Cohérence des données de bases RDF en évolution constante

Pierre Maillot[*,**] Thomas Raimbault[*] David Genest[**]

[*]De Vinci Technology Lab, ESILV, 92 916 Paris La Défense Cedex
{pierre.maillot, thomas.raimbault}@devinci.fr
[**]LERIA, Université d'Angers, 2 Boulevard Lavoisier, 49 045 Angers
genest@info.univ-angers.fr

Résumé. Le maintien de la qualité et de la fiabilité de bases de connaissances RDF du Web Sémantique est un problème courant. De nombreuses propositions pour l'intégration de « bonnes » données ont été faites, se basant soit sur les ontologies de ces bases, soit sur des méta-données additionnelles. Dans cet article, nous proposons une approche originale, basée exclusivement sur l'étude des données de la base. Le principe est de déterminer si les modifications apportées par la mise à jour candidate rendent la partie ciblée de la base plus similaire – selon certains critères – à d'autres parties existantes dans la base. La mise à jour est considérée cohérente avec cette base et peut être appliquée.

1 Introduction

Le *Web Sémantique* a été lancé en 2001 par le W3C [1] pour promouvoir le partage et la création de données structurées sur le Web en proposant des recommandations pour la description de données (RDF), d'ontologies (RDFS, OWL), et des méthodes et outils associés (SPARQL, ...) pour gérer les connaissances. Actuellement le Web Sémantique correspond à des centaines de bases RDF communautaires (*e.g.* DBPedia [2] ou Yago [3]). Ces bases contiennent des données de sources variées, résultant soit de contributions d'experts dans des domaines spécifiques, soit du moissonnage de pages Web ou de fichiers, soit enfin et surtout de contributions issues de la production participative. Les méthodes de moissonnage et de production participative sont hélas sources d'incohérences résultantes d'erreurs humaines directes ou indirectes.

Pour contrer l'apparition de ces incohérences, de nombreuses méthodes ont été proposées pour maintenir la qualité et la fiabilité des données au sein des bases RDF. Par exemple, Mendes et al. (2012) utilise des méta-données telles que la provenance ou l'historique des éditions précédentes, tandis que Jacobi et al. (2011) utilise les ontologies pour évaluer une valeur de confiance. Bonatti et al. (2011) combine les deux approches. Enfin, un état de l'art de l'évaluation de la qualité de des données RDF est fait dans Zaveri et al. (2013). On constatera néanmoins que les ontologies souffrent de leurs difficultés à s'adapter aux évolutions dans la description des données et que les approches d'intégration basées sur des méta-données requièrent l'utilisation de méthodes ad-hoc pour ces méta-données additionnelles.

1. http://www.w3.org/standards/semanticweb/
2. http://dbpedia.org
3. http://www.mpi-inf.mpg.de/yago-naga/yago/

Notre contribution est de fournir une méthode originale d'évaluation de mises à jour, inspirée du raisonnement par cas, utilisant exclusivement les données de la base RDF mise à jour (*i.e.* ne nécessitant pas l'utilisation d'une ontologie ou de méta-données). Par cette méthode, une mise à jour *candidate* est évaluée positivement si ses modifications dans la base RDF rendent – selon certains critères – la partie cible mise à jour dans la base plus structurellement similaire à d'autres parties de la base. Notre méthode d'évaluation de la cohérence peut être décomposée en 3 étapes : (i) extraction des contextes de la mise à jour depuis la base, (ii) récupération des parties de la base similaires à la mise à jour et à ses contextes et (iii) évaluation par similarité de la cohérence des données de la mise à jour par rapport à la base.

En Section 2 nous définissons une mise à jour RDF et ses contextes (première étape de notre approche). En Section 3 nous détaillons notre méthode d'évaluation de mise à jour en définissant quelles sous-parties de la base sont prises en compte lors de l'évaluation d'une mise à jour (Section 3.1), et enfin comment évaluer la cohérence d'une mise à jour RDF par rapport à une base (Section 3.2).

