Un graphe de connaissance évolutif pour la représentation d'ontologies dynamiques

Silvio Domingos Cardoso, Chantal Reynaud-Delaître, Marcos da Silveira, Cédric Pruski

Un graphe de connaissance évolutif pour la représentation d'ontologies dynamiques

Silvio Domingos Cardoso *,**, Chantal Reynaud-Delaître **, Marcos Da Silveira*, Cédric Pruski *

*Luxembourg Institute of Science and Technology
5, avenue des Hauts-Fourneaux, L-4362, Esch-sur-Alzette, Luxembourg
{silvio.cardoso, marcos.dasilveira, cedric.pruski}@list.lu
**LRI, Univ. Paris-Sud, CNRS, Université Paris-Saclay, France
chantal.reynaud@lri.fr

Résumé. Les ontologies prennent une place de plus en plus importante dans les applications informatiques que nous utilisons au quotidien. Cependant, la nature évolutive de la connaissance pousse à la révision régulière de leur contenu, conduisant à la publication de nouvelles versions de ces dernières. La relation qui relie les versions successives des concepts d'une ontologie est rarement définie. Les utilisateurs doivent par conséquent travailler avec plusieurs versions tout en garantissant une utilisation optimale de leurs applications. Pour permettre une exploitation simultanée de plusieurs versions d'ontologies, nous proposons, dans cet article, de représenter les aspects structurels et évolutifs d'une ontologie dans un graphe de connaissance. Nous analysons un cas d'étude sur l'évolution des annotations sémantiques.

1 Introduction

L'annotation des documents du Web à l'aide d'ontologies permet de rendre la sémantique de ces documents explicite pour les machines et donc une recherche d'information plus pertinente par les moteurs de recherche (Guelfi et al., 2007). Le Web des données repose aussi sur l'utilisation de vocabulaires controlés pour lier les données (Bizer et al., 2011) créant ainsi un vaste réseau de connaissances. Or, la nature évolutive de la connaissance pousse les gestionnaires des vocabulaires existants à réviser leur contenu afin de maintenir un niveau de cohérence le plus élévé possible entre les connaissances représentées dans ces ontologies et celles du monde réel. Par conséquent, les applications informatiques doivent suivre leur évolution pour une utilisation fiable (Da Silveira et al., 2015). Actuellement, la gestion de l'évolution des vocabulaires est traitée suivant la méthode des versions. Suivant cette approche, chaque fois qu'un besoin d'évolution se fait sentir, une nouvelle version du vocabulaire est publiée rendant obsolète la précédente sans pour autant spécifier le lien entre deux versions successives d'un même vocabulaire. Les utilisateurs n'ont donc aucun moyen simple de controler l'évolution et de propager ses changements.

Pour répondre à ce problème nous proposons dans cet article une approche à base de graphe de connaissance permettant de représenter les évolutions successives des concepts d'une on-

tologie et de spécifier les liens d'évolution entre ces concepts. Notre approche s'appuie sur des techniques existantes de construction de graphes dynamiques (Debatty et al., 2016) et sur les technologies du Web Sémantique pour leur représentation (Bizer et Cyganiak, 2014). La suite de cet article est structurée de la manière suivante. La section 2 présente les travaux relatifs à ceux discutés ici. La section 3 contient la description du graphe de connaissance que nous proposons, en particulier sa conceptualisation et sa construction. La section 4 décrit un cas d'utilisation du graphe de connaissance que nous proposons pour l'adaptation des annotations sémantiques en santé incluant des expérimentations et les résultats associés. La section 5 conclut l'article et présente quelques perspectives.

