Détection et Représentation des changements dans les sources de données RDF

Daniel Mercier[1], Nathalie Pernelle[1], Fatiha Saïs[1], Sujeeban Thuraisamy[1]

UNIVERSITÉ PARIS SUD, LABORATOIRE DE RECHERCHE EN INFORMATIQUE
91405 Orsay cedex, France
{pernelle, sais}@lri.fr, {DanielMercier,Sujeeban.Thuraisamy}@u-psud.fr

Résumé : De nombreuses sources de données RDF sont en évolution constante que ce soit au niveau des données ou du vocabulaire utilisé. Or, de nombreuses tâches d'intégration sont impactées par ces modifications. Nous présentons une approche permettant de détecter et de représenter des changements plus ou moins complexes que l'on peut détecter lorsque l'on s'intéresse aux seules données. Une première expérimentation a été menée sur différentes versions de DBPedia.

Mots-clés : Ontologies, Représentation des connaissances, Evolution des données

1 Introduction

Le Web des données est en évolution permanente. De nombreuses modifications sont apportées quotidiennement sur les données et les vocabulaires publiés sur le LOD. En 2012, 76% des documents RDF du LOD ont subi au moins une modification (Käfer *et al.* (2012)). Ces modifications peuvent intervenir au niveau ontologique (classes, propriétés, axiomes et correspondances avec les autres ontologies) ou/et au niveau données (entités, valeurs des propriétés et liens d'identité entre entités). De nombreuses tâches d'intégration de données sont impactées par ces modifications (e.g synchronisaton, liage, fusion de données). Dans ce contexte, il est important de disposer d'outils permettant de détecter et de représenter ces changements de façon à ce que les tâches impactées puissent mettre à jour leurs résultats sans devoir être ré-exécutées sur toutes les données. De nombreux travaux se sont focalisés sur la détection, la représentation et la gestion des changements au niveau ontologique (Zablith *et al.* (2015); Dinh *et al.* (2014)). Quelques travaux récents (Papavasileiou *et al.* (2013); Roussakis *et al.* (2015)) se sont intéressés au problème de détection de changement dans les données. Cependant, ces derniers ne sont pas représentés dans une ontologie ou sont synthétiques et ne gardent pas trace des triplets impliqués. Nous proposons une approche permettant de détecter et de représenter sémantiquement des changements plus ou moins complexes que l'on peut identifier lorsque l'on s'intéresse aux seules données.

2 Approche de détection et de représentation de l'évolution de sources de données RDF

Etant données deux versions TDB_{old} et TDB_{new} d'une source de données, nous détectons tout d'abord les ensembles de de triplets ajoutés et supprimés. Ensuite, une étape de représentation sémantique des changements est réalisée. Pour cela, nous avons défini une ontologie appelée O^{DE} qui permet de représenter sémantiquement les types de changements. Deux types généraux de changements sont distingués *AddedStatement* et *DeletedStatement* pour lesquels

quatre sous-types sont définis. Pour le type *AddedStatement* nous avons défini [1] :
-AddedSchemaElement qui représente les assertions impliquées dans l'introduction d'une nouvelle propriété (sous-type *AddedProperty*) ou d'une nouvelle classe (sous-type *AddedClass*).
-AddedTyping qui regroupe les assertions concernant l'introduction de nouveaux types pour des instances existantes. Parmi ces assertions, certaines concernent la première utilisation d'une classe dans la source de données (sous-type*AddedClass*).
-AddedInstance qui représente l'ensemble des assertions décrivant des instances nouvelles.
InstanceDescriptionEnrichment qui décrit l'ensemble des assertions qui viennent enrichir des instances existantes.
Chaque triplet supprimé ou ajouté est réifié et automatiquement associé à un ou plusieurs types de changements définis dans O^{DE}. L'ontologie peuplée peut ensuite être exploitée pour répondre à des requêtes d'experts simples (e.g. le nombre d'instances supprimées) ou plus complexes (e.g. les propriétés fonctionnelles dont la valeur a été modifiée).

TABLE 1 – Requêtes simples exploitant la classe DeletedClass

/* Liste des classes supprimées

```
SELECT DISTINCT ?deletedClass
WHERE {
?node rdf:type   ode:DeletedClass
?node rdf:object ?deletedClass .
}
```

/* Instances conduisant à une suppression de classe

```
SELECT ?s
WHERE {
?node rdf:type ode:DeletedClass .
?node rdf:subject ?s .
}
```

3 Premières experimentations

Nous avons évalué notre approche en utilisant les versions 3.5, 3.8 et 3.9 de DBPedia [2]. Nous nous sommes intéressés à la classe $Person$ (fichier PersonData). Le tableau 2 présente le nombre de triplets, le nombre d'instances, le nombre de propriétés ainsi que le nombre de types différents pour les trois différentes versions de cette classe.

