

Réécriture de patrons de requêtes à l'aide d'alignements d'ontologies

Pascal Gillet¹, Cassia Trojahn¹, Ollivier Haemmerlé¹

IRIT & Université de Toulouse le Mirail, Département de Mathématiques-Informatique,
5 Allées Antonio Machado, F-31058 Toulouse Cedex
pascalgillet@ymail.com, {ctrojahn, ollivier.haemmerle}@univ-tlse2.fr

Résumé : Cet article s'intéresse à l'utilisation d'alignements d'ontologies pour aider la tâche de réécriture automatique de patrons de requêtes. Cette approche s'inscrit dans le contexte du système SWIP, qui permet d'interroger des données RDF à partir de requêtes en langue naturelle, masquant ainsi la complexité du langage SPARQL. SWIP est fondé sur l'utilisation de patrons qui caractérisent des familles de requêtes, et qui sont instanciés en fonction de l'ensemble des mots-clés dans les requêtes initiales en langue naturelle. Cependant, ces patrons sont spécifiques au vocabulaire utilisé pour décrire la source de données à interroger. Nous expérimentons différentes stratégies pour aligner deux ontologies afin d'y trouver des entités correspondantes décrivant deux sources de données différentes. À partir d'un ensemble d'alignements et des patrons de requêtes initiaux, nous réécrivons ces patrons afin de pouvoir interroger les données décrites en utilisant l'ontologie cible. Ces expérimentations ont été menées sur un jeu de données contenant une ontologie du cinéma et d'autres ontologies sur des domaines similaires.
Mots-clés : alignement d'ontologies, patrons de requêtes, SPARQL.

1 Introduction

Si SPARQL est le langage standard *de facto* pour interroger des données RDF, sa complexité restreint son utilisation à grande échelle, notamment pour les utilisateurs non-spécialistes de RDF. La traduction de requêtes exprimées en langue naturelle en requêtes SPARQL fait l'objet de recherches dans les domaines du traitement du langage naturel et du Web sémantique. Dans le système SWIP (Pradel *et al.*, 2012a), les utilisateurs expriment des requêtes avec des phrases en langue naturelle et des patrons de requêtes pré-écrits sont instanciés en fonction de l'analyse syntaxique de la requête initiale. Dans (Elbassuoni *et al.*, 2010; Elbassuoni & Blanco, 2011), les

requêtes sont limitées à des mots-clés, et un ensemble de sous-graphes, correspondant aux mots-clés de la requête, est classé en fonction de modèles statistiques. Dans (Russell & Smart, 2008), les utilisateurs peuvent exprimer leurs requêtes en utilisant un langage de requête visuel, appelé vSPARQL, qui fournit un formalisme graphique pour spécifier des requêtes SPARQL.

Notre travail se concentre sur les approches fondées sur des patrons de requêtes. L'une des principales limitations de ces approches est que la réutilisation des patrons pour différents ensembles de données n'est possible que dans un cadre très limité. Pour chaque source de données à interroger, les patrons de requêtes doivent être (manuellement) construits. Considérant que ces patrons sont définis à partir des ontologies décrivant les sources de données, une stratégie pour automatiser la génération de patrons consiste à exploiter des approches d'alignement d'ontologies. Cette tâche consiste à trouver un alignement (i.e., ensemble de correspondances) entre les entités de deux ontologies (concepts, propriétés ou instances) (Euzenat & Shvaiko, 2007).

Notre objectif est de proposer un mécanisme automatique pour générer des patrons dans le système SWIP. Nous présentons les premières expérimentations menées dans ce sens. À partir d'alignements entre une ontologie source et une ontologie cible, et de patrons de requêtes initiaux disponibles pour l'ontologie source, nous réécrivons ces patrons afin de pouvoir interroger les données décrites à l'aide de l'ontologie cible. Différentes techniques sont utilisées pour aligner les ontologies d'un jeu de données satisfaisant l'ontologie du Cinéma (Pradel *et al.*, 2012b) et d'autres ontologies sur des domaines similaires. Nous utilisons également des patrons de requêtes construits manuellement pour un sous-ensemble de ces ontologies.

