Ingénierie des Connaissances IC

SoWeDo'16

Des Sources Ouvertes au WEb de DOnnées

Atelier IC 2016

Organisateurs :

Fatiha Saïs (LRI, CNRS & Université Paris Sud, Université Paris Saclay)
Danai Symeonidou (INRA/GAMMA, Montpellier)

INRA SCIENCE & IMPACT
université PARIS-SACLAY
UNIVERSITÉ PARIS SUD
Comprendre le monde, construire l'avenir
cnrs dépasser les frontières
www.cnrs.fr

Atelier SoWeDo'16 : des Sources Ouvertes au WEb de DOnnées

Présentation

Aujourd'hui, le *Web des documents* est en cours d'évolution vers un *Web des données* où des données structurées (e.g., RDF, RDFa, MicroFormat) sont accessibles via le Web. Une initiative telle que le "Linked Open Data cloud (LOD)", consistant à publier des données RDF et à les lier les unes aux autres, est aujourd'hui un phénomène mondial qui fait émerger de nombreuses applications innovantes.

Depuis 2007, le nombre de sources de données structurées rendues disponibles sur le Web est en croissance fulgurante aboutissant à un espace global de données de l'ordre de milliards d'assertions (81 milliards[1] en janvier 2016). Dans cet espace de données, des liens sémantiques peuvent être établis entre les documents mais aussi entre les données. Ces liens permettent aux robots d'exploration, aux navigateurs ou aux applications de naviguer parmi les sources de données et de combiner les informations provenant de sources différentes. Pourtant, dans un environnement ouvert comme le Web, des URIs différentes sont créées régulièrement pour identifier le même objet. Les liens entre URIs peuvent être configurés manuellement mais, les données étant nombreuses, certaines approches s'intéressent à la génération automatique de liens entre sources de données RDF.

De plus, même si des vocabulaires reconnus existent, permettant de représenter les données sur le Web (FOAF, Dublin-Core, ...), ces vocabulaires évoluent et sont souvent insuffisants pour certains domaines d'application qui développent leur propre schéma (ou ontologie). Se pose alors le problème de l'intégration de données liées malgré l'hétérogénéité des vocabulaires utilisés. Ces données liées (ou les liens) peuvent être imprécises, périmées, fausses ou soumises à des restrictions d'usage et certaines approches s'intéressent à la provenance des données ou à leur qualité.

Enfin, différentes applications peuvent être définies pour exploiter les sources ouvertes, telles que l'application *BBC music guide* qui utilise des données liées pour obtenir une réelle valeur ajoutée dans le domaine de la musique. La méthodologie de déploiement a consisté à récolter les données d'intérêt à partir du Web pour créer un référentiel de données liées privé pour chaque application spécifique.

Dons cet atelier nous aborderons les problématiques liées à la publication de données, au liage de données et de vocabulaires mais aussi à leur capitalisation via diverses applications consommatrices de ces données et vocabulaires liés.

Pour cette nouvelle édition, l'atelier SoWeDo a souhaité s'intéresser également au traitement des grandes masses de données ("Big data"). L'explosion récente des données disponibles sur le Web a fait émerger de nouvelles problématiques visant à adapter et optimiser toute la chaîne de traitement de l'information face aux nouveaux volumes à traiter.

Les quatre articles présentés dans le cadre de cet atelier joint à la conférence IC'2016 présentent des approches traitant des problème liés aux sources ouvertes ainsi qu'aux services permettant d'exploiter leur contenu. Certaines de ces approches s'intéressent plus spécifiquement à l'hétérogénéité et à la qualité des données structurées. Le premier article présente ainsi une approche sur des données ouvertes en agronomie. En suite, Le deuxième article traite du problème de la detection et la representation des changements dans les sources RDF. Le troisième, s'intéresse à la modélisation sémantique des entrepôts de documents sémantiques. La quatrième et dernière article de cet atelier présente une évaluation de l'évolution de la complétude de DBpedia.

Enfin, nous tenons à remercier les membres du comité de programme pour leur implication dans le processus d'évaluation des articles et pour la qualité de leurs évaluations. Enfin, nous remercions Konstantin Todorov pour sa présentation en tant qu'invité à l'atelier SoWeDo 2016.

[1] `http://stats.lod2.eu/stats`

Thèmes

- Accès et collecte d'information à partir de sources ouvertes (Web, réseaux sociaux, flux RSS, etc.),
- Extraction d'information à partir de textes non structurés et/ou utilisant des vocabulaires spécifiques (blogs, langage SMS, forums, etc.), à partir de gros volumes de données multimédia (texte, image, vidéo, audio),
- Evaluation et qualification des informations extraites à partir de sources ouvertes,
- Exploitation du Crowdsourcing,
- Génération et publication des données,
- Intéropérabilité des sources de données et alignement d'ontologies,
- Inférence, découverte et validation de liens entre données,
- Fusion de données liées,
- Détection d'erreurs et résolution de conflits dans les données,
- Provenance et confiance des données et de leurs liens,
- Evolution d'ontologies, de données et de liens,
- Evaluation des requêtes dans le Web de données,
- Développement de services pour les données liées,
- Privacy/contrôle d'accès aux données liées,
- Modélisation et capitalisation des connaissances extraites à partir de sources ouvertes et du Web de données (ontologies, annotations sémantiques, etc.).

Comité de Programme

- Nathalie Abadie, IGN / COGIT
- Patrice Buche, INRA
- Olivier Corby, INRIA
- Madalina Croitoru, LIRMM, Univ. Montpellier II
- Mariana Damova, Mozaika (Bulgarie)
- Catherine Faron Zucker, Université Nice Sophia Antipolis
- Sébastien Ferre , Université de Rennes 1
- Fayçal Hamdi, Conservatoire National des Arts et Métiers
- Liliana Ibanescu, UMR MIA-Paris, AgroParisTech, INRA, Université Paris-Saclay
- Chantal Reynaud, LRI, Univ. Paris Sud, CNRS, Université Paris-Saclay
- Mathieu Roche Cirad, Cemagref - UMR TETIS
- Christian Sallaberry, University of Pau et Pays de l'Adour
- Si-Said Cherfi, CEDRIC - Conservatoire National des Arts et Métiers
- Fabian Suchanek, Télécom ParisTech University
- Danai Symeonidou, INRA, Supagro
- Maguelonne Teisseire, Cemagref - UMR Tetis

Présentation invitée

- Doing Reusable Musical Data with DOREMUS. Par *Konstantin Todorov*

Liste des articles acceptés

- AgroLD API suite: an API-centric approach towards knowledge extraction in the Agronomic Linked Data knowledge base.
  Aravind Venkatesan, Gildas Tagny Ngompe, Nordine El Hassouni, Manuel Ruiz and Pierre Larmande

- Évaluation de la qualité des liens sémantiques entre vocabulaires contrôlés.
  Melissa Mary, Lina F. Soualmia and Xavier Gansel

- Évaluation de l'évolution de la complétude de DBpedia: une étude de cas.
  Fayçal Hamdi, Samira Si-Said Cherfi and Subhi Issa

- Détection et Représentation des changements dans les sources de données RDF.
  Daniel Mercier, Nathalie Pernelle, Fatiha Saïs and Sujeeban Thuraisamy

AgroLD API suite: an API-centric approach towards knowledge extraction in the Agronomic Linked Data knowledge base

Aravind Venkatesan[1], Gildas Tagny Ngompe[1], Nordine El Hassouni[1,2], Manuel Ruiz[1,2] and Pierre Larmande[1,3]

[1] IBC Institut Computational Biology, Montpellier, France
{aravindvenkatesan}{tagnyngompe}@gmail.com

[2] AGAP, Plateforme South Green, CIRAD, Montpellier, France
{manuel.ruiz}{nordine.el_hassouni@cirad.fr}@cirad.fr

[3] DIADE, Plateforme South Green, IRD, Montpellier, France
pierre.larmande@ird.fr

Résumé : Agronomy is an overarching field constituting various research areas such as genetics, plant molecular biology, ecology and earth science. The last several decades has seen the successful development of high-throughput technologies that have revolutionised and transformed agronomic research. The application of these technologies have generated large quantities of data and resources over the web. In most cases these sources remain autonomous and disconnected. The Agronomic Linked Data project (AgroLD) is a Semantic Web knowledge base designed to integrate data from various publically available plant centric data sources. These include Gramene, Oryzabase, TAIR and resources from the South Green platform among many others. The aim of AgroLD project is to provide a portal for bioinformaticians and domain experts to exploit the homogenized data towards enabling to bridge the knowledge.