2 Mise à jour RDF

Nous introduisons ici quelques définitions pour formaliser les notions de *mises à jour* RDF et de *contextes* associés dans une base RDF.

Nous rappelons quelques vocabulaires du Web Sémantique et introduisons quelques termes utilisés dans la suite de l'article. Ainsi, nous considérerons comme synonymes les termes *document RDF*, *base RDF* et *ensemble de triplets RDF* ; pour un document RDF D nous noterons $\mathcal{R}_D$ l'ensemble des ressources – sujet, prédicat ou objet – des triplets de D ; nous appellerons une *ressource nœud* une ressource étant soit sujet, soit objet d'un triplet ; pour un document RDF D, nous noterons $\mathcal{N}_D$ l'ensemble des ressources nœud de D ; nous appellerons *document RDF connexe* un document RDF dans lequel il y a un chemin connectant chaque ressource nœud du document à une autre, en d'autres termes si le graphe RDF représentant le document est un graphe connexe ; nous appellerons *degré* d'une ressource nœud le nombre de triplets la contenant dans une base RDF, en d'autres termes son degré dans le graphe RDF.

Notons aussi que implicitement nous désignons toujours les données d'une base RDF sans (avant) que les modifications d'une mise à jour ne lui soient appliquées. Enfin, toute mise à jour d'une base RDF peut être vue en tant que combinaison de deux sections : une *section d'ajout* qui contient ce que la mise à jour ajoute à la base et une *section de suppression* qui contient ce que la mise à jour supprime dans la base.

Définition 1 (Mise à jour RDF)**.** *Une mise à jour RDF u d'une base RDF B est un couple d'ensembles de triplets RDF (A, R) tels que :*

- *A est appelé* section d'ajout*, avec $A \nsubseteq B$ et $A \cap B \neq \emptyset$;*
- *$\mathcal{N}_B \cap \mathcal{N}_A \neq \emptyset$;*
- *R est appelé* section de suppression*, avec $R \subseteq B$;*
- *$A \cap R = \emptyset$ et $A \neq \emptyset$;*
- *$A \cup R$ est un document RDF connexe.*

Une mise à jour RDF qui ajoute des informations à une base doit apporter de nouveaux éléments liés à des données déjà existantes. Une mise à jour qui supprime des informations peut uniquement supprimer des données déjà existantes dans la base. Les sections d'ajout et de

suppression ne contiennent pas de triplets en commun (l'ordre d'application de la suppression ou de l'ajout dans la base n'a pas d'importance). Une mise à jour contient nécessairement une section d'ajout et forme un document connexe (autrement il s'agit de 2 mises à jour distinctes).

De la Définition 1 nous pouvons classer les mises à jour RDF en deux catégories : les *mises à jour d'ajout*, définies par $A \neq \emptyset$ et $R = \emptyset$, et les *mises à jour de modification*, définies par $A \neq \emptyset$ et $R \neq \emptyset$. Le cas des suppressions « pure » est discuté en conclusion.

Pour comparer les données d'une mise à jour à une base RDF selon notre évaluation, nous utilisons le voisinage dans la base de toutes les ressources de la mise à jour. Ainsi les contextes d'une mise à jour sont obtenus grâce aux voisinages des sections d'ajout et de suppression.

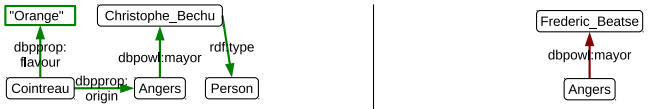
Définition 2 (Contextes de mise à jour RDF)**.** *Soient un ensemble de triplets RDF B, une mise à jour RDF $u = (A, R)$ candidate à B et $n \in \mathbb{N}$ un rang de voisinage. Soit la fonction voisinage$^n_B(r) : \mathcal{N}_B \rightarrow B$ retournant tous les triplets de B connectés à r par un chemin de longueur égale ou inférieure à n, appelée* fonction de voisinage *de r.*

