : classe (section 3.2), relation (propriété sur les objets) (section 3.3) ou propriété (propriété sur les données) (section 3.4).
4. Vérifier la cohérence globale de l'alignement en fusionnant les deux ontologies à l'aide des correspondances.
5. Filtrer les correspondances obtenues pour éviter la redondance (section 3.5).

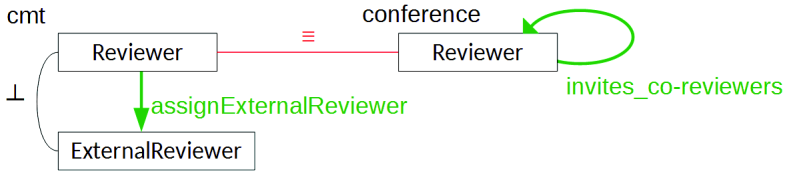


FIGURE 2 – Fragments d'ontologies du dataset

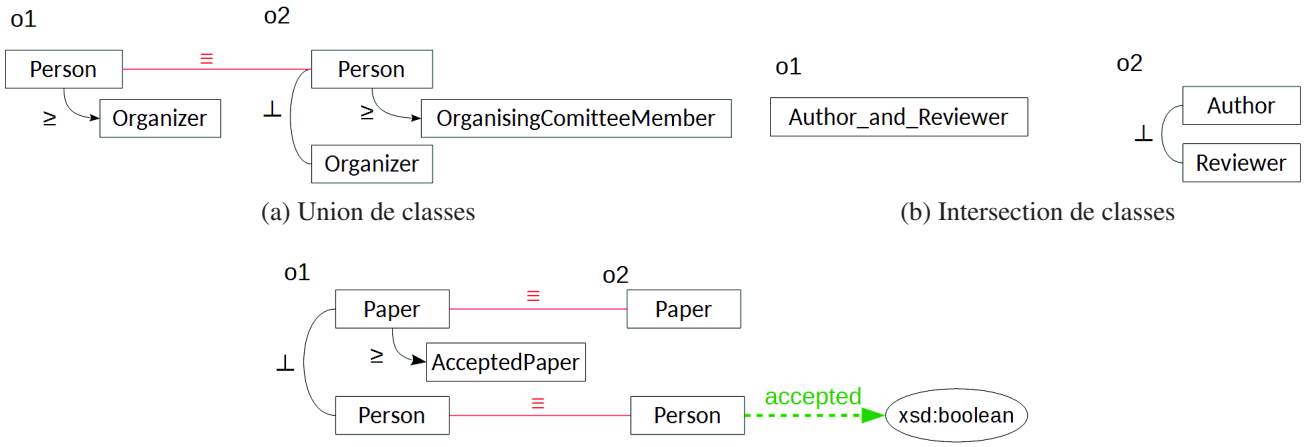
Pour l'étape 3, nous proposons des conditions de cohérence à partir des patrons unitaires de Scharffe (2009). En effet, les patrons permettent de décomposer les correspondances complexes, de les classifier, donc de proposer des conditions de cohérence génériques. Pour assurer la cohérence d'un alignement, il faut s'assurer de la cohérence de toutes ses correspondances individuellement et dans leur ensemble. Individuellement, aucun membre d'une correspondance ne doit représenter un ensemble vide (figure 3b, contraintes de disjonction dans $o2$ seulement : $\forall x$, *o1#Author*(x) $\perp$ *o2#Reviewer*(x)). Considérées dans leur ensemble, les correspondances ne doivent pas se contredire. La disjonction entre deux classes peut venir d'une disjonction couplée à une équivalence : dans la figure 2 la correspondance $\forall x$, *cmt#Reviewer*(x) $\equiv$ *conference#Reviewer*(x) (1) contredit $\forall x, y$, *cmt#assignExternalReviewer*(x, y) $\equiv$ *conference#invites_co-reviewer*(x, y) (2) :

— $(1) \Rightarrow \forall x$, *cmt#ExternalReviewer*$(x)$ $\perp$ *conference#Reviewer*(x)
— $(2) \Rightarrow \forall x$, *cmt#ExternalReviewer*$(x)$ $\equiv$ *conference#Reviewer*(x)

Ces deux correspondances ne peuvent donc pas appartenir au même alignement.

3.2 Correspondances complexes entre classes

Pour chaque classe $c1$ de $o1$ alignée, on vérifie les conditions de cohérence suivantes présentées pour des patrons unitaires. Elles sont toutes à considérer pour une composition de patrons.