L'article est structuré comme suit : en premier lieu, nous présentons les bases de l'alignement d'ontologies et des patrons de requêtes (§2). Nous présentons ensuite les premières expérimentations que nous avons menées pour aligner les différentes ontologies (§3). Nous discutons de nouvelles propositions pour étendre notre approche initiale (§4). Enfin, nous concluons le papier et présentons les perspectives pour des travaux futurs (§5).

2 Contexte

2.1 Alignement d'ontologies

Le processus d'alignement détermine un alignement A' (ensemble de correspondances) pour une paire d'ontologies o et o' . En général, chaque correspondance exprime une relation r entre les entités des deux ontologies, avec une mesure de confiance n . Les correspondances sont définies dans (Euzenat & Shvaiko, 2007) :

Definition 1 (Correspondance simple)

Étant donné deux ontologies, o and o' , une correspondance simple est un quadruple: $\langle e, e', r, n \rangle$, e et e' sont des entités nommées de l'ontologie (classes, propriétés ou instances); r est une relation d'équivalence (\equiv), est plus général (\sqsupseteq), est plus spécifique (\sqsubseteq), ou de disjonction (\perp); et n est un nombre dans l'intervalle $[0, 1]$.

La correspondance $\langle e, e', n, r \rangle$ affirme que la relation r relie les entités e et e' , avec une valeur de confiance n .

Différentes approches d'alignement d'ontologies ont émergé dans la littérature (Rahm & Bernstein, 2001; Kalfoglou & Schorlemmer, 2003; Euzenat & Shvaiko, 2007). Leurs principales différences sont dues à la façon dont la connaissance encodée dans chaque ontologie est utilisée lors de l'identification des correspondances entre les entités ou les structures au sein des ontologies. Des méthodes terminologiques comparent lexicalement les chaînes de caractères (tokens ou n-grammes) utilisées pour nommer les entités (les étiquettes et les commentaires des entités), alors que les méthodes sémantiques utilisent des modèles sémantiques (interprétation logique) afin de déterminer si une correspondance existe entre deux entités. D'autres approches considèrent la structure interne des ontologies, comme l'ensemble des valeurs possibles des propriétés, leur cardinalité, le fait qu'elles soient transitives et/ou symétriques, ou encore leur structure externe, comme la position des entités dans chacune des deux hiérarchies. Les instances (ou extensions) peuvent également être comparées en utilisant par exemple l'intersection des ensembles d'instances. En outre, de nombreux systèmes d'alignement ne s'appuient pas sur une méthode unique.

Les systèmes d'alignement peuvent également être caractérisés selon le type de correspondance qu'ils sont capables de générer. La plupart des systèmes génèrent des correspondances simples entre les entités de deux ontologies, selon une relation d'équivalence.

2.2 Patrons de requêtes

Notre travail s’inscrit dans le cadre des systèmes de Questions-Réponses (QR), qui exploitent des requêtes formulées en langage naturel, et plus particulièrement le système SWIP, qui implémente une approche originale fondée sur des patrons de requêtes (Pradel *et al.*, 2011). Le principe de SWIP repose sur le postulat que, dans les applications réelles, les requêtes formulées sont généralement des variations de quelques familles de requêtes typiques. Par exemple, la famille de requêtes demandant les acteurs qui jouent dans des films. Un autre exemple sont les requêtes demandant les membres d’un groupe musical ou les albums d’un artiste. L’approche de SWIP diffère de celles existantes et vise à améliorer l’efficacité de l’étape de construction des requêtes en utilisant des patrons de requêtes prédéfinis. L’utilisation des patrons évite ainsi d’explorer l’ontologie pour lier les concepts identifiés à partir des mots-clés, puisque les relations potentiellement pertinentes sont déjà présentes dans les patrons. Ainsi, le processus bénéficie des familles de requêtes pré-établies pour lesquelles nous savons qu’un réel besoin en information existe.