2 Travaux relatifs

La notion de graphe dynamique abordée dans cet article, a pour source d'inspiration la méthode *Evolving Graph Sequence* (EGS) définie par Kosmatopoulos et al. (2016). Celle-ci repose sur l'utilisation d'extraits de graphes pris à différents moments du temps pour les reconnecter et ainsi en représenter l'évolution. Des approches similaires récemment publiées sur ce sujet visent à optimiser au moins un des trois critères suivants : (i) Réduction de l'espace de stockage des extraits de graphes, (ii) recherche efficace de motifs temporels dans le graphe Moffitt et Stoyanovich (2017), (iii) amélioration de l'indexation de l'information dynamique (Labouseur et al., 2015; Akiba et al., 2014). Ainsi, Caro et al. (2015) proposent plusieurs stratégies pour compresser le contenu d'un graphe temporel en introduisant des structures de données auto-indexées. Kirsten et al. (2009) s'intéressent à la représentation de versions d'ontologies à l'aide de base de données relationnelles en ne représentant qu'une fois les éléments ontologiques invariants. Cependant, la mesure de l'incertitude entre l'évolution de ces éléments n'est pas conservée, empêchant toute évaluation de l'interprétation de l'évolution des concepts au moment de la construction des tables. De plus, la navigation dans une telle structure est lourde et peu intuitive par rapport à la nature de graphe de l'ontologie et de ses évolutions.

3 Représentation d'une ontologie et de son évolution sous forme de graphe

La représentation de l'information sous forme de graphe est utilisée par Google dans son moteur de recherche (Singhal, 2012). Depuis, l'intérêt pour les *graphes de connaissance* (GC) n'a cessé de croître. Un des avantages des graphes repose sur leur capacité pour représenter des relations complexes entre entités tout en s'appuyant sur de solides bases mathématiques et des outils informatiques efficaces.

3.1 Conception et formalisation du graphe

Nous nous intéressons à la représentation sous forme de graphe d'une ontologie et de son évolution à travers le temps. Plus particulièrement, nous nous focalisons sur les concepts d'une ontologie ainsi que sur les relations entre concepts. Nous nous restreignons aux relations hiérarchiques ansi qu'aux relations liant les concepts à travers le temps, que nous nommerons dans

la suite relations d'évolution. Le graphe de connaissance évolutif (GCE) que nous proposons est défini par un ensemble de sommets V et un ensemble d'arcs E selon la structure suivante.

$$GCE = (V, E)$$

avec

$$V = \left\{ (C_{label}, P_V, NL) \;\middle|\; \begin{array}{l} C_{label} \subseteq O, \\ P_V = (D_V, F_V), D_V, F_V \in \mathbb{N}, 0 \leq D_V \leq F_V, \\ NL = \{(c_i, RE, simi_V) | c_i \in O, \\ \qquad RE \in \{highLvl, lowLvl, none\}, \\ \qquad simi_V \in \mathbb{R}\} \end{array} \right\} (1)$$

l'ensemble des sommets du graphe et

$$E = \left\{ (u, v, P_E, R, Sim_E) \;\middle|\; \begin{array}{l} u, v \in V, \\ P_E = (D_E, F_E), D_E, F_E \in \mathbb{N}, 0 \leq D_E \leq F_E, \\ R \in \{sup, sub, sib\}, \\ Sim_E \in \mathbb{R}^+ \end{array} \right\} (2)$$

l'ensemble des arcs. Un sommet contient : le label C_{label} du concept ; une période de validité P_V pendant laquelle le concept a existé dans l'ontologie O et une liste de voisins NL. Un arc de GCE contient : les deux sommets qu'il relie, u et v ; la période de validité pendant laquelle la relation R est valide ; la relation R (structurelle ou d'évolution) entre les sommets ; et le coefficient Sim_E mesurant la similarité entre u et v (Cardoso et al., 2018). Pour déterminer les relations évolutives, nous nous appuyons sur le *Diff* entre deux versions d'une ontologie et la similarité entre les éléments du *Diff* et ceux de l'ontologie. A chaque pas d'évolution, les valeurs de similarité sont réajustées. Notez que nous typons les relations afin de bien distinguer le sens de la relation et sa nature. Ainsi, quand un concept est déplacé vers une autre partie de l'ontologie, il est encore possible de le retrouver en suivant les relation d'évolution.

