SemLAV : Interroger le Web profond et le Web des données avec SPARQL

Pauline Folz, Gabriela Montoya, Hala Skaf-Molli, Pascal Molli, Maria-Esther Vidal

SemLAV : Interroger le Web profond et le Web des données avec SPARQL *

Pauline Folz[1,2], Gabriela Montoya[1,3], Hala Skaf-Molli[1], Pascal Molli[1] et Maria-Esther Vidal[4]

[1] LINA – Université de Nantes, France
{pauline.folz,gabriela.montoya,hala.skaf,pascal.molli}@univ-nantes.fr
[2] Nantes Métropole – Direction Recherche, Innovation et Enseignement Supérieur, France
[3] Centre National de la Recherche Scientifique (CNRS) : UMR6241, France
[4] Universidad Simón Bolívar, Venezuela
mvidal@ldc.usb.ve

Résumé SemLAV permet d'exécuter des requêtes SPARQL à travers des sources provenant du Web profond et du Web des données. SemLAV implémente l'architecture *médiateur* basée sur la définition des vues par rapport aux sources de données distantes. Les requêtes SPARQL sont exprimées en utilisant le vocabulaire des médiateurs et SemLAV sélectionne les sources pertinentes et les classe. La stratégie de classement est conçue pour délivrer des résultats rapidement en se basant seulement sur la définitions des vues, c-à-d, ni statistiques, ni sondage sur les sources ne sont nécessaires. Dans cette démonstration, nous montrons l'efficacité de SemLAV avec de vraies données issues de réseaux sociaux et du Web des données. En effet, la matérialisation d'un sous ensemble des vues préalablement sélectionnées et classées est suffisant pour produire une partie significative des résultats attendus.

1 Introduction

Le Web profond est constitué de données non indexées par les moteurs de recherches traditionnels sur lesquelles il est difficile de disposer de statistiques. Le Web profond représente environ 500 fois la taille du Web indexable [2]. L'éxecution de requêtes SPARQL sans prendre en considération le Web profond peut entraîner une pauvreté des résultats. Par exemple, l'exécution de la requête SPARQL : « Quels membres de la communauté du Web sémantique sont intéressés par le *Dalai Lama*, *Barck Obama* ou *Rihanna* ? » (cf. figure 1) sans prendre en compte le Web profond, ne produit pas de réponse. Quelques outils tels que Virtuoso avec SPONGER [1] répondent à cette problématique en déclarant des *wrappers* capables d'interroger des données non-sémantifiées (ex. : fichiers CSV, appels aux web services, ...). Ce type d'approche est pertinent si le nombre de sources pour répondre à la requête est faible. De plus, le temps pour produire la première réponse peut être très élevé, car le moteur de requête doit d'abord contacter tous les *wrappers* déclarés pour cette requête.

∗. Ce travail a déjà été accepté pour publication comme démonstration à la conférence ESWC 2014.

```
SELECT DISTINCT *
WHERE {
  ?P foaf:member ?C .
  ?C rdfs:label "Semantic Web" .
  ?P foaf:knows ?WKP .
  ?WKP foaf:name ?N .
   FILTER (?N="Dalai Lama" || ?N="Barack Obama" || ?N="Rihanna")
}
```

FIGURE 1: Quels membres de la communauté du Web sémantique sont intéressés par le *Dalai Lama*, *Barck Obama* ou *Rihanna* ?

À contrario, SemLAV [3] est capable de produire des réponses rapidement pour cette même requête. Il suit l'approche médiateur où les données du Web profond peuvent être récupérées grâce à la définition des vues et des *wrappers*. Une vue représente une source de données pour le médiateur. Pour une requête SPARQL donnée, SemLAV sélectionne les vues pertinentes et les classe. Le classement est estimé selon le degré de couverture des vues sur les réécritures de la requête originale. SemLAV utilise les *wrappers* pour sémantifier à la demande les données des sources sélectionnées. Il récupère les données des sources qui sont classées dans un ordre spécifique, ce qui permet d'avoir une forte probabilité de produire des résultats, même en présence d'un nombre important de vues pertinentes. Ces réponses sont produites en un temps raisonnable. Dans ce papier, nous démontrons comment SemLAV est capable de produire rapidement des résultats pour des requêtes SPARQL mixant des données du Web profond et du Web des données, en utilisant environ 250 vues. Une vidéo de la démo est disponible à : `https://www.youtube.com/channel/UCMQO5QVq5UcztE8kkkRRXKQ/videos`.