Un patron est notamment composé d’un graphe RDF. Par conséquent, un patron est composé d’un ensemble de triplets RDF sous la forme “sujet-prédicat-objet”. En d’autres termes, un patron est un ensemble de sommets et d’arcs représentant respectivement des concepts et des propriétés ; lors de la phase d’instanciation du patron, des parties de ce graphe pourront être affectées de valeurs conformes aux mots-clés de la requête de l’utilisateur. Un aspect important est que le graphe est connexe (i.e., il existe un chemin reliant chaque paire de sommets). La Figure 1 présente un exemple de patron de requêtes dans le format de fichier spécifique à SWIP. Il est composé de 6 triplets RDF dont le sujet *mo : MusicalManifestation* est commun à tous les triplets. *mo : MusicalManifestation*, *foaf : maker*, et *mo : MusicalArtist* sont respectivement le sujet, le prédicat et l’objet du premier triplet. *mo : MusicalManifestation*, *mo : producer*, et *foaf : Agent* sont respectivement le sujet, le prédicat et l’objet du deuxième triplet, et ainsi de suite. Un patron est également décrit par une phrase en langage naturel dans lequel des sous-chaînes indépendantes sont associées à chaque sommet instancié. La phrase modifiée en conformité aux instanciations de sommets sera retournée à l’utilisateur comme une proposition de requête devant approcher au mieux sa requête initiale. Le lecteur pourra se reporter à (Pradel *et al.*, 2011) pour des informations plus détaillées sur le format utilisé pour représenter les patrons de requêtes.

```
pattern cd_info
  query
  [ 1_mo:MusicalManifestation 2_foaf:maker(3) 3_mo:MusicArtist;]creator:0..2/3
  [ 1 6_mo:producer(7) 7_foaf:Agent; ]producer:0..1/7
  [ 1 8_mo:composer(9) 9_foaf:Agent; ]composer:0..1/9
  [ 1 10_mo:singer(11) 11_mo:Performer; ]vocal:0..2/11
  [ 1 12_mo:release_status(13) 13_mo:ReleaseStatus;]status:0..1/13
  [ 1 14_mo:lyricist(15) 15_mo:MusicArtist;]lyricist:0..1/15
  end query
  sentence
  <explanatory sentence here...>
  end sentence
end pattern
```

Figure 1: Exemple de patron sur l'ontologie Music (préfixe mo) représentant les questions sur l'édition/production/publication d'un travail musical.

3 Les premières expérimentations

Notre objectif est d'utiliser des alignements pour la réécriture des patrons de requêtes. Dans cette section, nous présentons les premières expérimentations que nous avons menées pour atteindre cet objectif. Nous présentons brièvement la première version du transformateur de patrons (§3.1), les ontologies que nous avons utilisées (§3.2), et les systèmes d'alignement utilisés pour générer les alignements (§3.3). Les résultats du processus de génération des patrons sont ensuite discutés (§3.4).

3.1 Transformation de patrons de requêtes

Nous avons implémenté un transformateur simple de patrons. Il prend un alignement en entrée, exprimé dans le format d'alignement (David *et al.*, 2011), et transforme un ensemble de patrons de requêtes donné en un autre ensemble de patrons. Pour chaque concept ou propriété e d'un patron donné nous cherchons dans l'alignement une correspondance contenant e . L'entité e' dans cette correspondance est ensuite écrite dans le patron de sortie. Si une entité e source appartient à plusieurs correspondances dans l'alignement, l'outil demande à l'utilisateur de sélectionner l'entité cible e' la plus appropriée (un processus complètement automatique choisirait la correspondance avec la valeur de confiance la plus élevée mais la justification et l'attribution de cette valeur ne sont pas normalisées et sont inconsistantes sur l'ensemble des outils testés). Ce transformateur a été développé en Java, en s'appuyant sur l'API d'Alignement (David *et al.*,

Table 1: Le jeu de données.