Les contextes d'une mise à jour *u candidate à B sont les deux ensembles de triplets RDF I_u et F_u définis par :*

- *I_u appelé le* contexte initial *de u dans B tel que $I_u = \{t | t \in$ voisinage$^n_B(r)$, $r \in \mathcal{N}_{A \cup R}\}$ et $I_u \subseteq B$;*
- *F_u appelé le* contexte final *de u dans B tel que $F_u = I_u \cup A \setminus R$.*

Le contexte initial représente l'état initial de la partie de la base autour des ressources de la mise à jour candidate, le contexte final représente l'état théorique de la base si la mise à jour était appliquée.

Exemple 1 (Mise à jour u_1)**.** Considérons la mise à jour fictive u_1 pour la base DBPedia ajoutant des informations à propos du goût et de l'origine de la liqueur « Cointreau » et modifiant les informations à propos du nouveau maire de la ville d'Angers, d'où vient la liqueur. Fig. 1 est une représentation graphique de la section d'ajout et de la section de suppression de u_1.



A_1 *section d'ajout de* u_1. | R_1 *section de suppression de* u_1.

FIG. 1 – *Représentation graphique de $u_1 = \{A_1, R_1\}$ (éléments ajoutés en trait épais vert, éléments supprimés en trait épais rouge).*

Exemple 2 (Contextes de u_1)**.** Le contexte initial I_{u_1} et le contexte final F_{u_1} de u_1, sont graphiquement représentés en Fig. 2 avec un rang de voisinage de 1 dans la base DBPedia.

3 Évaluer la cohérence par mesure de la similarité

Nous considérons qu'une mise à jour est cohérente avec une base si on peut trouver suffisamment de sous-parties de la base suffisamment similaires avec les contextes de la mise à jour. Nous procédons en 3 étapes : (i) recherche dans la base de sous-parties structurellement comparables aux contextes de la mise à jour, (ii) quantification de la similarité entre chaque sous-partie et les contextes de la base, (iii) conclusion sur la cohérence de la mise à jour.

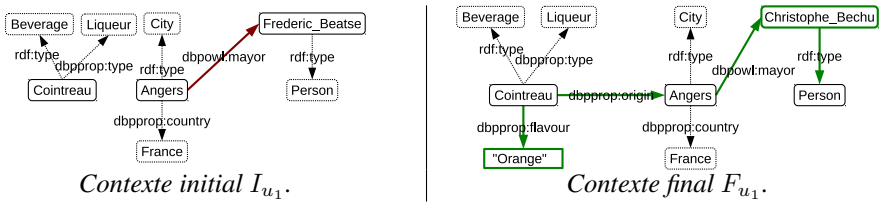


FIG. 2 – *Contextes de u_1 (En pointillés : voisinage dans la base des ressources de u_1)*

3.1 Trouver des références dans la base

Deux ensembles de triplets RDF peuvent être structurellement comparés si leurs ressources nœuds sont liées de façon similaire, incluant (au moins) une ressource commune.

Définition 3 (Ensembles de triplets RDF comparables)**.** *Soit deux ensembles de triplets RDF connexes G et H.*

G est comparable à H s'il existe une fonction de transformation $f : \mathcal{R}_G \to \mathcal{R}_H$ telle que :

— $\mathcal{R}_G \cap \mathcal{R}_H \neq \emptyset$.

— $\exists (s, p, o) \in G$ *tel que* $(f(s), f(p), f(o)) \in H$ *ou* $(f(o), f(p), f(s)) \in H$ *;*

Deux ensembles comparables contiennent au moins une ressource commune. De plus, en théorie des graphes, on dira qu'un ensemble de triplets RDF est comparable à un autre si il est homomorphique à une partie d'un autre, sans considérer l'orientation des arcs.