(a) Union de classes

(b) Intersection de classes

(c) Restriction de classes

FIGURE 3 – Exemple de patrons de correspondance de classes

Union de classes Pour garantir la cohérence d'une union de classes $c2_i$ avec $c1$, il faut que $c1$ ne soit disjointe avec aucune des $c2_i$. Dans le cas présenté dans la figure 3a, la correspondance $\forall x$, *Organizer(*x*)* $\geq$ *OrganisingCommitteeMember(*x*)* $\vee$ *Organizer(*x*)* n'est pas cohérente car *o1#Organizer* et *o2#Organizer* sont disjointes par extension de la correspondance simple et de la disjonction dans $o2$.

Intersection de classes Dans le cas où $c1$ est mise en correspondance avec une conjonction de classes $c2_i$, $c1$ ne doit pas être disjointe avec aucune des $c2_i$. Les $c2_i$ ne doivent pas non plus être disjointes entre elles. Dans le cas présenté dans la figure 3b, la correspondance $\forall x$, *Author_and_Reviewer(*x*)* $\equiv$ *Author(*x*)* $\wedge$ *Reviewer(*x*)* n'est pas cohérente car *o2#Reviewer* et *o2#Author* sont disjointes, leur intersection vaut donc l'ensemble vide.

Restriction de classes Ces conditions de cohérence sont valables pour les patrons appelés *restrictions de classe* : CAV, CAT, restrictions d'occurrence d'un attribut (CAO). Si $c1$ est mise en correspondance avec une restriction de classe basée sur un attribut (relation ou propriété) $r2$, $c1$ doit être subsumée par le domaine de $r2$, par conséquent, $c1$ ne doit pas être disjointe avec le domaine de $r2$. Dans la figure 3c, la correspondance $\forall x$, *o1#AcceptedPaper(*x*)* $\equiv$ *o2#Paper(*x*)* $\wedge$ *o2#accepted(*x*,true)* n'est pas cohérente. *o1#Paper* et *o2#Person* sont disjointes par extension des correspondances simples et de la disjonction dans $o1$.

3.3 Correspondances complexes entre relations

Pour chaque relation $r1$ de domaine $c1d$ et de co-domaine $c1r$ dans les correspondances 1:n produites, on vérifie les conditions de cohérence suivantes. Si $r1$ est mis en équivalence avec une construction de relation(s) $r2$, le domaine $c2d$ résultant de $r2$ doit être équivalent à $c1d$ et le co-domaine $c2r$ résultant de $r2$ doit être équivalent à $c2r$. Dans le cas d'une subsomption $\forall x, y, r1(x, y) \leq r2(x, y)$, il faut $c1d \subseteq c2d$ et $c1r \subseteq c2r$. Les domaines et co-domaines résultant d'une construction $r2$ suivant le type de patron mis en oeuvre sont les suivants :

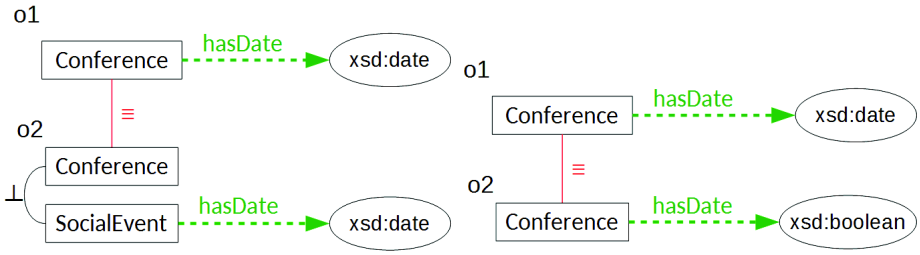
Union de relations $r2$ est une union de relations $r2_i$. $c2d$ vaut l'union des domaines des $r2_i$. $c2r$ vaut l'union des co-domaines des $r2_i$.

Intersection de relations $r2$ est une conjonction de relations $r2_i$. $c2d$ vaut l'intersection des domaines des $r2_i$. $c2r$ vaut l'intersection des co-domaines des $r2_i$.

Inverse d'une relation $r2$ est l'inverse d'une relation $r2_0$. $c2d$ vaut le co-domaine de $r2_0$. $c2r$ vaut le domaine de $r2_0$.