Ontologie	Langue	#Classes	#Propriétés (#ObjectProperties + #DataProperties)
Cinéma IRIT ²	Français	198	80 + 22
Movie Ontology ³	Anglais	78	38 + 4
Cinéma JEDFILM ⁴	Anglais	343	70 + 241
Music Ontology ⁵	Anglais	91	126 + 32
DBPedia 3.8 ⁶	Anglais	408	821 + 984
BBC Program ⁷	Anglais	103	128 + 80

2011).

3.2 Ontologies

Nous avons tout d'abord travaillé sur l'ontologie du Cinéma (Pradel *et al.*, 2012b). Cette ontologie a été développée en parallèle du système SWIP, et des patrons de requêtes l'ayant pour cible sont fournis par défaut. Pour expérimenter des techniques d'alignement en utilisant cette ontologie, nous avons récupéré un ensemble d'ontologies sur des domaines similaires¹ (Tableau 1).

Les ontologies Cinéma IRIT et Movie visent à fournir un vocabulaire contrôlé pour décrire sémantiquement des concepts liés aux films, comme Movie, Genre, Director, Actor. Le projet d'ontologie de JEDFILM produit également une ontologie sur le domaine du cinéma, décrivant l'industrie du cinéma, la culture cinématographique, comme le visionnage d'un film, les critiques de films, l'étude d'un film, ou encore la sous-culture propre à un ensemble de fans. L'ontologie Music propose des concepts et des propriétés pour décrire la musique (artistes, albums, chansons, mais aussi spectacles, arrangements, etc.). Le système SWIP fournit également un

¹Certaines de ces ontologies sont disponibles depuis <http://ontologies.alwaysdata.net/cinema/description/>

²<http://ontologies.alwaysdata.net/cinema>

³<http://www.movieontology.org/2010/01/movieontology.ow>

⁴<http://www.jedfilm.com/cinema-ontology>

⁵<http://purl.org/ontology/mo/>

⁶<http://mappings.dbpedia.org/server/ontology/classes/>

⁷<http://purl.org/ontology/po/>

ensemble de patrons de requêtes visant l'ontologie Music. DBpedia, quant à elle, est la seule ontologie dans notre ensemble de données ayant un large éventail d'entités couvrant différents domaines de connaissance (information géographique, personnes, organisations, films, musique, livres, etc.). Enfin, l'ontologie BBC Program fournit un vocabulaire très simple décrivant les programmes audiovisuels et couvre les séries (saisons), les épisodes, des événements de diffusion, les services de diffusion, etc.

Pour expérimenter l'alignement d'ontologies, nous avons créé différents jeux de tests. Un jeu de tests consiste en une paire d'ontologies à aligner. Un alignement est directionnel et se réfère à une ontologie source et une ontologie cible. Dans notre contexte, les patrons de requêtes à transformer sont donc exprimés dans les termes d'une ontologie source. Nous aurions pu essayer toutes les combinaisons possibles des ontologies décrites dans le Tableau 1, mais certaines d'entre elles ne sont pas pertinentes (Cinéma IRIT et Music, par exemple), vu que l'intersection des deux ensembles d'entités est très réduite. Un autre exemple est BBC Program qui est en fait construite sur la base d'autres ontologies telles que Music et FOAF. Aligner les ontologies Music et BBC Program serait donc équivalent à aligner l'ontologie Music avec elle-même.

D'autre part, nous pouvons aligner une ontologie d'un domaine particulier avec une ontologie multi-domaines (Music-DBpedia), et nous attendre à obtenir une bonne couverture (large intersection). Par conséquent, DBpedia est une bonne candidate pour être une ontologie cible. Nous présentons les résultats d'alignements dans la section qui suit.

3.3 Systèmes d'alignement

Nous avons utilisé les systèmes⁸ d'alignement participant à l'*Ontology Alignment Evaluation Initiative* (OAEI) 2012 (Aguirre *et al.*, 2012) ainsi que les systèmes participant aux campagnes précédentes (Tableau 2). Ces systèmes sont pour la plupart libres et accessibles au public. Nous avons listé les seuls systèmes qui étaient en mesure de générer des alignements pour au moins une paire d'ontologies. Plusieurs d'entre eux ont en effet rencontré des difficultés pour analyser les ontologies ou ont généré des exceptions internes. Tous les systèmes ont généré des correspondances simples.