3.2 Construction du graphe

La génération du GCE nécessite la première version de l'ontologie dont l'utilisateur veut contruire l'historique. Les étapes suivantes décrivent la construction du graphe :

1. **Transformation initiale :** les informations importantes pour la construction du GCE sont extraites de l'ontologie initiale comme les concepts et attributs, les relations hiérarchiques (et l'orientation des arcs) ou en sont dérivées comme les valeurs de similarité associées à chaque relation (ligne 1). Notez que les relations d'évolution ne sont pas calculés pour cette première version du GCE.
2. **Mise à jour de P_V et relations stables :** quand une nouvelle version de l'ontologie est ajoutée, nous commençons par calculer le *Diff* entre les deux dernières versions (ligne 4). Notre algorithme utilise ces résultats pour identifier la partie stable de l'ontologie. Les concepts et les relations de la partie stable auront donc une mis-à-jour (i.e.,

Algorithme 1 : Construction d'un graphe de connaissance évolutif

Input : [GC : GC_i] ;
Output : graphe de connaissance évolutif : GCE

1 $GCE \leftarrow Concept(GC_1) \cup Subclass(GC1)$
2 $i \leftarrow 2$
3 **forall** $GC_i \in GC$ **do**
4 $\quad Diff \leftarrow GC_{i-1} \setminus GC_i$
5 $\quad$ MAJPeriodeValidite(GCE,$GC_{i-1} \cap GC_i$)
6 $\quad$ **forall** $c \in Diff$ **do**
7 $\quad\quad$ Ajouter(c,GCE)
8 $\quad\quad$ $c_{old} \leftarrow IdentifierPredecesseur(c)$
9 $\quad\quad$ CreerEvolRel(GCE,c_{old},c)
10 $\quad\quad$ CalculVoisinage(GCE,c)
11 **return** GCE

incrémentation) de leur période de validité (ligne 5). La modification de la période de validité évite la duplication du concept dans le GCE pour chaque version pour laquelle ce concept existe, réduisant ainsi l'espace de stockage requis.

3. **Mise à jour des concepts modifiés :** quand un concept est modifié, nous considérons qu'une nouvelle version de ce concept est apparue. Par conséquent, la période de validité de l'ancienne version du concept n'est plus mis à jour, et la nouvelle version du concept est insérée dans le graphe (ligne 7).
4. **Mise à jour des relations (ligne 8-9) :** Nous considérons deux changements pour les relations :
   (a) la nouvelle version du concept garde les mêmes relations hiérarchiques que l'ancienne version. Dans ce cas, nous créons une relation structurelle équivalente pour le nouveau concept, nous attribuons une valeur initiale à la période de validité de cette nouvelle relation, et nous calculons la valeur de similarité entre la nouvelle version du concept et son voisinage. Il est aussi nécessaire d'établir quelle relation d'évolution existe entre les deux versions du concept.
   (b) Tous les voisins de la nouvelle version du concept ont changé. C'est le cas, par exemple, lorsqu'un concept est déplacé ou ajouté. Nous devons alors recalculer les similarités avec le voisinage.
5. **Calcul du voisinage (ligne 10) :** le voisinage est une liste de concepts qui possèdent les plus grandes similarités et potentiellement présentent une relation d'évolution avec un nouveau concept (ou une nouvelle version d'un concept). L'identifiant du concept, le type de relation d'évolution, et la valeur de similarité constituent les informations stockées sur le voisinage. Afin de calculer la similarité entre les concepts, nous utilisons une méthode hybride qui prend en compte les aspects lexicaux et sémantiques des concepts (Cardoso et al., 2018). Pour déterminer quels concepts feront partie du voisinage, nous avons implémenté l'algorithme *Fast k-NN Graph process* proposé par Debatty et al. (2016), qui consiste en :

(a) La recherche des *k* plus proches voisins d'un concept, parmi les concepts du GCE.