Chaîne de relations $r2$ est une chaîne de relations $r2_i = \{r2_0, ..., r2_n\}$. $c2d$ vaut le domaine de la première relation de la chaîne : $r2_0$. $c2r$ vaut le co-domaine de la dernière relation de la chaîne : $r2_n$. Pour la cohérence d'une chaîne, il faut aussi s'assurer que le co-domaine d'une relation $r2_{j-1}$ ne soit pas disjoint avec le domaine de la relation $r2_j$ suivante dans la chaîne.

3.4 Correspondances complexes entre propriétés sur les données



(a) Incohérence liée au domaine (b) Incohérence de type de données

FIGURE 4 – Fragments d'ontologies ayant des propriétés sur les données

Pour chaque relation $p1$ de domaine $c1$ et de type de données $t1$ mise en correspondance dans l'alignement, on vérifie les conditions de cohérence suivantes. Comme pour les conditions de cohérence sur les relations, si $p1$ est mis en équivalence avec une construction de propriété(s) $p2$, le domaine $c2$ résultant de $p2$ doit être équivalent à $c1$ et le type de données $t2$ résultant de $p2$ doit être compatible avec $t1$. Nous nous basons sur la définition de la compatibilité employée dans Ritze *et al.* (2009) : deux types de données sont compatibles si on peut traduire l'un en l'autre. Dans le cas d'une subsomption $\forall x, y, p1(x, y) \leq p2(x, y)$, il faut $c1 \subseteq c2$ et $t1$ compatible avec $t2$. Pour les patrons d'union, d'intersection et de chaîne, le domaine résultant de la construction $p2$ est le même que pour des patrons de relations. Dans la figure 4a, $\forall x, y$, *o1#hasDate*(x, y) $\equiv$ *o2#hasDate*(x, y) est incohérente car les domaines de ces deux propriétés ne sont pas équivalents (et même disjoints). Dans la figure 4b, $\forall x, y$, *o1#hasDate*(x, y) $\equiv$ *o2#hasDate*(x, y) est incohérente car les types de données sont incompatibles. On distingue deux cas de figure pour vérifier la compatibilité des types de données :

Patrons sans transformation Si $p2$ est une construction impliquant des propriétés $p2_i$ sans fonction de transformation, chaque type de données des $p2_i$ doit être compatible avec $t1$. Cette condition est valable pour les unions, intersections, chaînes de propriétés.

Fonctions de transformation Le type de données résultant d'une fonction de transformation appliquée à des propriétés dépend de cette fonction. Il doit être compatible avec $t1$.

3.5 Filtrage des correspondances

Les correspondances sont ensuite filtrées globalement pour éviter la redondance. Si un alignement simple déclare deux entités $e1$ et $e2$ (de $o1$ et $o2$ respectivement) équivalentes, les

correspondances complexes 1:n ayant $e1$ pour entité source sont supprimées. Par exemple, $\forall x$, *o1#Reviewer(*x*)* $\equiv$ *o2#Reviewer(*x*)* et $\forall x$, *o1#Reviewer(*x*)* $\equiv \exists y$, *o2#reviewes(*x, y*)* sont deux correspondances ayant pour entité source *o1#Reviewer*. On garde la première correspondance et supprime la seconde. Les correspondances dressant une équivalence sont préférées à celle dressant une subsomption (i.e. une correspondance complexe d'équivalence est préférée à une correspondance simple de subsomption) car on peut déduire des subsomptions à partir d'équivalences mais pas l'inverse. Les constructions dans une correspondance complexe sont exprimées de la manière la plus simple possible : le moins d'entité et de constructeurs possibles. Par exemple, on utilisera dans une construction une relation dans son sens direct et non le constructeur inverse appliqué à la relation inverse si cela est possible.

4 Le jeu de données de correspondances complexes

Le jeu de données conférence est utilisé comme banc de test dans l'OAEI (Achichi *et al.* (2016)). Ce jeu de données se compose de 16 ontologies sur le domaine de l'organisation de conférences et d'alignements simples de référence entre 7 de ces ontologies. Ce jeu de données a été choisi car il est fréquemment utilisé dans le domaine de l'alignement d'ontologies (Zamazal & Svátek, 2017). Les ontologies du jeu de données conférence comportent des axiomes, peu d'annotations et présentent plus de réalisme que le jeu de données benchmark synthétiquepar exemple. Parmi les ontologies du jeu de données conférence, trois ontologies formant deux paires ont été utilisées pour créer le jeu de données de correspondances complexes : *cmt-conference* et *cmt-edas*. Leurs alignements de référence enrichis manuellement ont été utilisés dans la méthodologie. Les alignements complexes du jeu de données sont disponibles en ligne [3].

	cmt	conference	edas
Nombre de classes	30	60	104
Nombre de relations	49	46	30
Nombre de propriétés	10	18	20

TABLE 1 – Caractéristiques des trois ontologies du jeu de données.