Les meilleurs résultats (étant donné l'objectif de couvrir le plus d'entités

⁸La description détaillée de chaque système ainsi que leur URL sont disponibles sur <http://om2012.ontologymatching.org/>

possible dans l'ontologie source) sont obtenus entre l'ontologie Music et DBpedia. Pour obtenir un ensemble de correspondances unique, nous avons fusionné tous les alignements générés par les systèmes listés dans le Tableau 2, pour chaque paire d'ontologies. L'alignement obtenu est principalement caractérisé par sa cardinalité 1-n, où une entité de l'ontologie source peut correspondre à une ou plusieurs entités de l'ontologie cible. Pour la paire Music-DBpedia, sur 119 correspondances, 97 entités distinctes dans l'ontologie Music (qui en compte 249 au total) ont au moins une correspondance dans DBpedia, ce qui donne une couverture de 39% de l'ontologie source. Pour la paire Cinéma IRIT-DBpedia, seulement 10% des entités sources ont une correspondance dans DBpedia. Nous pouvons noter dans ce cas que le nombre de correspondances est pratiquement le même pour tous les outils. Cinéma IRIT étant en français et DBpedia en anglais, nous nous attendions à ce que les outils d'alignement multilingues surpassent ceux qui ne sont pas conçus dans ce sens ; ce n'est pourtant pas le comportement observé. Mis à part des erreurs internes, ces derniers ont en effet trouvé des correspondances, grâce aux similitudes de structure dans les graphes des ontologies. Enfin, pour les paires Cinéma IRIT-JEDFILM et Cinéma IRIT-Movie, la couverture est réduite à 1%. Sans pouvoir l'affirmer avec certitude (manque de verbosité des outils), nous pensons que cela est dû à des erreurs de parsing de ces ontologies.

L'ontologie Cinéma IRIT n'a pas été alignée auparavant avec d'autres ontologies, et des alignements de référence pour cette ontologie ne sont donc pas disponibles. Le seul ensemble d'alignements de référence que nous avons trouvé contient les correspondances entre les ontologies Music et DBpedia. Ces correspondances ont été réalisées à la main par des experts et ont été initialement utilisées pour tester la qualité des alignements générés par le système BLOOMS⁹. Étonnamment, l'alignement ne contient que des correspondances entre concepts et ignore leurs propriétés. De plus, même si l'équivalence a déjà été établie entre deux concepts, les experts ont identifié toutes les relations de subsomption possibles entre les concepts des deux ontologies (Ex: $Work \equiv Work$, mais également $Work \geq Newspaper$, $Work \geq Book$, $Work \geq Software$, $Work \geq MusicalWork$, etc.). De ce fait, l'alignement de référence contient pléthore de correspondances, mais il s'agit principalement de relations de subsomption et les entités sources sont alignées plusieurs fois (1-n). En raison de cette redondance et du fait que l'alignement ne se concentre que sur les concepts, nous ne sommes pas en mesure d'évaluer la qualité des alignements.

⁹http://knoesis.org/projects/BLOOMS#Resources_for_Download

Table 2: Les systèmes d'alignement utilisés. Les systèmes qui n'ont pas généré d'alignement pour certaines paires d'ontologies sont indiqués par des cellules vides. Le nombre de classes et de propriétés pour chaque ontologie dans une paire est précisé entre parenthèses pour donner une idée du nombre total d'entités à apparier.

	YAM	WeSeE	GOMMA	AUTOMSV2	WikiMatch	Aflood	Falcon	AROMA	GO2A	Hertuda	LogMap	LogMapLt	ServOMapL	Merge	Couverture
Music-Dbpedia (249 * 2213)		31	14		22	58	51			31	8	13		97	39%
Cinéma-Dbpedia (300 * 2213)	13		13	6				12	17		17	14	12	31	10%
Cinéma-Jedfilm (300 * 654)			4											4	1%
Cinéma-Movie (300 * 120)			1											1	1%

ments générés conformément à l'alignement de référence: le calcul du rappel et de la précision s'en trouveraient faussés dès le départ.