TABLE 2 – Evolution des triplets décrivant les Personnes dans trois versions de DBPedia

	Version 3.5	Version 3.8	Version 3.9
#triplets	482 080	18 719 429	22 008 122
#instances	48 692	2 853 529	3 733 629
#propriétés	9	9	9
#types	71	348	434

Nous avons détecté les changements élémentaires entre deux versions successives de la classe $Person$ et nous avons peuplé l'ontologie O_{DE} (temps d'exécution sur plus de 18 millions de triplets : 55 mn, 155 millions de triplets après réification). Presque toutes ces assertions sont de type $AddedStatement$ (tab. 3), mais près de la moitié des triplets existants ont été supprimés

1. Les sous-classes de *DeletedStatement* peuvent être décrites de façon symétrique.
2. http ://dbpedia.org/services-resources/datasets

(classe $DeletedStatement$). La table 3 montre par exemple comment les assertions instances de la classe $AddedStatement$ se répartissent dans les différentes classes de l'ontologie O_{DE}.

TABLE 3 – Type et nombre de changements pour la classe $AddedStatement$

	#Added Statements	#Added SchemaElement	#Added Property	#Added Class	#Added Typing	#Added Instance
v3.5 -> v3.8	18 469 394 (12 :43 mn)	284 (5 :16 mn)	5 (3 :28 mn)	279 (1 :48 mn)	13 596 447 (6 :43 mn)	2 835 666 (7 :20 mn)
v3.8 ->v3.9	4 813 958 (01 :49 mn)	86 (14 s)	0 (2 s)	86 (12 s)	4 015 870 (1 :21 mn)	1 103 520 (45 s)

Les classes $addedTyping$ et $deletedtyping$ montrent que le typage des instances évolue mais l'approche permet également de détecter que les éléments de l'ontologie utilisés ont évolué. Ainsi, entre v3.5 et v3.8, 284 classes sont apparues dans la description des personnes et deux ont disparu. Nous avons exécuté différentes requêtes permettant par exemple d'utiliser O_{DE} pour lister les nouvelles instances d'une sous-classe donnée ou pour rechercher les assertions de type $InstanceDescriptionEnrichment$ qui ont été ajoutées pour une URI donnée. Une requête simple permet ainsi de montrer que 7 assertions ont été ajoutées pour Barak Obama (e.g. la propriété $description$ prend la valeur "American politician, 44th President of the United States"@en dans v3.8) et que cette URI est nouvellement typée par les classes $Agent$ et $OfficeHolder$.

4 Conclusion

Nous proposons une approche originale de détection et de représentation sémantique des changements dans une source RDF. Nous avons conçu une ontologie O^{DE} qui représente sémantiquement les différents types de changements pouvant survenir au niveau des données et qui est automatiquement peuplée par notre approche. Cela permet à un expert ou à un outil d'interroger l'ontologie via des requêtes SPARQL pour observer l'évolution des données. Nous souhaitons maintenant détecter d'autre types de changements (changements induits par des sources externes via des liens $SameAs$, axiomes).

Références

DINH D., DOS REIS J. C., PRUSKI C., SILVEIRA M. D. & REYNAUD-DELAÎTRE C. (2014). Identifying relevant concept attributes to support mapping maintenance under ontology evolution. *J. Web Sem.*, **29**, 53–66.

KÄFER T., UMBRICH J., HOGAN A. & POLLERES A. (2012). Dyldo : Towards a dynamic linked data observatory. In *WWW2012 Workshop on Linked Data on the Web, Lyon, France, 16 April, 2012*.

PAPAVASILEIOU V., FLOURIS G., FUNDULAKI I., KOTZINOS D. & CHRISTOPHIDES V. (2013). High-level change detection in rdf(s) kbs. *ACM Trans. Database Syst.*, **38**(1), 1 :1–1 :42.

ROUSSAKIS Y., CHRYSAKIS I., STEFANIDIS K., FLOURIS G. & STAVRAKAS Y. (2015). A flexible framework for understanding the dynamics of evolving RDF datasets. In *The Semantic Web - ISWC 2015 - 14th International Semantic Web Conference, Bethlehem, PA, USA*, p. 495–512.

ZABLITH F., ANTONIOU G., D'AQUIN M., FLOURIS G., KONDYLAKIS H., MOTTA E., PLEXOUSAKIS D. & SABOU M. (2015). Ontology evolution : a process-centric survey. *Knowledge Eng. Review*, **30**(1), 45–75.