La table 1 présente le nombre de classes, relations et propriétés des ontologies. La table 2 présente le type (patron) des correspondances dans le jeu de données par paire d'ontologies et par type de l'entité traduite. L'ontologie source est écrite en premier dans la paire (dans *cmt-conference*, *cmt* est l'ontologie source et *conference* l'ontologie cible). Les relations des correspondances ($\equiv$, $\geq$, $\leq$) ne sont pas représentées. La table 3 montre le nombre d'entités de chaque ontologie source traduites par équivalence dans l'ontologie cible. Chaque case a pour forme $(n_s + n_c)/n_t$ où n_s est le nombre d'entités traduites par une équivalence simple, n_c le nombre d'entités traduites par une équivalence complexe et n_t le nombre total d'entité de ce type dans l'ontologie source. La table 4 présente des exemples de correspondances et leur type.

Évaluation du jeu de données

Il n'existe pas encore de méthode automatique permettant d'évaluer la qualité d'alignements

3. https://cloud.irit.fr/index.php/s/JMTMonRBadOzzM2
https://cloud.irit.fr/index.php/s/gJdcRj0PT5fv4Fd

complexes. Toutefois, en s'inspirant de Meilicke & Stuckenschmidt (2008), les ontologies alignées ont été fusionnées à partir des correspondances obtenues dans l'alignement. La consistance de l'ontologie ainsi formée a été vérifiée par le raisonneur HermiT sous Protégé. Cela a aussi permis d'identifier des correspondances incohérentes dans l'alignement de référence de correspondances simples (fig. 2). Ces correspondances ont été retirées de notre jeu de données.

Paire	Classes	Relations	Propriétés
cmt-conference	13 simples, 2 CAT, 1 union	2 dom, 2 range, 2 chaines, 4 dom-range	1 simple,1 union (with string concatenation)
conference-cmt	13 simples, 2 CAT, 2 CAE (composites)	2 unions	1 simple, 1 dom, 2 string split
cmt-edas	9 simples, 1 CAE, 1 union	5 simples, 1 dom, 2 dom-range	1 union (with string concatenation), 1 dom
edas-cmt	9 simples, 2 CAT	5 simples, 4 unions	1 dom, 2 string split

TABLE 2 – Types de correspondances. *dom* représente une restriction de domaine, *range* représente une restriction de co-domaine, *dom-range* une combinaison des deux.

	Classes traduites	Relations traduites	Propriétés traduites
cmt-conference	(13 + 2)/30	(0 + 10)/49	(1 + 1)/10
conference-cmt	(13 + 4)/60	(0 + 0)/46	(1 + 2)/18
cmt-edas	(9 + 1)/30	(5 + 3)/49	(0 + 2)/10
edas-cmt	(9 + 2)/104	(5 + 0)/30	(0 + 3)/20

TABLE 3 – Entités traduites par équivalence (simple + complexe)/nombre d'entités (classes, relations ou propriétés) de l'ontologie $o1$ (alignement $o1$-$o2$)

entité source	rel.	construction cible	type
$\forall$ x, cmt#ConferenceMember(x)	$\equiv$	$\exists$ y, edas#isMemberOf(x,y)	CAT
$\forall$ x, cmt#Chairman(x)	$\geq$	edas#ConferenceChair(x) $\vee$ edas#SessionChair(x)	union
$\forall$ x, edas#AcceptedPaper(x)	$\equiv$	$\exists$ y, cmt#hasDecision(x,y) $\wedge$ cmt#Acceptance(y)	CAT
$\forall$ x, edas#RejectedPaper(x)	$\equiv$	$\exists$ y, cmt#hasDecision(x,y) $\wedge$ cmt#Rejection(y)	CAT
$\forall$ x, conference#Submitted_contribution(x)	$\equiv$	$\exists$ y, cmt#submitPaper(y,x)	CAE (inverse)
$\forall$ x, conference#Reviewed_contribution(x)	$\equiv$	$\exists$ y, cmt#readByReviewer(x,y) $\vee$ cmt#hasDecision(x,y)	CAE (union)
$\forall$ x,y, cmt#email(x,y)	$\equiv$	edas#Person(x) $\wedge$ edas#hasEmail(x,y)	dom
$\forall$ x,y, edas#hasRelatedDocument(x,y)	$\geq$	cmt#writeReview(x,y) $\vee$ cmt#writePaper(x,y)	union
$\forall$ x,y cmt#assignedTo(x,y)	$\equiv$	$\exists$ z, conference#has_a_review(x,z) $\wedge$ conference#has_author(z,y)	chaine relation

