Enrichissement de Données RDF Integrées à la Volée

Benjamin Moreau, Emmanuel Desmontils, Patricia Serrano-Alvarado

Enrichissement de Données RDF Integrées à la Volée

Benjamin Moreau[*,**] Emmanuel Desmontils[*],
Patricia Serrano-Alvarado[*]

[*]GDD-LS2N – Nantes University, France
{Name.LastName@}univ-nantes.fr,
https://www.ls2n.fr
[**]OpenDataSoft
{Name.LastName}@opendatasoft.com
https://www.opendatasoft.com

Résumé. Les règles d'inférence sous-jacente à une ontologie sont des atouts majeurs du web des données. Cependant, mettre en place l'inférence est très couteux en temps d'exécution, stockage et maintenance. Certains producteurs de données décident de ne pas matérialiser leurs données en RDF ce qui rend compliqué l'enrichissement des données. Dans cette démonstration, nous présentons une approche pour bénéficier de l'enrichissement sémantique pendant l'exécution de requêtes SPARQL sur des données non-RDF accessibles à travers une API comme Twitter, Github et LinkedIn.

1 Introduction et Motivation

Les règles d'inférence sous-jacente à une ontologie sont des atouts majeurs du web des données. Concrètement, des triplets RDF implicites peuvent être déduits à partir des triplets explicites en exploitant les ontologies utilisées pour décrire les données. Cet enrichissement du jeu de données permet d'évaluer des nouvelles requêtes. Pour être exploitables, les triplets implicites peuvent être ajoutés a priori au graphe RDF ou être retournés pendant l'exécution de la requête.

Des travaux comme (Weaver et Hendler, 2009) et (Subercaze et al., 2016) proposent de matérialiser les triplets implicites. Cette approche augmente fortement le volume d'un jeux de données. D'autres travaux comme (Pérez-Urbina et al., 2009) et (Rosati et Almatelli, 2010) utilisent des techniques de réécriture de requêtes pour tenir compte des triplets implicites sans les matérialiser. Avec cette approche, le coût de stockage est limité, mais le temps d'exécution des requêtes est augmenté.

Pour diminuer les coûts de stockage et de maintenance, certains fournisseurs proposent un accès à leurs données au format RDF en utilisant des approches d'intégration virtuelle. Ces approches utilisent la réécriture de requêtes pour évaluer des requêtes SPARQL sur des données non-RDF. Un *mapping*, décrit par un langage de mapping RDF (Dimou et al., 2014; Lefrançois et al., 2017), est utilisé pour représenter les correspondances entre les données non-RDF et RDF. Dans ce contexte, (Michel et al., 2016) propose l'exécution de requêtes SPARQL sur des documents MongoDB. Dans notre travail, nous visons l'intégration au web des données

de n'importe quelle source de données non-RDF accessible à travers une API (Moreau et al., 2017). L'approche, nommée ODMTP (On-Demand Mapping using Triple Patterns), utilise TPF (Triple Pattern Fragments) (Verborgh et al., 2016) dont l'interface est simple. Seuls des triplets requêtes sont évaluées sur le serveur.

Supporter la déduction de triplets avec les approches d'intégration RDF à la volée peut s'avérer difficile, car les triplets implicites ne peuvent pas être déduits sans les triplets explicites. De plus, les techniques de réécriture de requêtes augmentent le temps d'exécution des requêtes, lequel est déjà détérioré par l'intégration des données RDF à la volée. Nous proposons donc une approche simple qui consiste à étendre le mapping RDF pour permettre l'exécution d'un domaine plus large de requêtes SPARQL sur des données non-RDF. Notre objectif est de limiter les surcoûts en terme de stockage et de temps d'exécution des requêtes SPARQL.