Mots-clés : Molecular Biology, Agronomy, Semantic Web, Linked Data, RDF, SPARQL, RESTFul Web Services

1 Introduction

One of the main aims of agronomic research is to effectively improve crop production through sustainable methods. To this end, there is an urgent need to over lay research findings at different scales (e.g. genomics, proteomics and phenomics). However, the information currently available in agronomy is highly distributed and diverse in nature. The Semantic Web technology (SW) offers a remedy to the fragmentation of potentially useful information on the web by improving data integration and machine interoperability. Currently, SW is playing a pivotal role in data integration and knowledge management in the biomedical domain. To further build on this line of research in agronomy, we have developed the Agronomic Linked Data (AgroLD) knowledge base. AgroLD was launched in May 2015 with an aim to serve as a portal to consolidate distributed information, facilitating formulation of research hypotheses. To this end, the RDF knowledge base integrates information from various publicly available domain specific ontologies and data sources, such as, Gene Ontology, Plant Ontology, Plant Trait Ontology, Gramene, OryGenesDB and GreenPhylDB (more information at http ://volvestre.cirad.fr :8080/agrold/documentation.jsp).

Following the semantic web convention, AgroLD can be queried using SPARQL endpoint (http ://volvestre.cirad.fr :8080/agrold/sparqleditor.jsp), however, this requires at the least a moderate knowledge of SPARQL query language. From our interactions with the community, it

was evident that the SPARQL technology may be intimidating, resulting in the knowledge base not being exploited completely. To promote the advantages of the semantic web and to encourage the end-users to utilise these resources as part of their daily research activities requires the lowering of boundaries to adopt the new approach. For instance, exploiting various ontologies to performing ontology-driven data analysis could enhance the process of new hypotheses generation that can drive new experimental design. Therefore, to assist bioinformaticians, we have developed a RESTful API suite (http ://volvestre.cirad.fr :8080/agrold/api-doc.jsp) for the programmatic retrieval of entity specific knowledge represented in AgroLD.

2 State of art

In the area of Linked Data integration some projects have demonstrated the benefits of providing data access via RESTful APIs.

The Open PHACTS Discovery Platform Plateform (Groth *et al.*, 2014), operates Linked Data to provide integrated access to a pharmaceutical knowledge bases trough a RESTful API. Data from more than 12 sources are integrated in the Open PHACTS system (Uniprot, ENZYMES, DrugBank ...). Its architecture extends the standard architecture for open and linked data applications. Some modules for the Web of data access, matching vocabulary, identity resolution and quality assessment are used to create the core of the system. This architecture facilitates iteratively and incrementally the implementation of services APIs because each module has a specific function. In addition, Open PHACTS Discovery Platform has an access control through the API Manager, 3scale [1] , and a driver for integration into the KNIMES workflow system. Access control can be important when you want to manage such aspects like (Groth *et al.*, 2014) pay services, monitoring the use of the API, or the respect of the operating policy.

SADI (Gonzalez *et al.*, 2014) (Semantic Automated Discovery and Integration) is a set of implementation principles and exposure of REST web services. Unlike the Open Phacts Discovery Platform that uses the HTTP protocol standards, SADI requires an RDF / XML file, respecting a specific semantic, to pass the arguments ("input") services and the results ("output"), also only returned in RDF / XML . It exposes web services on the web from the SHARE registry (Semantic Health And Research Environment). Thus services can be discovered by applications such as the plugin SADI-Galaxy (Aranguren *et al.*, 2014) that integrates SADI services in the Galaxy workflow environment (Giardine *et al.*, 2005).

3 Architecture

We propose an architecture consisting of systems implementing four paradigms of semantics search systems Haag *et al.* (2014) . Each of these paradigms has their strengths but also their limits that we could reduce combining them together :

1. Text search queries
2. Visual query language
3. Assisted SPARQL queries
4. Form based queries

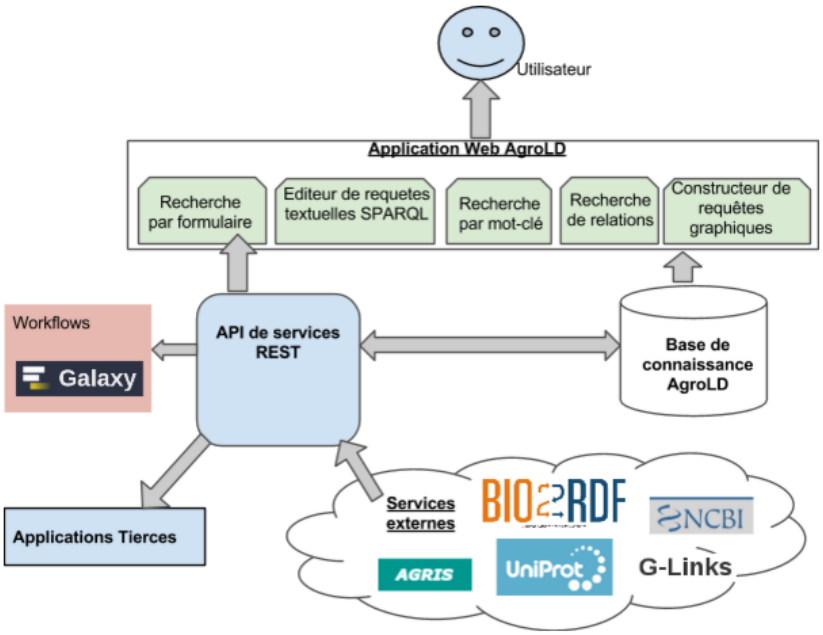

FIGURE 1 – Overall architecture developed for AgroLD

The API includes queries corresponding to biological questions. It is used when using the form and running the Galaxy workflows. The API can also query external services to complement the results found in AgroLD. Thus the AgroLD knowledge base can be expanded during query execution.

Third party applications can be developed based on the services API. Our case study here is the search form that uses the API. We offer a form search based on iterative and exploratory model defended by Uren *et al.* (2007) for better usability. This is indeed a search that offers to the user new search possibilities based on the results of previous search. Thus the user can find more information on his search.

4 Conclusion

The current version of the API suite (ver. 1) can be used to retrieve gene and protein information, metabolic pathways, proteins associated with ontological terms, to name a few. This is achieved by querying by name or the ID of the corresponding entity. Further, the APIs can be launched as part of a workflow environment (the APIs have been tested with Galaxy workflow tool). Additionally, to aid non-technical users in the exploration of the knowledge base, we have developed an 'Advanced search' query form. The query form is based on the API suite (http ://volvestre.cirad.fr :8080/agrold/advancedSearch.jsp), thus concealing the SPARQL layer, allowing biologists query AgroLD. The API suite is under active development and the subsequent versions will include querying other entities represented in AgroLD (e.g. : pheno-

1. www.3scale.net

types, genomic annotation and homology information). We believe that the API suite will aid in bringing semantic web technology closer to the domain experts. In the proposed paper, we will describe and demonstrate the utility of the APIs with various use cases.

Références

ARANGUREN M. E., GONZÁLEZ A. R. & WILKINSON M. D. (2014). Executing SADI services in Galaxy. *Journal of biomedical semantics*, **5**(1), 42.

GIARDINE B., RIEMER C., HARDISON R. C., BURHANS R., ELNITSKI L., SHAH P., ZHANG Y., BLANKENBERG D., ALBERT I., TAYLOR J., MILLER W., KENT W. J. & NEKRUTENKO A. (2005). Galaxy : A platform for interactive large-scale genome analysis. *Genome Research*, **15**(10), 1451–1455.

GONZALEZ A., CALLAHAN A., CRUZ-TOLEDO J., GARCIA A., EGANA ARANGUREN M., DUMONTIER M. & WILKINSON M. D. (2014). Automatically exposing OpenLifeData via SADI semantic Web Services. *J Biomed Semantics*, **5**(1), 46.

GROTH P., LOIZOU A., GRAY A. J., GOBLE C., HARLAND L. & PETTIFER S. (2014). API-centric Linked Data integration : The Open PHACTS Discovery Platform case study. *Web Semantics : Science, Services and Agents on the World Wide Web*, **29**(0), 12 – 18. Life Science and e-Science.

HAAG F., LOHMANN S., BOLD S. & ERTL T. (2014). Visual SPARQL Querying based on Extended Filter/Flow Graphs. In *Proceedings of the 12th International Working Conference on Advanced Visual Interfaces (AVI '14)*, p. 305–312 : New York, NY, USA : ACM.

UREN V., LEI Y., LOPEZ V., LIU H., MOTTA E. & GIORDANINO M. (2007). The usability of semantic search tools : a review. *The Knowledge Engineering Review*, **22**(04), 361–377.