(b) La mise à jour de NL. La relation est décidée en fonction du changement subi par les concepts. Ainsi, si un concept a été ajouté, la relation sera du type *highLvl*. Pour le cas où une relation d'évolution a existé dans le passé, mais qu'aucun changement n'a été observé entre les deux dernières périodes, nous utilisons la relation *none*. L'intérêt d'avoir ce type de relation est de réduire le temps de parcours de graphe. L'indication qu'un lien d'évolution a existé entre deux concepts permet de trouver plus rapidement le chemin entre les termes recherchés.

4 Cas d'utilisation du graphe de connaissance

Plusieurs utilisation de notre GCE sont envisageables. La recherche d'information temporelle sur le Web comme décrit dans (Guelfi et al., 2007) est un exemple. Dans nos travaux, nous nous intéressons à l'évolution des annotations sémantiques en santé (Cardoso et al., 2016). Dans notre contexte, les annotations consistent en l'association d'un concept provenant d'une ontologie et d'une information numérique comme un extrait de texte rendant ainsi la sémantique des données annotées utilisable par les logiciels. Cependant, la grande dépendance des annotations avec les ontologies engendre des problèmes de mise à jour afin que les annotations suivent les évolutions des ontologie (Cardoso et al., 2017). Dans ce cadre, notre GCE va permettre de retrouver l'historique d'un concept donné et de mettre automatiquement à jour les annotations qui en dépendent. Dans nos travaux, nous avons adopté le standard du W3C [1] pour représenter les annotations et avons étendu ce modèle avec la relation «evol_To» pour représenter l'évolution des annotations (Cardoso et al., 2016).

4.1 Validation expérimentale du graphe de connaissance

Pour valider le bien fondé de nos GCE, nous proposons une série d'expérimentations basées sur l'évolution des annotations sémantiques dans le domaine de la santé. L'objectif est de vérifier sur un corpus d'annotations validé par des experts du domaine que notre GCE contient les bonnes évolutions des concetps associés aux annotations pour, d'une part, détecter les annotations invalides et, d'autre part, les faire évoluer de manière cohérente. Nous comparons nos résultats avec ceux obtenus par la méthode directe (BK) décrite dans (Cardoso et al., 2018).

Dans nos expériences, nous avons généré quatre GCE, un pour MeSH, un pour ICD-9-CM, un pour SNOMED CT et un pour NCIt, quatre des principales resources termino-ontologiques du domaine de la santé. Les GCE représentent l'évolution de chaque ontologie pour la période 2009-2016. Nous avons construit ces graphes en utilisant les versions AA de ces resources exprimées dans le format OWL de l'UMLS et l'outil COnto-Diff (Hartung et al., 2013) pour identifier les changements au niveau des versions successives considérées. Concernant les annotations sémantiques, nous avons utilisé un corpus de 500 ressources annotées [2] à deux moments distincts. Ces ressources ont été extraites de la campagne TREC [3] Clinical Decision Support (version 2014) et ont été annotées grace à deux annotateurs GATE (Cunningham, 2002) et NCBO annotator (Whetzel et al., 2011). Parmi les annotations générées, nous en

1. http://www.w3.org/TR/annotation-model/
2. https://git.list.lu/ELISA/AnnotationDataset
3. http://www.trec-cds.org/2014.html

Méthode	ICD-9-CM				MeSH		
	P	R	F1		P	R	F1
BK	1	0.129	0.229		1	0.050	0.094
KG	1	0.500	0.667		0.974	0.306	0.465

Méthode	NCIt				SNOMED CT		
	P	R	F1		P	R	F1
BK	1	0.115	0.207		1	0.625	0.769
KG	0.975	0.750	0.848		0.917	0.668	0.786

TAB. 1 – *Précision (P), Rappel (R) et F-Score (F1) du processus de maintenance des annotations sémantiques.*

Méthode	ICD-9-CM		MeSH		NCIt		SNOMED CT
	AUC		AUC		AUC		AUC
BK	0.597		0.545		0.663		0.75
KG	0.593		0.545		0.615		0.74

TAB. 2 – *AUC pour les méthodes directe (BK) et utilisant les GCE (KG).*

avons sélectionnées 500 (environ 125 pour chaque ontologie) avec pour seul critère le fait que le concept utilisé pour annoter le document a évolué pendant cette période. Les 2 versions des 500 annotations sélectionnées ont été validées manuellement par des experts du domaine.