3.4 Résultats de la transformation de patrons

SWIP offre deux ensembles de patrons de requêtes : un pour l'ontologie Music et un autre pour l'ontologie Cinéma IRIT. À partir de l'alignement Music-DBPedia (Tableau 2), nous avons d'abord transformé les patrons de requêtes pour l'ontologie Music en de nouveaux patrons pour DBPedia. Nous avons obtenu le résultat suivant : 25 entités sur 60 ont été remplacées, soit 41%. Sur un total de 5 patrons, contenant au total 27 sous-patrons, seulement 2 ont été traduits complètement.

Concernant les patrons pour Cinéma IRIT, quel que soit l'alignement utilisé avec cette ontologie comme source, la couverture de la transformation est proche de zéro. Cela signifie que non seulement les alignements sont de mauvaise qualité, mais aussi que, pour le meilleur d'entre eux, les entités trouvées ne sont pas celles utilisées dans les patrons.

Quel que soit le système d'alignement utilisé, l'ontologie source n'est jamais complètement couverte, et certaines entités peuvent ne pas avoir d'équivalent dans l'ontologie cible. Ce résultat était prévisible et met en

évidence le problème de la complétude du processus de transformation. Nous avons néanmoins besoin d'évaluer très précisément cette insuffisance.

Nous pensons que les correspondances simples permettraient de traduire une grande partie d'un patron de requêtes. Nous aurions ensuite appliqué d'autres méthodes plus sophistiquées (ou au contraire des méthodes dites de force brute) pour transformer la partie restante. Par exemple, si une propriété de l'ontologie source ne peut pas être déterminée dans la cible alors que nous avons trouvé des concepts équivalents pour les deux extrémités de la relation, nous pourrions explorer tous les rôles possibles entre les deux classes correspondantes.

D'autre part, le degré de recouvrement (intersection) entre les ontologies dépend avant tout de leur domaine. En général, on suppose que les deux ontologies à aligner décrivent le même domaine, sinon elles ne se recouvrent pas ou à peine. En outre, les systèmes d'alignement ne doivent pas faire en sorte de trouver une correspondance pour chaque élément des deux ontologies, et ignorer les éléments qui ne sont pas pertinents à la place. En conséquence, l'incomplétude des alignements est tout à fait normale et nous devons savoir la gérer.

Outre le problème d'incomplétude des alignements, la plupart des systèmes produisent des correspondances de qualité : bien qu'elles ne soient pas nombreuses, les correspondances trouvées se révèlent en revanche vraies (selon nos propres jugements de pertinence). Mais les outils se limitent à la détection de correspondances simples, avec des relations d'équivalence ou de subsomption entre des entités uniques. Et les correspondances simples ne sont souvent pas suffisantes pour représenter correctement la relation entre les entités alignées.

Il y a aussi le problème majeur de la granularité des correspondances: remplacer les entités une par une dans un patron de requêtes ne permet pas d'obtenir la sémantique attendue résidant dans le graphe dans son ensemble. Notamment, le choix d'une correspondance plutôt qu'une autre avec la même classe source devra se faire en considérant les propriétés utilisées sur cette classe. Nous avons besoin de changer d'échelle, et nous devons être en mesure d'exprimer tout sous-ensemble du graphe. La sémantique réelle d'une classe, et donc sa correspondance, se révèle dans ses relations avec les autres classes. Cependant, les alignements atomiques et homogènes ont une sémantique bien définie, et sont les blocs de construction élémentaires pour obtenir des alignements plus riches, comme nous allons en discuter dans la section suivante.