TABLE 4 – Quelques correspondances complexes du jeu de données

5 Evaluation d'approches existantes

Parmi les systèmes publiquement disponibles, ceux de Ritze *et al.* (2009, 2010) ne nécessitent pas d'instances communes. Peupler les deux ontologies avec des instances communes

ajoute de nouvelles considérations à prendre en compte pour la création du jeu de données car le choix et la quantité des instances ne sont pas anodins. Pour cette raison, seules les deux approches de Ritze *et al.* sont évaluées sur le jeu de données que nous proposons. Ces approches prennent pour input l'alignement simple de référence et les deux ontologies à aligner.

5.1 Evaluation de Ritze *et al.* (2009)

La première approche proposée par Ritze *et al.* (2009) trouve les correspondances suivantes entre *cmt* et *conference* :

— $\forall x$, *cmt#AuthorNotReviewer(*x*)* $\equiv$ $\exists y$, *conference#contributes(*x, y*)* $\wedge$ *conference#Reviewed_contribution(*y*)*
(Un auteur non-relecteur est une personne qui contribue à un article relu)

— $\forall x$, *cmt#Reviewer(*x*)* $\equiv$ $\exists y$, *conference#contributes(*x, y*)* $\wedge$ *conference#Reviewed_contribution(*y*)*
(Un relecteur est une personne qui contribue à un article relu)

Ces deux correspondances, sans doute obtenues par similarité des chaînes de caractères "Reviewer" et "Reviewed", sont fausses. Le rappel et la précision sont 0 pour *cmt-conference*.

Entre *cmt* et *edas*, les correspondances suivantes sont détectées :

— $\forall x$, *edas#RejectedPaper(*x*)* $\equiv \exists y$, *cmt#hasDecision(*x, y*)* $\wedge$ *cmt#Rejection(*y*)*
(Un papier rejeté est un papier ayant une décision de rejet)

— $\forall x$, *edas#AcceptedPaper(*x*)* $\equiv \exists y$, *cmt#hasDecision(*x, y*)* $\wedge$ *cmt#Acceptance(*y*)*
(Un papier accepté est un papier ayant une décision d'acceptation)

— $\forall x$, *cmt#ConferenceMember(*x*)* $\equiv \exists y$, *edas#hasMember(y,x)* $\wedge$ *edas#Person(*x*)*
(Un membre d'une conférence est une personne qui a été membre d'une conférence)

— $\forall x$, *cmt#AuthorNotReviewer(*x*)* $\equiv \exists y$, *edas#hasRelatedDocument(*x, y*)* $\wedge$ *edas#Review(*y*)*
(Un auteur non-relecteur est une personne qui contribue à une relecture.)

Les trois premières correspondances sont correctes et figurent dans l'alignement *cmt-edas* (resp. *edas-cmt*) proposé. La correspondance $\forall x$, *cmt#ConferenceMember(*x*)* $\equiv$ $\exists y$, *edas#hasMember(y,x)* $\wedge$ *edas#Person(*x*)* n'est pas écrite sous la même forme mais une correspondance équivalente y est : $\forall x$, *cmt#ConferenceMember(*x*)* $\equiv$ $\exists y$, *edas#isMemberOf(*x, y*)*. Cette correspondance est équivalente car *edas#isMemberOf* est la relation inverse de *edas#hasMember*. La précision est 0.75 (3/4) et le rappel (sur l'ensemble des correspondances complexes par équivalence dans les deux sens (*cmt-edas* et *edas-cmt*)) 0.27 (3/11).