2 Architecture de SemLAV

Pour une requête donnée et un ensemble de vues, SemLAV calcule un ensemble de vues pertinentes et les classe pour répondre à la requête. SemLAV estime le nombre de réécritures équivalentes, c-à-d, combien de réécritures un moteur *Local-As-View* doit executer pour produire les mêmes résultats que l'exécution de la requête originale sur la matérialisation des vues selectionnées [3]. Les vues sont matérialisées en appelant les *wrappers* comme ceux définis dans SPONGER [1], en séquence ou en parallèle. Chaque fois qu'une nouvelle vue est complétement matérialisée, la requête d'origine est éxecutée pour produire des résultats aussi rapidement que possible. Les vues utilisées dans SemLAV peuvent égalament être générées par des outils comme Karma [4]. Pour illustrer les avantages de SemLAV, considérer la requête définie dans la figure 1 et les cinq vues suivantes :

```
v1(P,A,I,C,L):-made(P,A),affiliation(P,I),member(P,C),label(C,L)
v2(A,T,P,N,C):-title(A,T),made(P,A),name(P,N),member(P,C)
v3(P,N,R,M):-name(P,N),name(R,M),knows(P,R)
v4(P,N,G,R,C):-name(P,N),gender(P,G),knows(P,R),member(P,C)
v5(P,N,R,C,L):-name(P,N),knows(P,R),member(P,C),label(C,L)
```

SemLAV calcule les ensembles triés suivant pour chaque sous-objectif de la requête :

member(P, C)	label(C, L)	knows(P, WKP)	name(WKP, N)
v5(P,N,R,C,L)	v5(P,N,R,C,L)	v5(P,N,R,C,L)	v5(P,N,R,C,L)
v4(P,N,G,R,C)	v1(P,A,I,C,L)	v4(P,N,G,R,C)	v4(P,N,G,R,C)
v1(P,A,I,C,L)		v3(P,N,R,M)	v2(A,T,P,N,C)
v2(A,T,P,N,C)			v3(P,N,R,M)

L'éxecution de toutes les combinaisons possibles produit la réponse complète de la requête. Pour produire des réponses rapidement, SemLAV classe les vues pertinentes selon leur contribution à couvrir les sous-objectifs de la requête. En d'autres termes, les vues les mieux classées sont celles qui couvrent le plus de sous-objectifs. Par conséquent, le nombre de combinaisons couvertes augmente aussi vite que possible.

# Included	SemLAV ranking		Random order	
views (k)	Included views (V_k)	# Covered rewritings	Included views (V_k)	# Covered rewritings
1	v5	$1 \times 1 \times 1 \times 1 = 1$	v1	$1 \times 1 \times 0 \times 0 = 0$
2	v5, v4	$2 \times 1 \times 2 \times 2 = 8$	v1, v2	$2 \times 1 \times 0 \times 1 = 0$
3	v5, v4, v1	$3 \times 2 \times 2 \times 2 = 24$	v1, v2, v3	$2 \times 1 \times 1 \times 2 = 4$
4	v5, v4, v1, v3	$3 \times 2 \times 3 \times 3 = 54$	v1, v2, v3, v4	$3 \times 1 \times 2 \times 3 = 18$
5	v5, v4, v1, v3, v2	$4 \times 2 \times 3 \times 4 = 96$	v1, v2, v3, v4, v5	$4 \times 2 \times 3 \times 4 = 96$

3 Scénario

Dans cette démonstration, nous utilisons des sources populaires du Web profond comme les résaux sociaux Twitter et Facebook, ainsi que des sources du Web des données comme DBLP, Semantic Web Dog Food et DBpedia. Nous avons défini 253 vues sur l'ensemble des sources citées ci-dessus. Nous utilisons plusieurs vocabulaires RDF pour décrire les membres d'une communauté, pour les lier entre eux et avec le *cloud* du Web des données. Nous supposons : *i*) qu'une personne est membre d'une communauté s'il y a un lien entre cette personne et cette communauté. Ce lien est représenté avec le prédicat `foaf:member`. Il peut se traduire, par exemple, quand quelqu'un suit un compte Twitter d'une conférence de la communauté, quand quelqu'un est membre d'un groupe Facebook de la communauté, ou quand quelqu'un a publié un papier dans une conférence appartenant à la communauté ; *ii*) qu'une personne connaît une autre personne s'il y a un lien entre elles. Ce lien est représenté par le prédicat `foaf:knows` et se traduit par un utilisateur qui suit quelqu'un sur Twitter ou quand deux personnes sont co-auteurs d'un papier, par exemple.

```
SELECT DISTINCT *
WHERE {
  ?follower <http://xmlns.com/foaf/0.1/name> ?name .
  ?follower <http://xmlns.com/foaf/0.1/knows> ?followed .
  ?follower <http://xmlns.com/foaf/0.1/member> ?community .
  ?community <http://www.w3.org/2000/01/rdf-schema#label> "Semantic Web"
}
```

FIGURE 2: Description des *followers* du compte Twitter de la conférence ESWC.

Les sources sont décrites par des requêtes SPARQL (cf. figure 2).