4 Vers les correspondances complexes

Nos expérimentations initiales ont montré que les correspondances simples, i.e., exprimant des relations entre des entités atomiques, ne sont pas suffisantes pour décrire la relation entre les entités connexes dans diverses ontologies. Des correspondances complexes, comme par exemple sur les vins dont le terroir est situé dans la région "Aquitaine": $\text{Wine} \wedge \text{hasTerroir-locatedIn} = \text{"Aquitaine"} \geq \text{Bordeaux}$, sont donc nécessaires (Walshe, 2012). Nous envisageons d'utiliser ce type de correspondances pour la réécriture de patrons de requêtes. Cette approche est utilisée, par exemple, pour la réécriture de requêtes SPARQL (Correndo & Shadbolt, 2011).

En s'appuyant sur les constructeurs de la Logique de Description (DL) et les alignements simples entre les ontologies, nous avons construit à la main des alignements complexes qui relient l'ontologie Cinéma IRIT et l'ontologie DBpedia. Ces alignements sont présentés dans le Tableau 3. Par exemple, le concept *FilmLocalise* dans Cinéma IRIT est lié au concept complexe $\text{Film} \sqcap (\exists \text{basedOn.Film} \sqcup \exists \text{previousWork.Film})$ dans DBpedia, suivant une relation de subsomption. Pour la création des alignements, nous avons découpé les patrons de requêtes initiaux en 51 sous-graphes, et nous avons pu lier 49 d'entre eux à des concepts et des rôles de DBpedia (96% de couverture). Par exemple, le concept d'acteur de complément (figurant) dans Cinéma IRIT n'a pas d'équivalent dans DBpedia.

De plus, DBpedia ne fournit pas systématiquement des relations inverses, et nous devons parfois introduire l'inverse d'un rôle R , noté R^- (correspondance 3 dans le Tableau 3). Un rôle se traduit souvent par une composition de rôles (conjonction et/ou disjonction) dans l'ontologie cible. Notez que la disjonction n'est actuellement pas prise en charge dans les patrons de requêtes. Un cas intéressant est celui de la correspondance 7 : cette correspondance ne peut être qu'une subsomption car l'artiste qui remporte le prix est nécessairement l'un des nominés, alors que le nominé n'est que le bénéficiaire éventuel d'une récompense. Ce cas illustre la plus-value sémantique des correspondances complexes.

5 Conclusion

Dans cet article, nous avons présenté les expérimentations que nous avons menées sur l'utilisation des méthodes d'alignement d'ontologies pour aider la tâche de génération de patrons de requêtes. Nous avons observé que les

Table 3: Alignement complexes entre les ontologies Cinéma et DBPedia.

#id	Rel	Correspondance
1	\equiv	cinema:Artiste \sqcap \exists estRecompenseA.CesarDuCinema dbpedia:Artist \sqcap \exists cesarAward(Award \sqcap \exists event.FilmFestival)
2	\leq	cinema:Artiste \sqcap \exists contribueArtistiquementAOeuvre.Oeuvre dbpedia:Person \sqcap \exists associate ⁻ .Work
3	\equiv	cinema:Artiste \sqcap \exists acteurDoublePar.Doubleur dbpedia:Person \sqcap \exists dubber ⁻ .Artist
4	\equiv	cinema:Artiste \sqcap \exists incarne.Role dbpedia:Person \sqcap \exists portraye ⁻ .FictionalCharacter
5	\leq	cinema:FilmLocalise dbpedia:Film \sqcap (\exists basedOn.Film \sqcup \exists previousWork.Film)
6	\leq	cinema:Artiste \sqcap \exists estRecompensePour.Film dbpedia:Artist \sqcap \exists award(Award \sqcap \exists related.Film)
7	\leq	cinema:Artiste \sqcap \exists estRecompensePour.Film dbpedia:Artiste \sqcap \exists nominee ⁻ .Artist
8	\leq	cinema:Artiste \sqcap \exists estMembreDeJuryPourCompetition.Competition dbpedia:Person \sqcap \exists organisationMember ⁻ (Organisation \sqcap \exists event.FilmFestival)
9	\leq	cinema:LongMetrageNomine \sqcap \exists concourtPourCompetition.Competition dbpedia:Film \sqcap \exists film ⁻ .FilmFestival
10	\equiv	cinema:Film \sqcap \exists aPourRole.Role dbpedia:Film \sqcap \exists starring(Person \sqcap \exists portraye ⁻ .FictionalCharacter)

correspondances simples ne suffisent pas pour l'automatisation complète de cette tâche. Cependant, la plupart des systèmes d'alignement disponibles ne sont pas en mesure de générer ce type de correspondances.