4.2 Résultats expérimentaux et discussion

Nous observons dans le tableau 1 que l'utilisation de GCE fonctionne globalement mieux que l'approche directe (BK) pour la détection d'annotations sémantiques invalidées par l'évolution des concepts qui leur sont associés. La précision, le rappel et F-Score sont soit très proches (cf. Précision pour NCIt), soit meilleures lorsque nous utilisons les GCE (cf. F1 pour MeSH et ICD-9-CM). Nos GCE sont donc utiles pour la détection des annotations invalides.

Les résultats pour la maintenance des annotations identifiées comme invalides par le système du tableau 2 sont plus nuancés. L'aire sous la courbe (AUC) montre des performances proches mais légèrement supérieures pour l'approche directe (BK). Dans notre contexte, la valeur de l'AUC représente la probabilité de prendre la bonne décision pour bien migrer une annotation quelque soit l'annotation invalide considérée.

Cette méthode de validation est limitée par le nombre de cas que nous avons dans notre corpus (500 annotations). L'absence d'un corpus de référence et la disponibilité relative des experts pour la validation des annotations nous a contraint à réduire la taille du corpus pour nos expérimentations. Malgré cette limitation, nous avons pu mettre en évidence que :

— Lors de la migration des annotations, la méthode directe a étiqueté certaines adaptations d'annotations comme «irrésolues» (i.e. l'algorithme n'est pas capable de décider du nouveau concept à utiliser) et donc aucune migration n'a été appliquée. Le concept associé à l'annotation invalide n'est donc pas modifié. Or, dans notre corpus, à plu-

sieurs reprise le non changement de concept pour l'annotation coïncide avec la bonne décision. En conséquence, l'AUC concernant BK est augmentée.

— Dans certains cas, certaines annotations n'ont pas pu être migrées grace à l'approche directe car les concepts impliqués ne sont alignés avec aucun autre. Cependant, notre GCE permet de retrouver les formes morpho-syntaxiques des versions précédentes des labels de concepts qui ont été utilisés pour l'adaptation des annotations sémantiques.

5 Conclusion

Nous avons présenté une approche permettant la construction d'un graphe de connaissance évolutif pour la représentation de l'historique des évolutions successives d'une ontologie. Nous avons également montré l'utilité et le bien fondé de nos graphes sur un cas d'utilisation pour la maintenance des annotations sémantiques dans le domaine de la santé. Dans la suite de nos travaux, nous allons raffiner nos graphes en incluant notamment le type de relation d'évolution ayant conduit à une nouvelle version d'un concept (fusion de concept, éclatement de concept, etc). Nous allons aussi travailler sur la façon de determiner les relations d'évolution et envisager d'autres cas d'application comme la recherche d'information temporelle.

Références

Akiba, T., Y. Iwata, et Y. Yoshida (2014). Dynamic and historical shortest-path distance queries on large evolving networks by pruned landmark labeling. In *Proceedings of the 23rd International Conference on World Wide Web*, WWW '14, pp. 237–248. ACM.

Bizer, C. et R. Cyganiak (2014). RDF 1.1 TriG - W3C recommendation.

Bizer, C., T. Heath, et T. Berners-Lee (2011). Linked data : The story so far. In *Semantic services, interoperability and web applications : emerging concepts*, pp. 205–227. IGI Global.

Cardoso, S., C. Reynaud-Delaître, M. Da Silveira, Y.-C. Lin, A. Groß, E. Rahm, et C. Pruski (2018). Evolving semantic annotations through multiple versions of controlled medical terminologies. *Health and Technology 8*(5), 361–376.

Cardoso, S. D., M. Da Silveira, Y.-C. Lin, V. Christen, E. Rahm, C. Reynaud-Delaître, et C. Pruski (2018). Combining semantic and lexical measures to evaluate medical terms similarity. In M.-E. Vidal et S. Auer (Eds.), *Data Integration in the Life Sciences*, Lecture Notes in Computer Science. Springer International Publishing.