Considérons un exemple d'intégration au web des données des données de Twitter. L'API Twitter permet d'accéder au contenu d'un tweet, aux liens présents dans le tweet, aux hashtags et aux différentes méta-données. Le Listing 4 représente un extrait de l'ontologie qui peut être utilisée pour décrire ces données en RDF. Nous utilisons cette ontologie pour décrire, en RML, la transformation des données Twitter en RDF. Le Listing 1 représente un extrait de ce mapping.

```
...
<#Tweets>
  rr:subjectMap [ rr:template "https://twitter.com/statuses/{$.id_str}";
    rr:class schema:SocialMediaPosting; ];
  rr:predicateObjectMap [ rr:predicate it:includedHashtag;
    rr:objectMap [xrr:reference "$.entities.hashtags";]; ]
```

Listing 1 – *Extrait d'un mapping.*

Ce mapping permet à ODMTP d'interroger l'API Twitter pour répondre à des requêtes SPARQL comme celle du Listing 2 où on recherche les tweets (*schema:SocialMediaPosting*) qui correspondent à un Hashtag en particulier. Mais l'ontologie nous indique qu'un tweet est aussi un *schema:Article*. Afin d'élargir le domaine des requêtes possibles, nous utilisons le mapping et l'ontologie pour déduire les triplets implicites.

Le résultat de cette inférence nous donne le mapping étendu du Listing 5. Ce mapping permet d'exécuter des requêtes plus générales comme celle du Listing 3 ce que le mapping du Listing 1 ne peut pas faire. En effet, ce dernier ne décrit pas les triplets correspondants aux triplets de la requête.

```
SELECT ?tweet
WHERE { ?tweet a schema:SocialMediaPosting .
        ?tweet it:includedHashtag "SemanticWeb" }
```

Listing 2 – *Une requête SPARQL exécutable sur les triplets explicites.*

```
SELECT ?article
WHERE { ?article a schema:Article .
        ?article it:includedHashtag "SemanticWeb" }
```

Listing 3 – *Une requête SPARQL exécutable sur les triplets implicites.*

```
...
schema:SocialMediaPosting rdfs:subClassOf schema:Article;
                rdfs:subClassOf schema:CreativeWork;
                rdfs:subClassOf schema:Thing.
it:includedHashtag rdfs:domain schema:SocialMediaPosting;
                rdfs:range xsd:String.
```

Listing 4 – *Extrait de l'ontologie.*

```
...
<#Tweets>
  rr:subjectMap [ rr:template "https://twitter.com..."
    rr:class schema:SocialMediaPosting;
    rr:class schema:Article;
    rr:class schema:CreativeWork;
    rr:class schema:Thing; ];
  rr:predicateObjectMap [rr:predicate it:includedHashtag;
    rr:objectMap [xrr:reference "$.entities.hashtags";];]
```

Listing 5 – *Extrait d'un mapping étendu.*

Dans cette démonstration, nous montrons l'efficacité de notre approche en proposant une extension de ODMTP supportant l'inférence. Cette extension étend le mapping RDF pour permettre l'exécution de requêtes SPARQL sur les triplets explicites et implicites sans avoir besoin de les matérialiser.

2 Intégration à la demande avec mapping étendu

La Figure 1 illustre le mécanisme de raisonnement sur le mapping intégré à ODMTP. Au déploiement de ODMTP, le mapping est étendu en utilisant un raisonneur sémantique. Le raisonneur sémantique déduit les triplets implicites à partir du mapping, de l'ontologie utilisée pour décrire le mapping et des règles définies par RDFS ou OWL LD (Glimm et al., 2012).

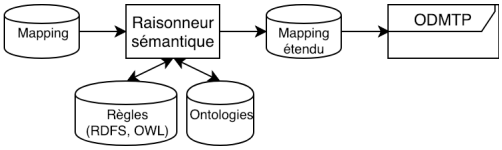

FIG. 1 – *ODMTP muni d'un raisonneur sémantique.*

Le mapping étant étendu au préalable, le temps d'exécution de la requête n'est pas détérioré par rapport à l'approche ODMTP classique. De plus, le temps de stockage est limité, car le nombre de triplets ajoutés dans le mapping est petit par rapport au nombre de triplets implicites d'un jeu de données RDF.