Évaluation de la qualité des liens sémantiques entre vocabulaires contrôlés

Mélissa Mary[1,2], Lina F. Soualmia[2,3] et Xavier Gansel[1]

[1] bioMérieux SA, Département Développement et Intégration,
38390 La Balme Les Grottes
{melissa.mary,xavier.gansel}@biomerieux.com
http://biomerieux.com
[2] LITIS EA 4108 et NormaSTIC CNRS 3638, Université de Normandie, 76000 Rouen
Lina.Soualmia@chu-rouen.fr
[3] LIMICS INSERM UMR_1142, Sorbonne Universités, 75000 Paris

Résumé : L'informatisation des données de santé doit relever le défi de l'interopérabilité syntaxique, mais surtout sémantique, entre les systèmes d'information et les Systèmes d'Organisation des Connaissances (SOC) sur lesquels ils reposent. L'alignement entre SOC est une solution qui répond en partie à cette problématique et il s'avère nécessaire d'évaluer la qualité des alignements produits. Nous proposons dans cet article des méthodes d'évaluation d'alignement de concepts issus du diagnostic in vitro (DIV) présents dans les SOC de référence disponibles en ligne. Les méthodes proposées reposent sur trois mesures de similarité syntaxique et un algorithme à base d'heuristiques. Les résultats que nous obtenons dans cette étude montrent que les métriques de similarité syntaxique ne se révèlent pas suffisamment probantes pour se voir appliquées de manière systématique au domaine des tests de laboratoire. En revanche, la qualité des alignements obtenus via l'algorithme heuristique, filtré a posteriori en fonction d'une dimension sémantique, permettent de conforter les critères de performance que nous avions établis. Cet algorithme est notre piste privilégiée pour obtenir des alignements de qualité dans le domaine du DIV. Il est en cours d'amélioration par l'enrichissement, à la volée, d'informations syntaxiques et sémantiques.

Mots-clés : algorithme d'alignement, domaine de la santé, évaluation, interopérabilité sémantique, systèmes d'organisation des connaissances.

1 Introduction

La centralisation des données du patient dans un répertoire électronique est gérée par les instituts de santé publique afin d'améliorer la prise en charge du patient et de maîtriser les coûts médicaux (Fieschi, 2009; Macary, 2007; Stroetmann, 2009). L'objectif de ces répertoires uniques est de les rendre accessibles et éditables par l'ensemble des acteurs de la chaîne de soins. De ce fait, l'interopérabilité entre les différents systèmes d'information utilisés par les professionnels de santé est un enjeu majeur dans la mise en place de ces dossiers électroniques. L'interopérabilité peut se décliner en plusieurs axes et notamment par une dimension sémantique. L'emploi de vocabulaires standards pour coder l'information au sein des dossiers patients fait partie des solutions pour résoudre la problématique d'interopérabilité sémantique. Ces vocabulaires spécialisés dans un domaine peuvent être représentés sous forme de terminologie, thesaurus, ou ontologie que nous désignerons sous le terme général de Système d'Organisation des Connaissances (SOC) (Hodge, 2000). Dans le domaine du diagnostic *in vitro* (DIV) deux SOC sont recommandés par les instances de standardisation nationales et internationales pour codifier les informations du laboratoire

(Blumenthal, 2010; Stroetmann, 2009) : la terminologie LOINC®[1] (Logical Observation Identifiers Names and Codes) qui est préconisée pour décrire les tests de laboratoire, et l'ontologie SNOMED CT® (Systematized Nomenclature of MEDicine – Clinical Terms) qui permet d'exprimer les résultats obtenus par ces tests. Pour interpréter le résultat obtenu il est important de connaître la nature du test réalisé. Les informations exprimées par la terminologie LOINC® et l'ontologie SNOMED CT® dans les dossiers patients sont donc interdépendantes, ce qui conduit à l'utilisation conjointe de ces SOC pour interroger et agréger les données d'un compte rendu d'analyse. Une collaboration a récemment été mise en place (IHTSDO & Regenstrief Institute, 2013; Vreeman, 2015) afin d'aligner LOINC® et SNOMED CT® et ainsi améliorer l'interopérabilité des données de comptes rendus de laboratoire. De plus en plus d'initiatives visent à réaliser des alignements entre SOC dans le domaine clinique, mais également sur des SOC disponibles sur le web. L'obtention des alignements est un défi majeur tant du fait de la volumétrie des données à aligner, de l'hétérogénéité de la représentation des concepts, que de la qualité attendue des alignements. De nombreuses méthodes, stratégies et métriques ont été développées permettant de réaliser des alignements notamment entre ontologies (Brahma & Refoufi, 2015; Euzenat & Shvaiko, 2013; Shvaiko & Euzenat, 2005) ou encore gérer l'évolution des alignements entre concepts (Dos Reis *et al.*, 2015). Dans cet article, notre étude vise à évaluer trois métriques de similarité et un algorithme heuristique sur des données de DIV. En tenant compte des caractéristiques des SOC et l'emploi de ces alignements, nous avons établi plusieurs critères d'évaluation pour discriminer ces méthodes. L'évaluation est possible grâce à des alignements entre LOINC® et SNOMED CT® réalisés par des experts (IHTSDO & Regenstrief Institute, 2013) et disponibles sur le web.

Cet article est organisé comme suit : dans la section 2 nous décrivons les données utilisées ainsi que les méthodes que nous proposons. La section 3 présente les critères d'évaluation, les principaux résultats et une discussion autour de la pertinence des méthodes évaluées pour l'alignement de SOC. La section 4 conclut cette étude et donne quelques pistes de recherches.

2 Matériel et Méthode

2.1 Matériel

Logical Observation Identifiers Names and Codes

La terminologie LOINC® a été construite en 1994 afin de standardiser la description des tests cliniques et de diagnostic *in vitro.* Elle est développée et mise à jour deux fois par an par le *Regenstrief Institute* (Sheide & Wilson, 2013). Un test codé en LOINC® se décompose en 6 dimensions. Les dimensions *Composant*, *Milieu*, *Technique* et *Temps* décrivent le principe du test de laboratoire alors que les dimensions *Échelles* et *Grandeur* caractérisent le type de résultat. Les valeurs permettant d'exprimer chacune de ces dimensions (nommées « parties » par la suite) sont organisées hiérarchiquement. Afin de compacter la description d'un test, les parties *Milieu*, *Échelles* et *Grandeurs* sont représentées par des codes mnémoniques. Par exemple le mot « *blood* » est représenté par le code mnémonique « bld » dans la partie *Milieu*. Les parties représentant des *Composant* ou des *Technique* peuvent contenir des abréviations (par exemple « Ab » pour « *Antibody* » ou « EIA » pour « *Elisa Immuno Assay* »).

Dans cette étude nous utilisons la version 2.5 de LOINC® décrivant plus de 68 000 tests composés par 40 000 parties. Les parties et leur hiérarchie sont extraites de la version 2.5 de la base de données utilisée par l'application RELMA[1].

Systematized Nomenclature Of MEDicine – Clinical Terms

1 http://loinc.org/

La SNOMED CT® est une ontologie sous licence créée et maintenue deux fois par an par l'*International Health Terminology Standards Development Organisation* (IHTSDO)[2] (Cornet & de Keizer, 2008; Lee, de Keizer, Lau, & Cornet, 2013).

L'ontologie SNOMED CT® permet de représenter le domaine clinique avec plus de 350 000 concepts organisés en 19 axes. Les concepts sont identifiés par un libellé unique (*Fully Specified Name*) qui se compose d'un terme spécifiant la sémantique du concept et d'un tag sémantique. Le tag sémantique est placé à la fin du libellé et apporte une information contextuelle sur la classification dans SNOMED CT® et l'utilisation du concept. Par exemple le terme *Penicillin* est utilisé pour identifier deux concepts (*373270004 |Penicillin -class of antibiotic- (substance)|* et *6369005 |Penicillin -class of antibiotic- (product)|)* appartenant respectivement aux axes *105590001 |Substance (substance)|* et *373873005 |Pharmaceutical / biologic product (product)|*. Cette étude a été réalisée à l'aide de la version de SNOMED CT® datant de janvier 2015[3].