Nous travaillons sur la génération de patrons de requêtes, à partir des alignements complexes construits manuellement et présentés dans ce papier. Nous envisageons également d'étudier les approches existant dans la littérature pour la génération automatique de correspondances complexes. Une première approche consiste à utiliser des patrons de correspondances complexes et les similitudes au niveau ABox des ontologies (instances). Cela est particulièrement pertinent lorsqu'il s'agit de sources de données issues du Web de données liées.

References

AGUIRRE J. L., ECKERT K., EUZENAT J., FERRARA A., VAN HAGE W. R., HOLLINK L., MEILICKE C., NIKOLOV A., RITZE D., SCHARFFE F., SHVAIKO P., STUCKENSCHMIDT H., SVÁB-ZAMAZAL O., TROJAHN C., JIMÉNEZ-RUIZ E., GRAU B. C. & ZAPILKO B. (2012). Results of the on-

- tology alignment evaluation initiative 2012. In *Proceedings of the 7th International Workshop on Ontology Matching*.
- CORRENDO G. & SHADBOLT N. (2011). Translating expressive ontology mappings into rewriting rules to implement query rewriting. In *Proceedings of the 6th International Workshop on Ontology Matching*.
- DAVID J., EUZENAT J., SCHARFFE F. & TROJAHN C. (2011). The alignment api 4.0. *Semantic Web*, **2**(1), 3–10.
- ELBASSUONI S. & BLANCO R. (2011). Keyword search over rdf graphs. In *Proceedings of the 20th ACM international conference on Information and knowledge management, CIKM '11*, p. 237–242: ACM.
- ELBASSUONI S., RAMANATH M., SCHENKEL R. & WEIKUM G. (2010). Searching rdf graphs with sparql and keywords. *IEEE Data Eng. Bull.*, **33**(1), 16–24.
- EUZENAT J. & SHVAIKO P. (2007). *Ontology Matching*. Berlin, Heidelberg: Springer-Verlag.
- KALFOGLOU Y. & SCHORLEMMER M. (2003). Ontology mapping: the state of the art. *Knowl. Eng. Rev.*, **18**(1), 1–31.
- PRADEL C., HAEMMERLÉ O. & HERNANDEZ N. (2011). Expressing conceptual graph queries from patterns: How to take into account the relations. In *Conceptual Structures for Discovering Knowledge*, Lecture Notes in Computer Science, p. 229–242. Springer Berlin Heidelberg.
- PRADEL C., HAEMMERLÉ O. & HERNANDEZ N. (2012a). A semantic web interface using patterns: The swip system. In *Graph Structures for Knowledge Representation and Reasoning*, Lecture Notes in Computer Science, p. 172–187. Springer Berlin Heidelberg.
- PRADEL C., HERNANDEZ N., KAMEL M. & ROTHENBURGER B. (2012b). Une ontologie du cinéma pour évaluer les applications du web sémantique. In *Atelier Ontologies et Jeux de Données pour évaluer le web sémantique, IC*.
- RAHM E. & BERNSTEIN P. A. (2001). A survey of approaches to automatic schema matching. *The VLDB Journal*, **10**(4), 334–350.
- RUSSELL A. & SMART P. R. (2008). Nitelight: A graphical editor for sparql queries. In *Proceedings of the Poster and Demonstration Session at the 7th International Semantic Web Conference (ISWC2008)*.
- WALSHE B. (2012). Identifying complex semantic matches. In *Proceedings of the 9th international conference on The Semantic Web: research and applications*, p. 849–853: Springer-Verlag.