Linkky: Extraction de clés de liage par une adaptation de l'analyse relationnelle de concepts

Jérôme David, Jérôme Euzenat, Jérémy Vizzini

Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP, LIG, F-38000 Grenoble France
Jerome.David@inria.fr, Jerome.Euzenat@inria.fr, jeremyvizzini@icloud.com

Résumé : Les clés de liage permettent de spécifier la manière d'engendrer des liens entre deux sources de données RDF. L'objet de cette démonstration est de présenter Linkky, un prototype implémentant l'extraction de familles de clés de liage dépendantes à l'aide de techniques d'analyse formelle de concept.

Mots-clés : Liage de données, clés de liage, RDF, analyse formelle de concepts, analyse relationnelle de concepts

Le besoin d'accès aux données par la société a conduit à la publication, par différents acteurs (gouvernement, universités, acteurs culturels), de vastes corpus de données exprimées dans les formalismes du web sémantique (principalement RDF).

Une part importante de la valeur ajoutée des données liées réside dans les liens identifiant la même entité dans différents jeux de données. Par exemple, cela permet d'identifier les mêmes ouvrages dans différentes sources de données bibliographiques. Les liens permettent d'exploiter conjointement les données de ces sources.

Par conséquent, la génération de tels liens est une tâche importante pour le web des données. Elle est en général pilotée par une spécification de liage. Différents types de spécifications sont disponibles. La plus répandue consiste à calculer une distance entre les identifiants de ressources et à estimer que plus ils sont proches, plus ils ont de chance d'identifier la même ressource.

Un autre type de spécification est ce que nous appelons *clé de liage*. Les clés de liage généralisent les clés de bases de données dans deux directions indépendances : elles fonctionnent avec des données représentés en RDF, et elles s'appliquent à deux jeux de données indépendants. Un exemple de clé de liage est :

$$\{\langle \mathsf{auteur}, \mathsf{creator}\rangle\}\ \ \{\langle \mathsf{titre}, \mathsf{title}\rangle\}\ linkkey\ \langle \mathsf{Livre}, \mathsf{Book}\rangle \tag{1}$$

qui signifie que si deux instances des classes Livre et Book respectivement, ont les mêmes valeurs pour les propriétés auteur et creator et au moins une valeur commune pour les propriétés titre et title, alors elles dénotent la même ressource.

Une première méthode a été conçue pour extraire les clés de liage entre deux classes (Atencia *et al.*, 2014). Elle commence par extraire des clés candidates puis les évalue à l'aide de mesures adaptées.

L'analyse formelle de concepts (FCA ou AFC) est une technique pour extraire des concepts entre deux ensembles ordonnés interdépendants (Ganter & Wille, 1999). L'analyse relationnelle de concepts (RCA) en est une extension permettant d'extraire des descriptions interdépendantes entre différents concepts (Rouane-Hacene *et al.*, 2013). L'AFC a déjà été utilisée pour extraire les clés dans le modèle relationnel.

L'étape d'extraction de clés candidates a été reformulée en un problème d'analyse de concepts formels pour le cas simple des bases de données (Atencia *et al.*, 2014). Nous avons étendu ce travail pour prendre en compte les attributs non fonctionnels et les dépendances entre clés de liage (lorsque les conditions d'une clé nécessitent de déterminer l'égalité de deux instances d'autres classes, ce qui utilise une autre clé, voir Table 1). Pour prendre en compte les références circulaires, il est devenu nécessaire d'adapter les techniques de RCA au cas des clés de liage.

$K_{Person,Inhabitant}$	$\forall\langle lastname, given\rangle$	$\forall\langle lastname, name\rangle$	$\exists\langle lastname, given\rangle$	$\exists\langle lastname, name\rangle$	$\forall\langle home, address\rangle_{C4}$	$\exists\langle home, address\rangle_{C4}$	$\forall\langle home, address\rangle_{C5}$	$\exists\langle home, address\rangle_{C5}$	$\forall\langle home, address\rangle_{C8}$	$\exists\langle home, address\rangle_{C8}$	$\forall\langle home, address\rangle_{C1}$	$\exists\langle home, address\rangle_{C1}$	$\forall\langle home, address\rangle_{C0}$	$\exists\langle home, address\rangle_{C0}$
$\langle z_3, i_3\rangle$		×		×	×	×	×	×	×	×	×	×	×	×
$\langle z_3, i_2\rangle$		×		×							×	×	×	×
$\langle z_3, i_1\rangle$									×	×	×	×		
$\langle z_1, i_3\rangle$									×	×	×	×		
$\langle z_1, i_2\rangle$							×	×			×	×		
$\langle z_1, i_1\rangle$		×		×	×	×	×	×	×	×	×	×	×	×
$\langle z_2, i_3\rangle$		×		×							×	×	×	×
$\langle z_2, i_2\rangle$		×		×	×	×	×	×	×	×	×	×	×	×
$\langle z_2, i_1\rangle$							×	×			×	×		