Définition 4 (Référence d'une mise à jour)**.** *Soit une mise à jour RDF $u = (A, R)$ candidate à une base RDF B.*

Une référence *D de u dans B est telle que $D \subseteq B$ et D est comparable à $A \cup R$.*

Comparer une mise à jour à une base entière signifie comparer structurellement les contextes initial et final de la mise à jour à chaque sous-partie de la base comparable à la mise à jour. Nous appelons références ces sous-parties de la base dépendantes de la mise à jour.

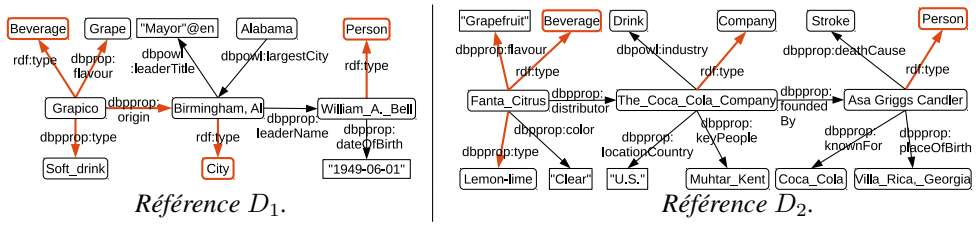


FIG. 3 – *Sous-parties D_1 et D_2 de DBPedia, en tant que* références *pour u_1 en Fig. 1 (Éléments en commun avec u_1 en traits épais orange).*

De la base DBPedia, deux références pour u_1 peuvent être extraites, notées D_1 et D_2 et représentées en Fig. 3. D_1 et D_2 contiennent plusieurs ressources en commun avec u_1. D_1 suit le même modèle que u_1 avec une boisson liée à une ville liée à une personne, alors que D_2 concerne une boisson liée à une entreprise liée à une personne.

3.2 Évaluer la cohérence d'une mise à jour

Dans notre approche, si les modifications d'un mise à jour rendent la partie ciblée de la base plus similaire à d'autres parties (existantes) de la base alors nous évaluons positivement cette mise à jour.

Nous proposons d'évaluer la similarité de la mise à jour par rapport à chacune de ses références à l'aide d'une mesure de la similarité structurelle entre deux graphes. Dans l'évaluation en Définition 5, nous supposons l'usage d'une mesure donnant un score de similarité dans $\mathbb{R}^+$ tel que plus le score est élevé, plus la similarité est grande (un score de 0 signifie aucune similarité). Plusieurs mesures sont utilisables telles que la distance d'édition entre deux graphes, le coefficient de Jaccard, *etc.*

Définition 5 (Évaluation de la cohérence d'une mise à jour par similarité)**.** *Soit une mise à jour u candidate à une base RDF B et ses contextes I_u et F_u, l'ensemble $\mathcal{D}_u$ des références associées à u dans B, $\mathcal{D}_u = \{D_1, ..., D_n\}$, un nombre minimum de références $d \in \mathbb{N}^*$ et un nombre minimum d'évaluations positives $m \in \mathbb{N}^*$. On note similarity la mesure de similarité entre deux graphes RDF avec similarity$(u, D_i) \in \mathbb{R}^+$.*

La fonction eval(u, B, d, m) telle que :

Si $|\mathcal{D}_u| \geq d$ *et* $|\{D_i|$ *similarity*$(I_u, D_i) -$ *similarity*$(F_u, D_i) > 0\}| \geq m$
alors eval$(u, B, d, m) =$ `true` *sinon eval*$(u, B, d, m) =$ `false`

est la fonction d'évaluation de la cohérence *de u dans B pour laquelle si eval(u, B, d, m) retourne* `true` *alors u est cohérente avec B.*

L'évaluation de la similarité dépend d'un ensemble de références associées à la mise à jour et d'un nombre minimum d'évaluations positives. Une mise à jour est cohérente avec une base si on peut lui trouver suffisamment de références ($\geq d$) et si ses modifications rendent la sous-partie cible de la base plus similaire a un nombre suffisant ($\geq m$) de références.

La valeur de similarité seule n'importe pas dans notre évaluation, seul le signe de la différence entre l'état final et initial indique si la mise à jour apporte des informations similaires à ce qui est déjà connu.