Alignement entre LOINC® et SNOMED CT®

Les alignements entre LOINC® et SNOMED CT® utilisés dans cette étude sont le résultat d'une collaboration initiée en 2013 entre l'IHTSDO et le *Regenstrief Institute* (IHTSDO & Regenstrief Institute, 2013). La collaboration a pour objectif d'aligner la description des tests LOINC® sur des concepts SNOMED CT® pour améliorer l'agrégation de données dans les dossiers patients informatisés (Vreeman, 2015). L'alignement de LOINC® sur SNOMED CT® est réalisé à deux niveaux. Dans un premier temps les parties décrivant les dimensions d'un test ont été alignées sur des concepts SNOMED CT®. Par la suite les tests LOINC® sont décrits par des définitions formelles en SNOMED CT®. Dans cette étude nous utilisons la première version d'alignement publiée en septembre 2014, qui couvre 0,15% des tests LOINC® et 2 115 parties[4]. L'alignement partie-concepts n'est pas bijectif. Sur les 2 177 alignements, 62 parties LOINC® sont alignées sur plusieurs concepts et 92 concepts sont alignés avec plus d'une partie LOINC®.

2.2 Méthodes

Les méthodes présentées ont été implémentées dans le langage R (R. Core Team, 2014).

2.2.1 Similarités syntaxiques

La similarité syntaxique est un score calculé par comparaison des chaînes de caractères. Nous avons étudié trois méthodes de calcul de ce score qui sont présentées dans les sections suivantes, le tableau 1 illustre les similarités obtenues par les trois méthodes.

TABLE 1 *Exemple de similarités obtenues par les méthodes DL, Stoilos et WGram.*

	DL	**Stoilos**	**WGram**
Terme 1 : nitrite	0,29	0,84	**terme1→terme2** : 1
Terme 2 : **nitrite salt**			**terme 2 →terme1** : 0,36

Damerau Levenshtein

La similarité de Damerau Levenshtein (Damerau, 1964; Levenshtein, 1966)(DL) se calcule à partir de la distance d'édition du même nom disponible dans le package stringdist (Van der Loo, 2014). Elle repose sur l'équation suivante :

$$sim_{DL}(t_1, t_2) = 1 - \frac{dist_{DL}(t_1, t_2)}{min(|t_1|, |t_2|)} \qquad (1)$$

2 http://www.ihtsdo.org/

3 https://uts.nlm.nih.gov/home.html

4 https://loinc.org/news/draft-loinc-snomed-ct-mappings-and-expression-associations-now-available.html/

Où t_1, t_2 représentent deux termes et $|t_1|$ $|t_2|$ leur longueur.

Stoilos

La similarité de Stoilos est une métrique développée spécifiquement pour les libellés de concepts d'ontologies (Stoilos et al., 2005). Cette similarité pondère positivement les termes ayant un préfixe commun (fonction *winkler*). Elle repose sur l'idée que la similitude entre deux chaînes est liée à leurs points communs ainsi qu'à leurs différences. La distance de Stoilos entre deux chaînes de caractères t_1 et t_2 est définie par la formule suivante :

$$sim_{Stoilos}(t_1, t_2) = common(t_1, t_2) - diff(t_1, t_2) + winkler(t_1, t_2) \quad (2)$$

$$common(t_1, t_2) = \frac{2 * \sum_{i=1}^{n} |substring(t_1, t_2)|_i}{|t_1| + |t_2|} \quad (3)$$

$$diff(t_1, t_2) = \frac{diff_1 * diff_2}{p_{diff} + (1 - p_{diff})(|diff_1| + |diff_2| - |diff_1| * |diff_2|)} \quad (4)$$

$$winkler(t_1, t_2) = p_{winkler} * (1 - common(t_1, t_2)) * min(4, |common_{prefix}|) \quad (5)$$

Où *common(t_1, t_2)* (équation 3) représente la communauté entre les deux chaînes de caractères t_1 et t_2, *diff(t_1, t_2)* (équation 4) la différence entre t_1 et t_2, et *winkler(t_1, t_2)* permet l'amélioration du résultat en utilisant la méthode introduite par Winkler (équation 5) (Winkler, 1999).

WGram

La méthode WGram que nous avons développée, s'appuie sur la décomposition des termes en vecteur de mots (w) et produit pour un alignement deux scores de similarité (terme 1 vers terme2 et terme2 vers terme1). La similarité d'un terme par rapport à un autre (équation 6) se calcule par le biais des similarités des mots qu'ils ont en communs. Dans cette étude l'alignement des mots (w) composant les termes du SOC 1 et ceux du SOC 2 sont obtenus par calcul de similarités de Damereau Levenshtein (DL).

$$sim_{WGram}(t_1 \rightarrow t_2) = \frac{\sum_{i=1}^{K} |w_1^i| * sim_{DL}(w_1^i, w_2^i)}{\sum_{i=1}^{N} |w_1^i|} \quad (6)$$

Où K représente le nombre de mots alignés (w^i) entre les termes t_1 et t_2, N représente le nombre de mots (w^i_1) composant t_1 et w^i_2 un mot composant le terme t_2.

2.2.2 Similarité sémantique

La similarité sémantique est obtenue à partir du Metathesaurus® développé par l'US National Library of Medicine (Fung & Bodenreider, 2005)[3]. Le Metathesaurus® est construit à partir de 195 ressources biomédicales dont LOINC® et SNOMED CT® dans 25 langues (version 2015AB). Les 13 millions de termes issus des différentes ressources sont représentés par 3 millions de concepts identifiés de manière unique par un code (CUI) dans le Metathesaurus®. Chaque concept du Metathesaurus® est associé à un (ou plusieurs) type(s) sémantique(s) issu(s) du réseau sémantique de l'UMLS® (McCray, 1989). L'UMLS® est très souvent employé pour indexer et aligner des SOC médicaux (Dolin, Huff, Rocha, Spackman, & Campbell, 1998; Fung & Bodenreider, 2005).

Extraction des informations sémantiques d'un terme à partir de l'UMLS

L'extraction des concepts et le calcul de la similarité sémantique sont réalisés à partir des résultats de MetaMap (Aronson & Lang, 2010) qui est un outil développé pour l'indexation de textes avec des concept issus du Metathesaurus® (Pratt & Yetisgen-Yildiz, 2003). La figure 1-A illustre les résultats bruts de MetaMap pour la partie LP66730-0 LOINC® « *Parrot serum proteins feathers dropping Ab* ». MetaMap segmente le texte en unités sémantiques (notées *Phrase* dans la figure) alignés sur un ou plusieurs concept de l'UMLS®

(*MetaMapping*). La confiance de l'alignement entre l'unité sémantique dans le texte et le concept UMLS® se mesure par un score d'alignement dans l'intervalle [0 ; 1000].

Nous avons développé un algorithme qui permet, pour un terme indexé par MetaMap, de retourner la liste des termes sémantiques candidats. Nous définissons un terme sémantique candidat comme étant la combinaison de concepts UMLS® chacun d'eux représentant une unité sémantique distincte. À chaque terme nous associons une confiance sémantique qui correspond à la moyenne des scores d'alignement. La partie LP66730-0 est représentée sur l'UMLS® par deux termes sémantiques candidats, chacun étant composés par une combinaison distinct de trois CUIs et une confiance sémantique de 898 (illustrée dans la figure 1-B).

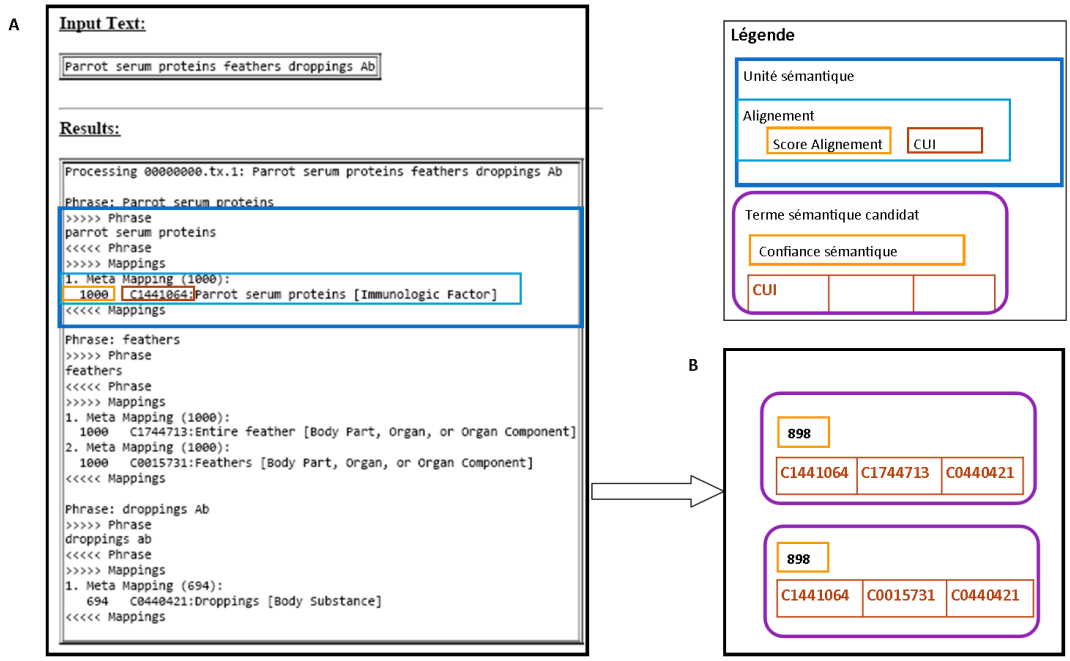

FIGURE 1 Schéma d'extraction des informations sémantiques pour la partie LP66730-0. **A** : Fichier de sortie MetaMap. **B** : Liste des termes sémantiques candidats extraits du fichier de résultat.