$K_{House,Place}$	$\forall\langle city, city\rangle$	$\exists\langle city, city\rangle$	$\forall\langle owner, owned By\rangle_{C6}$	$\exists\langle owner, owned By\rangle_{C6}$	$\forall\langle owner, owned By\rangle_{C2}$	$\exists\langle owner, owned By\rangle_{C2}$	$\forall\langle owner, owned By\rangle_{C9}$	$\exists\langle owner, owned By\rangle_{C9}$	$\forall\langle owner, owned By\rangle_{C3}$	$\exists\langle owner, owned By\rangle_{C3}$	$\forall\langle owner, owned By\rangle_{C7}$	$\exists\langle owner, owned By\rangle_{C7}$
$\langle h_1, a_2\rangle$					×	×			×	×		
$\langle h_1, a_1\rangle$	×	×	×	×	×	×	×	×	×	×	×	×
$\langle h_1, a_3\rangle$									×	×	×	×
$\langle h_3, a_2\rangle$	×	×					×	×	×	×		
$\langle h_3, a_1\rangle$									×	×	×	×
$\langle h_3, a_3\rangle$	×	×	×	×	×	×	×	×	×	×	×	×
$\langle h_2, a_2\rangle$	×	×	×	×	×	×	×	×	×	×	×	×
$\langle h_2, a_1\rangle$					×	×			×	×		
$\langle h_2, a_3\rangle$	×	×					×	×	×	×		

TABLE *1 – Contexte formel étendu après six itérations d'analyse relationelle de concepts conduisant au treillis de la Figure 1.*

Linkky [1] est un démonstrateur de ces techniques implémenté en Python 3 (Vizzini, 2017). Les bibliothèques RDFLib et Graphviz sont utilisées pour charger les graphes RDF et afficher les treillis de concepts respectivement (voir Figure 1). L'implémentation utilise l'algorithme de Norris (Norris, 1978) pour extraire les concepts. L'algorithme est étendu pour traiter des couples d'identifiants dans l'extant et des couples de propriétés quantifiées et qualifiées par la clé à utiliser pour la comparaison dans l'intant. Le processus d'analyse relationnelle de concepts est implémenté en appliquant itérativement deux opérateurs d'échelonnage.

Linkky prend en entrée deux jeux de données en RDF et retourne l'ensemble des clés candidates. Le système ne prend en compte aucun alignement a priori. Il utilise aussi les mesures développées dans (Atencia *et al.*, 2014) pour déterminer les familles de clés de liage candidates compatibles.

Le processus peut donc être résumé ainsi :

1. Charger les deux jeux de données RDF ;
2. Construire la famille de contextes relationnels ;
3. Appliquer FCA aux contextes formels ;