Exemple 4. Dans cet exemple, nous choisissons d'utiliser une mesure de similarité en considérant dans chaque ensemble de triplets l'ensemble des ressources et l'ensemble des couples de ressources (sujet, relation) et (relation, objet) où le score est calculé simplement – pédagogiquement – avec similarity $= +1$ pour chaque ressource commune et $+2$ pour chaque couple de ressources communes. La différence de similarité entre la référence D_1 et les contextes de u_1 est positive (similarity(F_{u_1}, D_1) − similarity(I_{u_1}, D_1) = $12 - 11$) et celle entre D_2 et les contextes de u_1 est nulle (similarity(F_{u_1}, D_2) − similarity(I_{u_1}, D_2) = $9 - 9$), ainsi, avec un nombre minimum de références de 1, on a $eval(u_1, B, 2, 1) =$ `true`.

La mise à jour u_1 est donc cohérente avec la base DBPedia : les modifications de la mise à jour créent des données structurellement similaires à des parties de la base. Cette mise à jour peut être appliquée à la base.

4 Conclusion

Dans cet article nous proposons une approche d'intégration, ou de mise à jour, de données dans des bases RDF par une évaluation de la cohérence des mises à jour en fonction de leur

similarité avec le contenu de la base, inspirée du raisonnement par cas. Dans un Web Sémantique où les ontologies évoluent beaucoup plus lentement que les données et leurs utilisations, notre approche est une méthode originale et adaptée pour l'évaluation de mises à jour candidates dans un milieu communautaire. Notre méthode a donné des résultats encourageants lors de tests sur DBPedia, a partir des informations de mise à jour disponibles via DBPedia Live [4]. Actuellement nous nous penchons sur le cas particulier des mises à jour ne contenant que des suppressions (section d'ajout vide). Notre méthode peut bien sûr être utilisée conjointement avec d'autres méthodes, particulièrement celles basées sur l'ontologie.

Notre approche peut à terme permettre d'identifier dans les mises à jour refusées les triplets qui sont à l'origine des rejets pour proposer des avertissements ou des corrections lors de l'édition d'une mise à jour. D'autres utilisations de notre approche sont possibles, par exemple pour étudier les parties de l'ontologie qui sont rarement respectées mais acceptées par similarité, révélant ainsi des parties des ontologies rendues obsolètes par les changements d'habitudes de description des données par la communauté et, pourquoi pas, proposer des évolutions de l'ontologie pour intégrer ces nouvelles habitudes.

Références

Bonatti, P. A., A. Hogan, A. Polleres, et L. Sauro (2011). Robust and scalable linked data reasoning incorporating provenance and trust annotations. *Web Semantics : Science, Services and Agents on the World Wide Web 9*(2), 165–201.

Jacobi, I., L. Kagal, et A. Khandelwal (2011). Rule-based trust assessment on the semantic web. In *Rule-Based Reasoning, Programming, and Applications*, pp. 227–241. Springer.

Mendes, P. N., H. Mühleisen, et C. Bizer (2012). Sieve : linked data quality assessment and fusion. In *Proceedings of the 2012 Joint EDBT/ICDT Workshops*, pp. 116–123. ACM.

Zaveri, A., A. Rula, A. Maurino, R. Pietrobon, J. Lehmann, S. Auer, et P. Hitzler (2013). Quality assessment methodologies for linked open data. *Submitted to Semantic Web Journal*.

Summary

Trust and quality maintenance have always been problematic in the Web Sémantique RDF bases. Numerous propositions to address these problems of data integration have been made, either based on ontologies or on additional metadata. In this article we propose an original approach, based exclusively on data from the base, to evaluate the consistency of a candidate update to a RDF base, and finally to know if this update is relevant to the base. If the modifications of a candidate update make the target part of the base more similar to other part(s) of the base, then this candidate update is considered consistent with the base and can be applied.

4. `http://wiki.dbpedia.org/DBpediaLive`