3.1 Requêtes

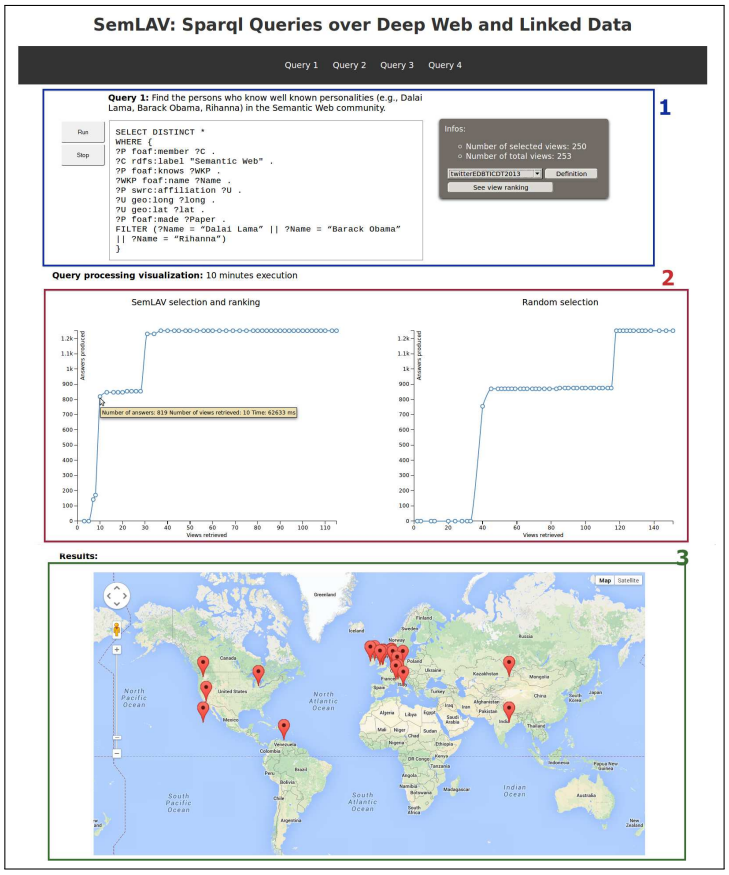

FIGURE 3: Capture d'écran pour l'excécution de la requête 1. Pendant la démonstration tous les résultats seront calculés sur demande.

Nous montrons le comportement de SemLAV à travers 4 requêtes : *requête 1*) les membres de la communauté du Web sémantique qui connaissent des personnalités (comme le Dalai Lama, Barack Obama ou Rihanna) ; pour ces personnes nous montrons leur affiliation, leur localisation et le nombre de contributions qu'ils ont fait dans la communauté ; *requête 2*) les membres des différentes communautés scientifiques qui connaissent Tim Berners-Lee ; *requête 3*) les membres de la communauté du Web sémantique

qui ont été les plus actifs sur Twitter en postant des tweets avec le *hashtag* ESWC2014 ; *requête 4*) les membres de la communauté de base de données qui sont connus par les membres du Web sémantique. Pour ces personnes, nous montrons leur affiliation et leur localisation.

La figure 3 représente l'interface de la démonstration. Les résultats sont calculés et affichés dynamiquement. Les requêtes peuvent être sélectionnées avec le menu en haut. Le rectangle bleu (numéro 1) permet de lancer et d'interrompre une requête, via les boutons `Run` et `Stop`. Ainsi, que de connaître le nombre de vues pertinentes et total pour cette même requête. Pour l'ensemble des vues il est possible de voir leur définition (`Definition`) et pour les vues sélectionnées, leur classement.

Le rectangle rouge (numéro 2) montre l'état de l'exécution de la requête 1 après 10 minutes d'éxecution que ce soit en utilisant SemLAV ou une approche aléatoire. Les graphiques en ligne illustrent la relation entre le nombre de réponses produites et le nombre de vues qui ont été matérialisées. Le nombre de réponses produites augmente de façon rapide après avoir matérialisé 6% des vues (15 des 253 vues), soit plus de 50% des réponses attendues. De plus, pour le même nombre de vues matérialisées, par exemple pour 10 vues, SemLAV produit 819 réponses alors que l'approche aléatoire ne produit pas de réponse. Dans le rectangle vert (numéro 3), le résultat des requêtes est affiché sur une carte selon la localisation des réponses.

4 Conclusion

Dans cette démonstration, nous montrons comment SemLAV exécute des requêtes SPARQL sur des sources de données du Web profond et du Web des données. Dans ces différents cas d'utilisation, les sources sont sélectionnées et classées par SemLAV de façon à produire des réponses de manière incrémentale et rapide, même avec un petit nombre de vues matérialisées, SemLAV est capable de produire des réponses.

Références

1. Virtuoso sponger. White paper, OpenLink Software.
2. B. He, M. Patel, Z. Zhang, and K. C.-C. Chang. Accessing the Deep Web. *Commun. ACM*, 50(5) :94–101, 2007.
3. G. Montoya, L. D. Ibáñez, H. Skaf-Molli, P. Molli, and M.-E. Vidal. SemLAV : Local-As-View Mediation for SPARQL. *Transactions on Large-Scale Data- and Knowledge-Centered Systems XIII, Lecture Notes in Computer Science, Vol. 8420*, pages 33–58, 2014.
4. M. Taheriyan, C. A. Knoblock, P. A. Szekely, and J. L. Ambite. Rapidly Integrating Services into the Linked Data Cloud. In *International Semantic Web Conference (1)*, pages 559–574, 2012.