1. `http://moex.inria.fr/software/linkky/`

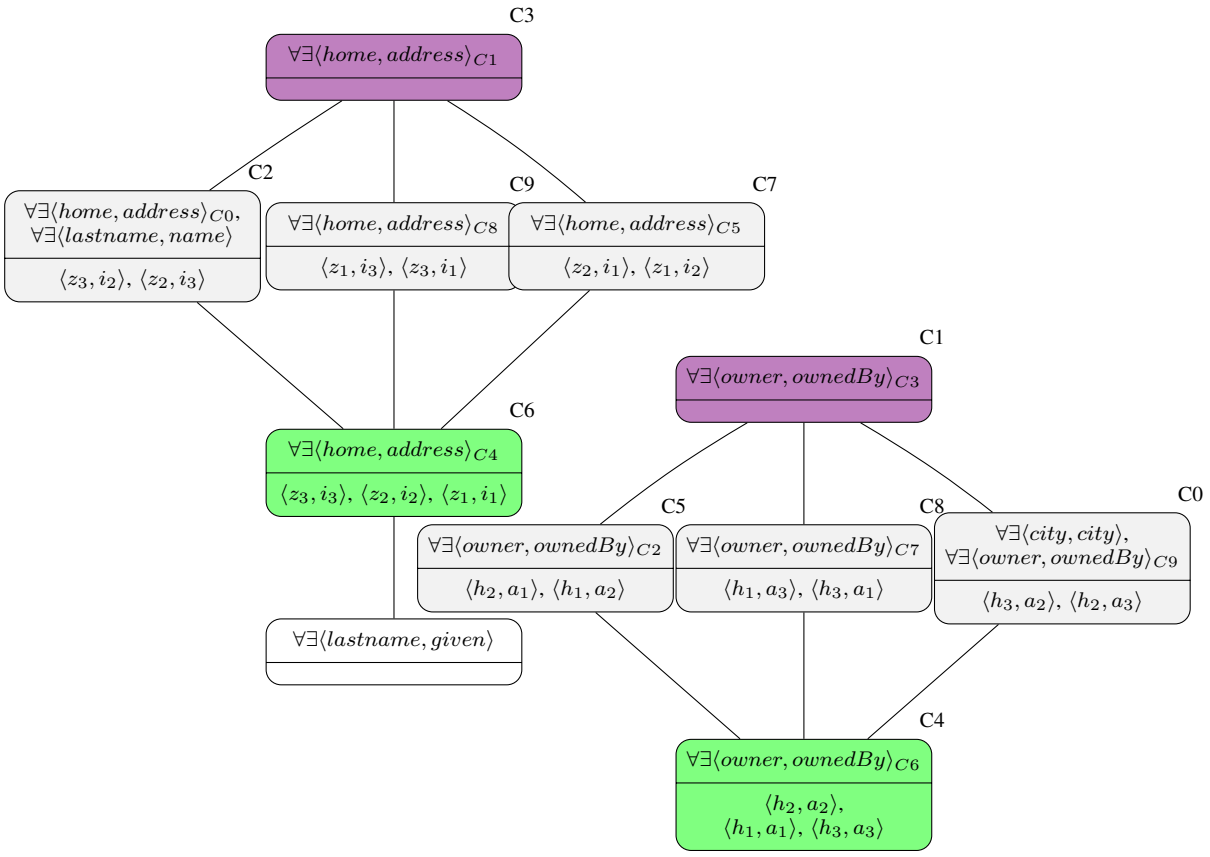

FIGURE *1 – Deux treillis de clés de liage candidates présentant des familles de clés interdépendantes (en vert et violet). Les deux clés vertes sont celles avec la meilleure évaluation (1. de couverture et de discriminabilité) et produisent le résultat espéré.*

4. Utiliser les opérateurs d'échelonnage pour introduire les nouveaux concepts créés dans les contextes formels ;
5. Si les contextes sont différents, aller en 3 ;
6. Extraire les familles de concepts compatibles (n'utilisant que des clés de la famille) ;
7. Évaluer la couverture et la discriminabilité des familles ainsi obtenues ;
8. Afficher contextes et treillis réduits.

Les différentes originalités de Linkky, outre d'être une implémentation de l'extraction de clés de liages en RDF, sont :

— il ne nécessite pas d'alignement entre les classes à considérer ;
— il peut extraire des clés entre différentes classes et une classe commune ;
— il extrait les familles de clés dépendantes ;
— il engendre directement l'affichage des treillis en LaTeX.

Remerciements

Ce travail a été financé en parti par le projet ANR Elker (ANR-17-CE23-0007-01) pour les deux premiers auteurs et par une subvention du LabEx PERSYVAL-Lab (ANR-11-LABX-0025-01) financé par le programme "Investissement d'avenir" pour Jérémy Vizzini.

Références

ATENCIA M., DAVID J. & EUZENAT J. (2014). Data interlinking through robust linkkey extraction. In *Proc. 21st European Conference on Artificial Intelligence (ECAI)*, p. 15–20 : IOS Press.

ATENCIA M., DAVID J. & EUZENAT J. (2014). What can FCA do for database linkkey extraction ? In *Proc. 3rd ECAI workshop on What can FCA do for Artificial Intelligence ? (FCA4AI), Praha (CZ)*, p. 85–92.

GANTER B. & WILLE R. (1999). *Formal Concept Analysis*. Berlin : Springer.

NORRIS E. (1978). An algorithm for computing the maximal rectangles in a binary relation. *Revue Roumaine de Mathématiques Pures et Appliquées*, **23**(2), 243–250.

ROUANE-HACENE M., HUCHARD M., NAPOLI A. & VALTCHEV P. (2013). Relational Concept Analysis : mining concept lattices from multi-relational data. *Annals of Mathematics and Artificial Intelligence*, **67**(1), 81–108.

VIZZINI J. (2017). *Data interlinking with relational concept analysis*. Mémoire de master, Université Grenoble Alpes.