Similarité sémantique entre deux termes

Le calcul de la similarité sémantique entre deux termes (t_1 et t_2) se détermine de manière indirecte à travers l'alignements de leurs termes sémantiques candidats (t^i_{sem1}, t^j_{sem2}). Nous définissons la similarité sémantique entre deux termes t_1 et t_2 comme étant la similarité de hamming ($sim_{hamming}$) maximal calculé pour toutes les combinaisons de termes sémantiques candidats (t^i_{sem1}, t^j_{sem2}) représentant respectivement les termes t_1 et t_2 (équation 7). Le calcul de la similarité de hamming est réalisé sur les vecteurs de CUIs de t^i_{sem1}, t^j_{sem2}.

$$sim_{sem}(t_1, t_2) = max(sim_{hamming}(t'_{sem1}, t'_{sem2})) \quad (7)$$
$$t'_{sem1} \in \lfloor t_{sem} \rfloor_1, t'_{sem2} \in \lfloor t_{sem} \rfloor_j$$

Nous mesurons la confiance de cette similarité sémantique comme étant le minimum entre la confiance sémantique (figure 1-B) des termes sémantiques candidats (t^i_{sem1}, t^j_{sem2}) ayant permis l'alignement de t_1 et t_2.

$$conf_{sim.sem}(t^i_{sem1}, t^j_{sem2}) = min(conf_{sem}(t^i_{sem1}), conf_{sem}(t^j_{sem2})), \quad (8)$$

Paramètres utilisés pour MetaMap

Pour cette étude, nous considérons chacun des termes des deux SOC comme étant un texte et nous effectuons la recherche de concepts sur une version modifiée du Metathesaurus® (2014AB) qui exclut les ressources LOINC® et SNOMED CT® afin de « simuler » la pertinence d'un alignement sémantique entre deux ressources dans le cas où celles-ci ne seraient pas incluses dans le Metathesaurus®.

2.2.3 Alignement heuristique par la méthode des ancres

La méthode des ancres que nous proposons est une stratégie d'alignement heuristique, initiée à partir d'un alignement préexistant entre les deux SOC. Le principe de cette méthode s'inspire d'un algorithme développé pour résoudre les problématiques d'alignement d'ontologies volumineuses (Seddiqui & Aono, 2009). Contrairement à l'algorithme proposé par Seddiqui et Aono, nous calculons les nouvelles ancres *via* des similarités entre termes sans prendre en compte les schémas sous-jacents.

Cette méthode part du postulat que les deux ressources à comparer reposent sur une organisation des données similaires. La probabilité d'obtenir de « vrais » alignements augmente si les termes que l'on cherche à aligner disposent d'un contexte similaire, c'est-à-dire s'il existe au moins un de leurs termes respectifs parents, enfants ou frères qui sont alignés. Le principe de la méthode consiste à étendre les alignements du jeu initial (les ancres initiales) grâce à la classification des données au sein des SOC. L'expansion de l'alignement est réalisée de manière récursive par comparaison des termes parents de termes t_1 et t_2, enfants de t_1 et t_2 et frères de t_1 et t_2. La comparaison des termes peut être effectuée avec l'ensemble des métriques syntaxiques et sémantique présentées dans les sections précédentes. Dans cette étude nous utilisons la méthode de DL pour comparer les termes, et nous filtrons les alignements en fonction d'un seuil de similarité.

2.3 Évaluation

L'alignement ($\mathcal{A}$) entre les SOC ($\mathcal{T}_1$et $\mathcal{T}_2$) est évalué grâce à un alignement de référence (noté $\mathcal{M}$) par les paramètres de précision (équation 9) et de rappel (équation 10).

$$precision(\mathcal{A}_{\mathcal{T}_1,\mathcal{T}_2}, \mathcal{M}_{\mathcal{T}_1,\mathcal{T}_2}) = \frac{|\mathcal{A}_{\mathcal{T}_1,\mathcal{T}_2} \cap \mathcal{M}_{\mathcal{T}_1,\mathcal{T}_2}|}{|\mathcal{A}_{\mathcal{T}_1,\mathcal{T}_2}|} \quad (9)$$

$$rappel(\mathcal{A}_{\mathcal{T}_1,\mathcal{T}_2}, \mathcal{M}_{\mathcal{T}_1,\mathcal{T}_2}) = \frac{|\mathcal{A}_{\mathcal{T}_1,\mathcal{T}_2} \cap \mathcal{M}_{\mathcal{T}_1,\mathcal{T}_2}|}{|\mathcal{M}_{\mathcal{T}_1,\mathcal{T}_2}|} \quad (10)$$

2.4 Normalisation des termes et filtre des alignements

2.4.1 Normalisation des termes

Nous avons défini une méthode de normalisation des termes par SOC. Dans les parties LOINC® les éléments de ponctuations sont supprimés ou remplacés par des espaces. Les libellés des concepts SNOMED CT® sont normalisés au niveau de la ponctuation et le tag sémantique est supprimé.

2.4.2 Paramètres de filtre des alignements

Nous utilisons deux filtres pour étudier les alignements calculés à partir des similarités syntaxiques. Par défaut, un alignement entre deux SOC ($\boldsymbol{\mathcal{T}}_1et\boldsymbol{\mathcal{T}}_2$) est composé des meilleurs alignements par terme du SOC 1 et des meilleurs alignements par terme du SOC 2 (équation 11).

$$\mathcal{A}(\mathcal{T}_1, \mathcal{T}_2) = \{\mathcal{A}(t_x, t_y) | sim(t_x, t_y) = max(sim(t_x, \mathcal{T}_2)) \vee sim(t_x, t_y) = max(sim(\mathcal{T}_1, t_y))\} \quad (11)$$

Le filtre `BestForBoth` (équation 12) que nous proposons permet de sélectionner les alignements qui sont les meilleurs à la fois pour t_1 et pour t_2.

$$BestForBoth(\mathcal{T}_1, \mathcal{T}_2) = \{\mathcal{A}(t_x, t_y) | sim(t_x, t_y) = max(sim(t_x, \mathcal{T}_2)) \\ = max(sim(\mathcal{T}_1, t_y))\} \quad (12)$$

Les résultats de ces deux filtres sont contextuels. Ils sont fonction à la fois de la métrique utilisée mais surtout des deux ensembles de termes utilisés pour l'alignement.

3 Résultats et discussion

L'objectif de cette évaluation est d'identifier les méthodes et métriques et les plus performantes pour l'alignement de termes spécifiques au domaine du diagnostic *in vitro*. Les SOC représentant des données de DIV ont trois caractéristiques majeures qui impactent directement les critères de performance requis pour un alignement. La première caractéristique, et la plus critique, concerne la qualité des ressources. Afin d'être mis en œuvre dans les systèmes, les alignements produits doivent être les plus précis possible, c'est-à-dire que la méthode doit maximiser le nombre de vrais alignements. La seconde caractéristique est relative à la quantité de données intégrée dans les SOC. Enfin, la troisième caractéristique concerne l'hétérogénéité des champs lexicaux en fonction du sous-domaine de connaissances représentés dans les SOC. Par exemple, les termes associés à la dénomination des microorganismes suivent une nomenclature stricte alors que le vocabulaire pour exprimer un milieu est plus riche en variation linguistique. Les métriques et méthodes sont donc évaluées sur deux critères :

- une précision maximale pour un nombre d'alignements cohérent avec l'ordre de grandeur du plus petit SOC aligné ;
- la robustesse des métriques pour des sous-domaines de connaissance dont l'expression des termes peut être plus ou moins permissive.

3.1 Évaluation des méthodes de similarité syntaxiques

3.1.1 Analyse des similarités syntaxiques

Tout d'abord il faut noter que pour utiliser la méthode DL, les parties LOINC® et SNOMED CT® ont été normalisées. Ces étapes de normalisation supposent des connaissances *a priori* sur la construction des termes au sein de ces SOC. Bien que différentes entre LOINC® et SNOMED CT®, elles sont appliquées de manière uniforme sur les deux SOC et ne dépendent pas des sous-domaines étudiés. Ces normalisations ne vont donc pas à l'encontre du critère de robustesse établi ci-dessus.