Cependant, toutes les règles de déduction ne sont pas applicables sur un mapping. L'extension d'un mapping RDF dépend de la catégorie de l'ontologie utilisée : RDFS, OWL Lite, OWL LD, etc. A chaque catégorie correspond des règles différentes. Par exemple, parmi les règles associées à RDFS [1], la règle *rdfs9* permet d'étendre un mapping. En effet, l'extension à partir des règles n'est possible que pour les règles s'appliquant aux concepts et aux propriétés manipulés dans le mapping. A l'inverse, les règles prenant en compte seulement les individus ne sont pas applicables car les individus ne sont pas matérialisés. Comme par exemple, dans la règle de transitivité sur les propriétés *prp-trp* de l'ontologie OWL LD. Une liste des règles implémentées est disponible sur le dépot Github [2].

3 Démonstration

Dans cette démonstration, nous utilisons une implémentation de ODMTP pour Twitter, Github et LinkedIn [3]. Les participants exécuteront des requêtes SPARQL sur les API de Twitter, Github et LinkedIn. Ils constateront que ODMTP supportant l'inférence peut exécuter plus de requêtes, que le temps d'exécution de la requête n'est pas détérioré et que le coût supplémentaire pour stocker le mapping étendu est minime.

1. https://www.w3.org/TR/rdf11-mt/#rdfs-entailment
2. https://github.com/benjimor/odmtp-tpf#supported-rules
3. https://github.com/benjimor/odmtp-tpf

La limite de cette approche est l'impossibilité d'appliquer certaines règles d'inférences se reportant aux individus. Il serait possible de les appliquer au moment de la matérialisation des données RDF. Cependant cela détériorerait le temps d'exécution des requêtes.

Références

Dimou, A., M. Vander Sande, P. Colpaert, R. Verborgh, E. Mannens, et R. Van de Walle (2014). RML: A Generic Language for Integrated RDF Mappings of Heterogeneous Data. In *LDOW in World Wide Web Conf (WWWC)*.

Glimm, B., A. Hogan, M. Krötzsch, et A. Polleres (2012). OWL: Yet to Arrive on the Web of Data ? In *LDOW in World Wide Web Conf (WWWC)*.

Lefrançois, M., A. Zimmermann, et N. Bakerally (2017). A SPARQL Extension for Generating RDF from Heterogeneous Formats. In *European Semantic Web Conf (ESWC)*.

Michel, F., C. Faron-Zucker, et J. Montagnat (2016). A Mapping-based Method to Query MongoDB Documents with SPARQL. In *Database and Expert Systems Applications (DEXA)*.

Moreau, B., P. Serrano-Alvarado, E. Desmontils, et D. Thoumas (2017). Querying non-RDF Datasets using Triple Patterns. *Demo in International Semantic Web Conf (ISWC)*.

Pérez-Urbina, H., I. Horrocks, et B. Motik (2009). Efficient Query Answering for OWL 2. In *International Semantic Web Conf (ISWC)*.

Rosati, R. et A. Almatelli (2010). Improving Query Answering over DL-Lite Ontologies. In *Principles of Knowledge Representation and Reasoning (KR)*.

Subercaze, J., C. Gravier, J. Chevalier, et F. Laforest (2016). Inferray: Fast In-memory RDF Inference. *VLDB Endowment*.

Verborgh, R., M. Vander Sande, O. Hartig, J. Van Herwegen, L. De Vocht, B. De Meester, G. Haesendonck, et P. Colpaert (2016). Triple Pattern Fragments: A Low-cost Knowledge Graph Interface for the Web. *Journal of Web Semantics. 37*.

Weaver, J. et J. A. Hendler (2009). Parallel Materialization of the Finite RDFS Closure for Hundreds of Millions of Triples. In *International Semantic Web Conf (ISWC)*.

Summary

Inference rules using ontologies is one of the major assets of the web of data. However, implementing inference is very expensive in terms of execution time, storage and maintenance. Some data producers decide not to materialize their data in RDF which makes it difficult to enrich the data. In this demonstration, we present an approach to benefit from the semantic enrichment during the execution of SPARQL queries on non-RDF data accessible through an API like Twitter, Github and LinkedIn.