Cardoso, S. D., C. Pruski, M. Da Silveira, Y.-C. Lin, A. Groß, E. Rahm, et C. Reynaud-Delaître (2016). Leveraging the impact of ontology evolution on semantic annotations. In *European Knowledge Acquisition Workshop*, pp. 68–82. Springer.

Cardoso, S. D., C. Reynaud-Delaître, M. Da Silveira, et C. Pruski (2017). Combining rules, background knowledge and change patterns to maintain semantic annotations. In *AMIA Annual Symposium Proceedings*, Volume 2017, pp. 505.

Caro, D., M. A. Rodríguez, et N. R. Brisaboa (2015). Data structures for temporal graphs based on compact sequence representations. *Inf. Syst. 51*(C), 1–26.

Cunningham, H. (2002). GATE, a general architecture for text engineering. *Computers and the Humanities 36*(2), 223–254.

Da Silveira, M., J. Dos Reis, et C. Pruski (2015). Management of dynamic biomedical terminologies : current status and future challenges. *Yearbook of Medical informatics 24*(01), 125–133.

Debatty, T., P. Michiardi, et W. Mees (2016). Fast online k-nn graph building. CoRR, abs/1602.06819, arXiv.

Guelfi, N., C. Pruski, et C. Reynaud (2007). Les ontologies pour la recherche ciblée d'information sur le Web : une utilisation et extension d'OWL pour l'expansion de requêtes. In *Ingénierie des connaissances 07 (IC07)*, Grenoble, France, pp. 61–72.

Hartung, M., A. Groß, et E. Rahm (2013). Conto–diff : generation of complex evolution mappings for life science ontologies. *Journal of biomedical informatics 46*(1), 15–32.

Kirsten, T., M. Hartung, A. Groß, et E. Rahm (2009). Efficient management of biomedical ontology versions. In R. Meersman, P. Herrero, et T. Dillon (Eds.), *On the Move to Meaningful Internet Systems : OTM 2009 Workshops*, pp. 574–583. Springer Berlin Heidelberg.

Kosmatopoulos, A., K. Giannakopoulou, A. N. Papadopoulos, et K. Tsichlas (2016). An overview of methods for handling evolving graph sequences. In *Revised Selected Papers of the First International Workshop on Algorithmic Aspects of Cloud Computing - Volume 9511*, ALGOCLOUD 2015, Berlin, Heidelberg, pp. 181–192. Springer-Verlag.

Labouseur, A. G., J. Birnbaum, P. W. Olsen, Jr., S. R. Spillane, J. Vijayan, J.-H. Hwang, et W.-S. Han (2015). The G* graph database : Efficiently managing large distributed dynamic graphs. *Distrib. Parallel Databases 33*(4), 479–514.

Moffitt, V. Z. et J. Stoyanovich (2017). Towards sequenced semantics for evolving graphs. In *Proc. of the 20th International Conference on Extending Database Technology (EDBT)*.

Singhal, A. (2012). Introducing the knowledge graph : Things, not strings. `https://googleblog.blogspot.co.at/2012/05/introducing-knowledge-graph-things-not.html`. Accessed : oct 2019.

Whetzel, P. L., N. F. Noy, N. H. Shah, P. R. Alexander, C. Nyulas, T. Tudorache, et M. A. Musen (2011). Bioportal : enhanced functionality via new web services from the national center for biomedical ontology to access and use ontologies in software applications. *Nucleic acids research 39*(suppl 2), W541–W545.

Summary

The evolving nature of knowledge forces the regular update of ontologies which leads to the publication of new versions. However, the link between the successive versions of ontological elements is never specified. Users have therefore to deal with several ontology versions in order to guarantee an optimal use in the underlying systems. To overcome this barrier, we propose in this paper a knowledge graph able to represent all the versions of concepts by preserving the structural properties of the ontology. We show the added value of our graph on the maintenance of semantic annotations.