5.2 Evaluation de Ritze *et al.* (2010)

L'approche de Ritze *et al.* (2010) à partir de l'alignement de référence n'a pas permis de détecter de correspondances complexes entre *cmt* et *conference*. Même si des correspondances de type CAT sont présentes entre cmt et conference (e.g. $\forall x$, *cmt#ProgramCommitteeMember(*x*)* $\equiv$ $\exists y$, *conference#was_a_member_of(*x, y*)* $\wedge$ *conference#Program_committee(*y*)*), elles sont difficiles à détecter par l'approche de Ritze *et al.* (2010) car les conditions de correspondance sont très restrictives. En effet, le *modificateur* (*ProgramCommittee*) du *substantif principal* (*Member*) n'est pas clairement délimité car *ProgramCommittee* est lui-même un nom composé. D'autre part, le modificateur *ProgramCommittee* n'est pas détecté comme nominalisation de *Program_committee*. Dans l'alignement *conference-cmt*, deux autres CAT auraient dû être

détectés. Toutefois, les restrictions linguistiques trop strictes n'ont pas permis de les identifier. Entre *cmt* et *edas*, Ritze *et al.* (2010) détecte les correspondances suivantes :

— $\forall x$, *edas#RejectedPaper(*x*)* $\equiv \exists y$, *cmt#hasDecision(*x, y*)* $\wedge$ *cmt#Rejection(*y*)*
(Un papier rejeté est un papier ayant une décision de rejet)

— $\forall x$, *edas#AcceptedPaper(*x*)* $\equiv \exists y$, *cmt#hasDecision(*x, y*)* $\wedge$ *cmt#Acceptance(*y*)*
(Un papier accepté est un papier ayant une décision d'acceptation)

Il n'y a pas de correspondance incorrecte. Toutefois, la correspondance $\forall x$, *cmt#ConferenceMember(*x*)* $\equiv$ $\exists y$, *edas#hasMember(y,x)* $\wedge$ *edas#Person(*x*)* n'est plus détectée. La précision est donc de 1 (2/2) et le rappel (sur l'ensemble des correspondances complexes par équivalence dans les deux sens (*cmt-edas* et *edas-cmt*)) de 0.18 (2/11). Les apports linguistiques de cette proposition améliorent la précision de l'approche précédente. Cependant, ils empêchent la détection de certaines correspondances et diminuent le rappel.

6 Discussion

L'analyse du jeu de données permet d'en observer les limites et mettre en évidence ses améliorations futures. Son utilisation pour évaluer des approches d'alignement complexe nous a également permis de mettre en exergue leurs limites. Les approches de Ritze *et al.* contiennent peu de conditions sur les correspondances entre relations et entre propriétés. De plus ces approches ne détectent pas des patrons composés. Si les ontologies étaient peuplées avec des instances communes, l'approche de Nunes *et al.* (2011) aurait pu détecter des combinaisons de propriétés. De même, Qin *et al.* (2007) aurait pu découvrir les chaînes de relations et/ou de propriétés présentes. Les approches de Walshe (2014) et Parundekar *et al.* (2010, 2012) ne découvriraient aucune correspondance du jeu de données car il ne comporte pas de CAV.Parmi les approches d'alignement complexe entre ontologies présentées ici, aucune ne serait en mesure de détecter des correspondances composées (patrons composés). Le jeu de données met en évidence quelques pistes d'amélioration des approches d'alignement complexe. Les axiomes et définitions sont source d'information mais sont peu (ou pas) utilisés par les approches existantes. Les bases de patrons des approches les utilisant peuvent être élargies. Certaines conditions de détection de correspondance par patron (de Ritze *et al.* (2010)) pourraient être complétées ou relâchées. Le jeu de données comporte certains patrons unitaires (CAT, restriction de domaine d'une relation, etc.), d'autres composés. Toutefois, certains patrons de correspondance ne sont pas représentés dans le jeu de données. En effet, aucun CAV n'est présent, de même pour les CAO (restriction de classe par occurrence d'un attribut). Même si on dénombre quelques CAE (restriction de classe par existence d'un attribut), qui sont des cas spéciaux de CAO, ils ne sont pas unitaires dans ce jeu de données mais composés. Aucune intersection n'est présente. Le jeu de données pourrait donc être étendu pour y ajouter d'autres paires d'ontologies dont les correspondances contiendraient des patrons pour l'instant absents. Une autre piste à explorer est le peuplement des ontologies pour permettre aux approches nécessitant des instances d'être évaluées sur ce jeu. Les alignements proposés pourraient aussi être exprimés en EDOAL, une syntaxe dédiée, intégrée à l'Alignment API, pour faciliter son intégration aux outils existants d'évaluation. D'autre part, certains choix de correspondance ont été faits lors de la création du jeu de données (section 3.5). La pertinence de ces choix pourrait être validée par la comparaison des correspondances établies par plusieurs experts qui suivraient la méthode proposée. Un dernier point d'amélioration serait d'ajouter des correspondances n:m au jeu de données.