Le tableau 2 résume l'évaluation des alignements obtenus par les métriques de similarité syntaxique. On peut observer que les méthodes DL, Stoilos et WGram génèrent deux fois plus d'alignements qu'il n'existe de vrais alignements (2 177 au total). Après application du filtre `BestForBoth` sur les alignements DL et Stoilos, on peut constater une augmentation significative de la précision (+50%) au détriment du nombre de vrais alignements retrouvés. L'application de ce filtre sur la méthode WGram a peu d'impact sur la précision de l'alignement (+ 15%) et sur le rappel.

Dans un second temps nous avons cherché à combiner les métriques pour maximiser la précision. Les meilleurs résultats sont obtenus par l'intersection des alignements de WGram et ceux de Stoilos. On observe que cette intersection permet de retrouver 50% des vrais alignements avec une précision de 95%. Nous pouvons en conclure que cette combinaison permet d'obtenir des résultats satisfaisant nos critères de qualité.

***TABLE 2** Résultat de l'évaluation des métriques de similarité syntaxique.*

Méthode	Normalisation des termes	Filtre	Nombre d'alignements	Précision	Rappel
DL	Oui		4 589	0,26	0,54
	Oui	`BestForBoth`	1 281	0,77	0,45
Stoilos (p_{diff} = 0.6 ; $p_{winkler}$ = 0.1)	Non		3 132	0,50	0,72
	Non	`BestForBoth`	**1 356**	0,92	0,57
WGram	Non		3 263	0,47	0,71
	Non	`BestForBoth`	2 322	0,63	0,67
WGram et Stoilos	Non	`BestForBoth pour la métrique Stoilos`	1 202	**0,95**	0,53
WGram ou Stoilos	Non	*`BestForBoth`* `pour la métrique Stoilos`	3 497	0,47	**0,76**

3.1.2 Comparaison des temps de calcul entre les métriques

Lors du développement des algorithmes d'alignement nous avons remarqué des temps d'exécution très hétérogènes pour les métriques DL, Stoilos et WGram (voir tableau 3). On observe un facteur 10 entre le temps d'exécution de la métrique DL (~7) et celui de WGram (~60) sur l'alignement de 1000 termes LOINC® et 1000 termes SNOMED CT® (1 million de combinaisons) pris aléatoirement dans le jeu d'alignement initial. On observe également que l'ajout du filtre `BestForBoth` n'a pas d'impact significatif sur le temps de calcul des alignements. Si l'obtention rapide des résultats ne fait pas partie des critères que nous avions identifiés initialement, il est important de le prendre en compte dans cette étude. En effet comme mentionné dans la partie Matériel, les SOC LOINC® et SNOMED CT® sont très volumétrique et mis à jour deux fois par an, ce qui entraîne pour chaque mise à jour de SOC la recompilation d'un nouvel alignement. Ces mises à jour ne concernent pas que des ajouts ou suppression de concept, mais également des changements de termes. Dans certains domaines du diagnostic *in vitro*, comme la taxonomie bactérienne, les termes évoluent mensuellement. Les curateurs évaluent pour 3000 espèces et genre recensés, entre 2 à 30 changements de noms par mois. Pour aligner ces domaines, il faut donc mettre en œuvre des algorithmes avec de bonnes performances en temps de calcul et qui peuvent être itérés tous les mois si nécessaire.

***TABLE 3** Temps de calcul (CPU total) des alignements LOINC® et SNOMED CT® (10 000 combinaisons) en fonction de la métrique utilisée avec et sans le filtre BestForBoth.*

Méthode	Temps CPU total sans filtre	Temps CPU total avec `BestForBoth`
DL	6,80	7,19
Stoilos (p_{diff} = 0.6 ; $p_{winkler}$ = 0.1)	39,17	39,08
WGram (index des mots)	58,5 (5,99)	57,58 (5,99)

3.1.3 Comparaison des performances du filtre BestForBoth avec le filtre par seuil

Les performances des méthodes de similarité sont généralement étudiées avec un paramètre de seuil comme filtre d'alignement. Dans cette étude nous avons décidé de ne pas

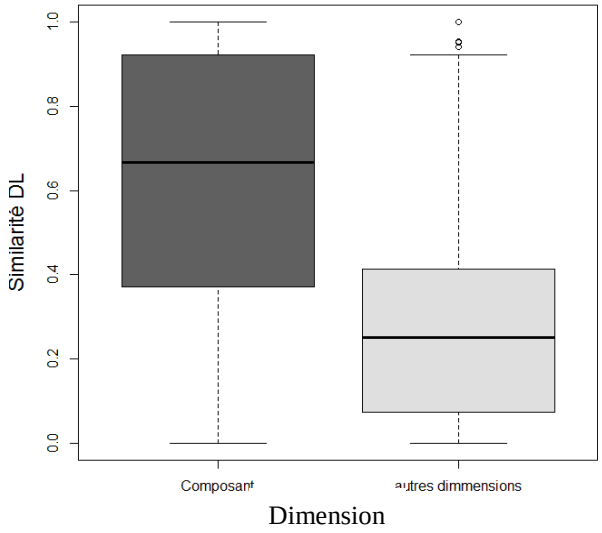

FIGURE 2: *Distribution de la similarité calculé avec la métrique DL en fonction des dimensions LOINC.*

investiguer les performances associées à ce type de filtre. Ce choix est motivé par une question majeure : Comment déterminer sans *a priori* le seuil de filtrage ?

Comme expliqué précédemment, la variabilité de la représentation terminologique des concepts dépend fortement du sous-domaine étudié. Le tableau 4 illustre cette problématique avec trois exemples retrouvés fréquemment dans les alignements. On observe que le filtre `BestForBoth` permet une meilleure discrimination des vrais alignements. Pour le sous-domaine des organismes, les termes *Babesia bovis* et *Babesia ovis* ne représentent pas les mêmes taxa malgré une forte similarité (0,92) alors que les termes *Leukocytes* et *Leukocyte* représentent le même concept avec une similarité plus faible (0,89). La figure 2 illustre la distribution de la similarité en fonction des dimensions LOINC®. On observe un écart de moyenne de 0,3 entre les alignements obtenus pour des composants (termes peu abrégés dans LOINC®) et les autres alignements souvent abrégés ou représentés sous forme mnémonique. L'emploi d'un seuil de similarité pour filtrer les alignements sur les similarités syntaxiques doit être paramétré en fonction de la variabilité lexicale du sous-domaine étudié. Avec un paramétrage *a priori* du seuil en fonction des sous-domaines étudiés, le critère de robustesse n'est pas respecté. Il faudrait donc être capable de paramétrer le seuil sans *a priori*, en utilisant par exemple des statistiques bayésiennes ou des analyses de variabilité intra SOC.

TABLE 4 *Exemple d'alignement entre LOINC® et SNOMED CT ® obtenu avec la méthode DL*

Partie LOINC®		**Concept SNOMED CT**		**Similarité DL**	**Vrai Alignement ?**	**BestForBoth**	**Sous domaine**
LP14078-7	Babesia bovis	22405002	Babesia bovis	1	Oui	Oui	Organisme
LP14078-7	Babesia bovis	43574002	Babesia ovis	0,92	Non	Non	Organisme
LP30867-3	Leukocytes	38476002	Leukocyte	0,89	Oui	Oui	Cellule
LP7057-05	Bld	87612001	Blood	0,33	Oui	Oui	Milieu

3.2 Évaluation de la stratégie des ancres

L'évaluation des similarités syntaxiques a permis de démontrer deux principaux freins à leur utilisation pour l'alignement des SOC volumétriques et hétérogènes. On a pu observer

que ces métriques ne permettaient pas d'aligner des termes abrégés ou sous forme mnémonique avec leurs versions longues. Nous avons également montré que les métriques Stoilos et WGram ont des temps de calcul (CPU) plus long ce qui risque d'être problématique pour l'alignement ou le réalignement de SOC volumineux. Dans cette partie nous nous intéressons à deux nouvelles méthode ou filtre d'alignement pour de répondre à ces problématiques de termes abrégés et du temps de calcul total d'un alignement.

3.2.1 Évaluation des termes sémantiques candidats

Avant d'utiliser le calcul de similarité sémantique pour filtrer *a posteriori* les données obtenues par la méthode des ancres, nous avons vérifié la cohérence des termes sémantiques candidats extraits de MetaMap.