7 Conclusion

Les alignements complexes sont nécessaires pour pallier l'hétérogénéité entre ontologies. Ils complètent les alignements simples en exprimant plus finement les niveaux de disparités. L'évaluation des approches d'alignement complexe est un besoin croissant. Pour l'instant, leur évaluation est basée sur la précision. Le jeu de données proposé ici est exhaustif pour les correspondances d'équivalence 1:n entre 2 paires d'ontologies issues du jeu de données conférence de l'OAEI. L'évaluation d'approches d'alignement complexe sur le jeu de données permet de mettre en avant leurs limites. Le jeu de données peut être étendu pour servir à l'évaluation de approches utilisant et nécessitant des instances. Les correspondances n:m sont aussi un autre aspect à développer. Il serait intéressant, comme dans Cheatham & Hitzler (2014) de faire valider ce jeu de données par plusieurs experts ainsi que d'explorer la confiance des correspondances.

Références

ACHICHI M., CHEATHAM M., DRAGISIC Z., EUZENAT J., FARIA D., FERRARA A., FLOURIS G., FUNDULAKI I., HARROW I., IVANOVA V. & OTHERS (2016). Results of the Ontology Alignment Evaluation Initiative 2016. In *11th ISWC workshop on ontology matching (OM)*, p. 73–129.

CHEATHAM M. & HITZLER P. (2014). Conference v2. 0 : An uncertain version of the OAEI Conference benchmark. In *International Semantic Web Conference*, p. 33–48 : Springer.

EUZENAT J. & LE DUC C. (2012). Methodological guidelines for matching ontologies. In *Ontology engineering in a networked world*, p. 257–278. Springer.

EUZENAT J. & SHVAIKO P. (2013). *Ontology Matching*. Springer Berlin Heidelberg.

KLEIN M. (2001). Combining and relating ontologies : an analysis of problems and solutions. In *IJCAI-2001 Workshop on ontologies and information sharing*, p. 53–62 : USA.

MEILICKE C. & STUCKENSCHMIDT H. (2008). Incoherence as a basis for measuring the quality of ontology mappings. In *3rd ISWC workshop on ontology matching (OM)*, p. 1–12.

NUNES B. P., MERA A., CASANOVA M. A., BREITMAN K. K. & LEME L. A. P. (2011). Complex Matching of RDF Datatype Properties. In *6th ISWC workshop on ontology matching (OM)*.

PARUNDEKAR R., KNOBLOCK C. A. & AMBITE J. L. (2010). Linking and building ontologies of linked data. In *International Semantic Web Conference*, p. 598–614 : Springer.

PARUNDEKAR R., KNOBLOCK C. A. & AMBITE J. L. (2012). Discovering concept coverings in ontologies of linked data sources. In *International Semantic Web Conference*, p. 427–443 : Springer.

QIN H., DOU D. & LEPENDU P. (2007). Discovering executable semantic mappings between ontologies. In *On the Move to Meaningful Internet Systems*, p. 832–849 : Springer.

RITZE D., MEILICKE C., ŠVÁB ZAMAZAL O. & STUCKENSCHMIDT H. (2009). A pattern-based ontology matching approach for detecting complex correspondences. In *4th ISWC workshop on ontology matching (OM)*, p. 25–36.

RITZE D., VÖLKER J., MEILICKE C. & ŠVÁB ZAMAZAL O. (2010). Linguistic analysis for complex ontology matching. In *5th ISWC workshop on ontology matching (OM)*, p. 1–12.

SCHARFFE F. (2009). *Correspondence Patterns Representation*. PhD thesis, Faculty of Mathematics, Computer Science and University of Innsbruck.

WALSHE B. (2014). *Detecting Restriction Class Correspondences in Linked Open Data*. PhD thesis, Trinity College, Dublin.

ZAMAZAL O. & SVÁTEK V. (2017). The Ten-Year OntoFarm and its Fertilization within the Onto-Sphere. *Web Semantics : Science, Services and Agents on the World Wide Web*, **43**, 46–53.