Nous observons que 24% des parties LOINC® (688) et 17% des concepts SNOMED CT® impliqués dans l'alignement n'ont aucun terme sémantique candidat. Nous remarquons également que les proportions de parties sans terme sémantique sont plus élevées dans les dimensions *Milieu* (60%), *Unité* (83%) et *Méthode* (40%) qui sont représentés par des termes abrégés ou des codes mnémoniques. Nous pouvons en conclure que l'utilisation des termes sémantiques candidats ne permet pas de résoudre les problématiques liées aux codes mnémoniques pour désigner un concept. Nous avons également observé que 45% des alignements de la collaboration impliquent un terme LOINC® ou SNOMED CT® représenter par aucun terme sémantique candidat. Ce résultat explique en partie le faible rappel obtenu par l'application du filtre sémantique sur les ancres initiales initiales (voir tableau 5). L'utilisation de l'UMLS® comme thésaurus ne permet pas de résoudre de manière systématique un alignement sémantique entre deux termes.

Cependant, on observe que plus de 50% des termes sont représentés par un unique terme sémantique candidat, ce qui nous conforte dans l'idée de que l'UMLS® est une ressource suffisamment précise pour calculer la similarité sémantique entre deux termes du DIV.

Pour confirmer la pertinence des termes sémantiques candidats, nous avons calculé les fréquences des types sémantiques associés aux termes sémantiques candidats pour chaque dimension de LOINC® et de chaque axe SNOMED CT® représentés. Nous observons que les types sémantiques les plus fréquemment retrouvés sont cohérents avec les sous-domaines représentés par les dimensions et les axes. Par exemple les concepts de l'axe *410607006|Organism* sont indexés par les types sémantiques représentant soit les procaryotes (*T007|Bacterium*), les eucaryotes (*T204|Eukaryote*) ou les virus (*T005|Virus*).

3.2.2 Performances de la stratégie des ancres et du filtre sémantique *a posteriori*

La méthode des ancres utilise un alignement existant pour déterminer les ancres initiales. Dans cette article nous avons utilisé le jeu d'alignement fournit par la collaboration, comme alignement initial que nous avons cherché à étendre avec la méthode des ancres.

Nous avons choisi de réaliser le calcul des nouvelles ancres par la méthode DL en appliquant un seuil de 0,8 pour filtrer les alignements générés. L'utilisation ici du filtre `BestForBoth` pendant la génération des ancres n'est pas judicieuse. En effet, `BestForBoth` est un filtre qui dépend notamment de l'ensemble des termes à aligner. Dans le cas où les parties et concepts n'ont qu'un parent par exemple, l'alignement entre leurs parents est le meilleur alignement possible et par conséquent il aurait été conservé par le filtre. Il a donc fallu retourner à des filtres basés sur des seuils afin de pouvoir contrôler la qualité et la quantité d'ancres générées. Le paramètre de seuil 0,8 permet d'obtenir une précision supérieure à 0,75 pour l'alignement réalisé à partir de alignements LOINC® SNOMED CT ® issus de la collaboration avec un nombre d'alignements cohérent par rapport au nombre de vrais alignements (< 1 200 alignements).

Les résultats bruts obtenus avec le paramètre 0,8 ont une précision très faible (0,33). L'application *a posteriori* des filtres de similarité sémantique ($\mathtt{sim}_{\mathtt{Semantique}}$) et `BestForBoth` permettent de doubler la précision de l'alignement sur les ancres générées. De plus, la

précision du filtre sémantique augmente de 10% si l'on tient compte de la confiance sémantique de l'alignement entre les termes et leurs termes sémantiques candidats. Les meilleures performances sont obtenues par combinaison des informations sémantiques et syntaxique. En effet, les filtres (i) $(sim_{Semantique} = 1 \wedge confiance_{Semantique} > 800) \vee sim_{DL} = 1$ et (ii) $sim_{Semantique} = 1 \wedge confiance_{Semantique} > 800 \wedge BestForBoth$ permettent d'obtenir 80% de précision pour 80% de rappel.

TABLE 5 *Résultats de l'évaluation de la méthode des ancres et de la méthode de filtrage basé sur la similarité sémantique. Les ancres initiales correspondent aux alignements proposés par le Regenstrief et l'IHTSDO. Pour les ancres générées la précision et le rappel sont calculés à partir de la curation manuelle des données brutes (1ère ligne). Les champs marqués avec un NA dans le tableau sont des valeurs non informatives. (1) Aucun rappel ne peut être calculé car le alignement de référence est obtenu par curation à partir de ces données. (2) L'alignement est un sous-ensemble du jeu de données initial (IHTSDO et Regenstrief Institute, 2013), la précision est donc de 1.*

Méthodes	**Nombre d'alignements**	**Paramètres de filtres**	**Précision (1)**	**Rappel (2)**
Ancres DL à 0,80	1 833 ancres générés	NA	0.33	NA
		BestForBoth	0.58	**0,97**
		$sim_{Semantique} \geq 0.5$	0,65	0,85
		$sim_{Semantique} = 1$	0,69	0,85
		$sim_{Semantique} = 1 \wedge confiance_{Semantique} > 800$	0,81	0,83
		$(sim_{Semantique} = 1 \wedge confiance_{Semantique} > 800) \vee sim_{DL} = 1$	**0,82**	0,87
		$sim_{Semantique} = 1 \wedge confiance_{Semantique} > 800 \wedge BestForBoth$	**0,82**	0,83
	2 177 initiales normalisés	$(sim_{Semantique} = 1 \wedge confiance_{Semantique} > 800)$	NA	**0,37**

4 Conclusion

Lors de cette étude nous avons cherché à identifier les méthodes les plus appropriées pour aligner des données du DIV. Nous avons montré que l'utilisation d'un filtre basé sur les meilleurs alignements par terme (BestForBoth) donne des résultats plus précis et complets qu'un seuil basé sur la similarité sans pour autant augmenter significativement le temps d'exécution des algorithmes. Nous avons également montré que la combinaison de métriques syntaxiques Stoilos et WGram permettait d'augmenter les performances. En nous intéressant au temps de calcul des métriques, nous avons observé que les métriques WGram et Stoilos ont des temps d'exécution 5 à 10 fois supérieure à la métrique DL ce qui peut être problématique pour l'alignement entre SOC volumétrique évoluant très fréquemment. Pour optimiser les temps de calcul de ces algorithmes nous envisageons de paralléliser les processus d'alignement.

Nous avons par la suite étudié les résultats issus d'un algorithme heuristique, qui couplé avec un filtre sémantique *a posteriori* permet d'améliorer la précision et le rappel. Pour compléter l'étude de la méthode des ancres, nous envisageons d'implémenter l'algorithme WGram pour générer des ancres, ainsi que d'intégrer les filtres de dimension sémantique pendant la recherche. Nous espérons ainsi obtenir plus de résultats d'alignement tout en conservant des performances acceptables (0,80 en précision et rappel).

La dimension sémantique est une méthode dépendant d'une ressource externe et est fonction des termes en entrée. Nous avons montré que la confiance sémantique variait en fonction de la présence dans le terme d'informations non spécifiques au concept, les étapes de normalisation des termes sont cruciales pour utiliser la similarité sémantique que nous avons mise en place. Nous avons également montré que le Metathesaurus® bien qu'intégrant près de 200 ressources ne permettait pas de garantir l'identification d'un concept UMLS pour

l'ensemble des parties LOINC®, ou concepts SNOMED CT®. Pour minimiser l'impact de l'incomplétude de cette ressource, nous préconisons donc l'utilisation des paramètres sémantiques en complément de métriques syntaxiques.

L'un des enjeux dans l'alignement entre LOINC® et SNOMED CT®, ou de manière plus générale entre les SOC concerne l'alignement d'abréviations. Les métriques syntaxiques sont par essence incapables de reconnaître les abréviations. Nous pensions que l'utilisation du Metathesaurus® améliorerait l'alignement entre termes abrégés. L'étude que nous avons réalisée n'a pas permis de confirmer cette hypothèse. La gestion des abréviations demeure donc une problématique que nous envisageons de résoudre grâce à la construction ou l'utilisation de dictionnaires d'abréviations.

En conclusion cette étude nous a permis d'identifier les forces et faiblesses de chaque algorithme et métrique d'alignement. L'ensemble de ces travaux nous a permis d'établir la stratégie d'alignement que nous allons réutiliser sur d'autre SOC représentant le domaine du diagnostic *in vitro.* Il nous semble important de souligner que quelque soit la méthode utilisée, il est fondamental d'impliquer des experts dans le processus d'alignement entre deux SOC. Les méthodes de filtres que nous proposons ne garantissent au mieux 80% de précision et 80% de rappel. Les similarités calculées doivent être considérées comme des indicateurs d'alignements plus que comme une vérité systématique. Ce constat nous pousse à élargir notre conception d'un processus d'alignement. En plus de l'aspect algorithmique, le processus doit également intégrer une part d'expertise humaine par une dimension visuelle et explicative de l'alignement. Ces points sont argumentés dans le domaine de recherche sur les alignements d'ontologies (Shvaiko & Euzenat, 2008).

Références

ARONSON, A. R. (1996). *Metamap technical notes*. Technical report, United States National Library of Medicine, Bethesda, MD.

ARONSON, A. R., & LANG, F.-M. (2010). An overview of MetaMap: historical perspective and recent advances. *Journal of the American Medical Informatics Association*, *17*(3), 229–236.

BLUMENTHAL, D. (2010). Launching HITECH. *New England Journal of Medicine*, *362*(5), 382‑385. doi:10.1056/NEJMp0912825

BRAHMA, B., & REFOUFI, A. (2015). *Ontology Matching Algorithms*. Communication présentée au Proceedings of the International Conference on Intelligent Information Processing, Security and Advanced Communication (p. 89:1–89:5), New York, NY, USA:ACM. doi:10.1145/2816839.2816928

CORNET, R., & DE KEIZER, N. (2008). Forty years of SNOMED: a literature review. *BMC Medical Informatics and Decision Making*, *8*(Suppl 1), S2. doi:10.1186/1472-6947-8-S1-S2

DAMERAU, F. J. (1964). A technique for computer detection and correction of spelling errors. *Communications of the ACM*, *7*(3), 171–176.

DOS REIS J. C., CÉDRIC PRUSKI C., REYNAUD-DELAÎTRE C. (2015). State-of-the-art on mapping maintenance and challenges towards a fully automatic approach. *Expert Syst. Appl.*42(3): 1465-1478

DOLIN, R. H., HUFF, S. M., ROCHA, R. A., SPACKMAN, K. A., & CAMPBELL, K. E. (1998). Evaluation of a « lexically assign, logically refine » strategy for semi-automated integration of overlapping terminologies. *Journal of the American Medical Informatics Association*, *5*(2), 203–213.

EUZENAT, J., & SHVAIKO, P. (2013). *Ontology matching*. Springer-Verlag.

FIESCHI, M. (2009). *La gouvernance de l'interopérabilité sémantique est au coeur du développement des systèmes d'information en santé* (rapport public).

FUNG, K. W., & BODENREIDER, O. (2005). Utilizing the UMLS for Semantic Mapping between Terminologies. *AMIA Annual Symposium Proceedings*, *2005*, 266‑270.

HODGE, G. (2000). *Systems of Knowledge Organization for Digital Libraries: Beyond Traditional Authority Files.* ERIC. Repéré à http://eric.ed.gov/?id=ED440657

IHTSDO ET REGENSTRIEF INSTITUTE. (juillet 2013). Regenstrief and the IHTSDO are working

together to link LOINC and SNOMED CT. Repéré à https://loinc.org/collaboration/ihtsdo

LEE, D., DE KEIZER, N., LAU, F., & CORNET, R. (2013). Literature review of SNOMED CT use. *Journal of the American Medical Informatics Association*, *21*(e1), e11‑e19. doi:10.1136/amiajnl-2013-001636

LEVENSHTEIN, V. I. (1966). *Binary codes capable of correcting deletions, insertions, and reversals*. Communication présentée au Soviet physics doklady (vol. 10, p. 707–710).

MACARY, F. (2007). IHDE, CDA et LOINC : des composants d'interopérabilité au service du partage des résultats de biologie médicale. *Spectra biologie*, *26*(158), 51‑57.

MCCRAY, A. T. (1989). *The UMLS Semantic Network.* Communication présentée au Proceedings/the... Annual Symposium on Computer Application [sic] in Medical Care. Symposium on Computer Applications in Medical Care (p. 503–507), American Medical Informatics Association.

PRATT, W., & YETISGEN-YILDIZ, M. (2003). *A study of biomedical concept identification: MetaMap vs. people*. Communication présentée au AMIA Annual Symposium Proceedings (vol. 2003, p. 529), American Medical Informatics Association.

R. CORE TEAM. (2014). *R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. 2013*. ISBN 3-900051-07-0.

SEDDIQUI, M. H., & AONO, M. (2009). An efficient and scalable algorithm for segmented alignment of ontologies of arbitrary size. *Web Semantics: Science, Services and Agents on the World Wide Web*, *7*(4), 344‑356. doi:10.1016/j.websem.2009.09.001

SHEIDE, A., & WILSON, P. S. (2013). Reading up on LOINC. *Journal of AHIMA/American Health Information Management Association*, *84*(4), 58–60.

SHVAIKO, P., & EUZENAT, J. (2005). A survey of schema-based matching approaches. Dans *Journal on Data Semantics IV* (p. 146–171). Springer.

SHVAIKO, P., & EUZENAT, J. (2008). Ten challenges for ontology matching. Dans *On the Move to Meaningful Internet Systems: OTM 2008* (p. 1164–1182). Springer.

STOILOS, G., STAMOU, G., & KOLLIAS, S. (2005). A String Metric for Ontology Alignment. Dans Y. Gil, E. Motta, V. R. Benjamins, & M. A. Musen (dir.), *The Semantic Web – ISWC 2005* (p. 624‑637). Springer Berlin Heidelberg.

STROETMANN, V. (2009). *Semantic Interoperability for Better Health and Safer Healthcare*. European Communities.

VAN DER LOO, M. P. (2014). The stringdist package for approximate string matching. *The R*.

VREEMAN, D. (7 novembre 2015). Guidelines for using LOINC and SNOMED CT Together. *Daniel Vreeman*. Repéré à https://danielvreeman.com

WINKLER, W. E. (1999). *The state of record linkage and current research problems*. Communication présentée au Statistical Research Division, US Census Bureau, Citeseer.

Evaluation de l'évolution de la complétude de DBpedia : une étude de cas

Fayçal Hamdi, Samira Si-said Cherfi, Subhi Issa

CEDRIC - CONSERVATOIRE NATIONAL DES ARTS ET MÉTIERS
292 rue saint martin, Paris, France

Résumé :
Les données du web, grâce à leur richesse sémantique et leur variété, sont de plus en plus utilisées par les chercheurs et les organisations. Cependant, la disponibilité de ces données repose sur l'effort collaboratif conduisant à un processus de publication des données peu contraint (des recommandations W3C et quelques contraintes technologiques). Ceci conduit à des descriptions de données éparses et hétérogènes avec un impact indéniable sur la qualité. Ceci explique l'intérêt croissant envers des approches d'amélioration de la qualité des données web. Dans cet article, nous nous sommes intéressés à la qualité des données Web et plus précisément de l'évolution de la complétude des sources de données au fil du temps. Nous présentons une série d'expériences visant à analyser l'évolution de la complétude des données sur plusieurs versions de DBpedia.
Mots-clés : Web de données, Qualité des données, Complétude, DBpedia

1 Introduction

De nos jours, les données jouent un rôle crucial dans les processus de prise de décision aussi bien pour les individus que pour les organisations. Cependant, alors que les bases de données traditionnelles contiennent des données homogènes et sont de taille raisonnable, les données du web ressemblent à un *patchwork* construit à partir de nombreuses et diverses sources. Ces données, bien que riches en contenu, sont souvent incomplètes et manquent de métadonnées permettant de les comprendre et de les analyser.

Par conséquent, lorsque les données sont utilisées pour faire des analyses ou tout simplement pour répondre aux requêtes des utilisateurs, il est important de tenir compte du biais que la qualité des données pourrait engendrer. Pour assurer cette qualité, une première solution consiste à impliquer l'utilisateur en le guidant dans le processus de publication des données (Heath *et al.*, 2008). La qualité qui est, dans ce cas, le degré de conformité avec les recommandations de publication est difficile à assurer et à évaluer. Une deuxième solution, plus objective, se concentre principalement sur l'évaluation et l'amélioration de la qualité. Ceci nécessite un important effort d'évaluation et d'amélioration de la qualité mais aussi celui du maintien de cette qualité. Une telle solution est coûteuse à mettre en oeuvre puisqu'elle implique un effort soutenu dans le temps face à des données qui ne cessent d'évoluer afin de préserver la qualité obtenue et assurer sa non régression.

Dans cet article, nous nous sommes intéressés à l'évolution de la complétude d'un jeu de données web. Nous pensons que la compréhension de cette évolution pourrait contribuer à définir des stratégies plus appropriées pour l'intégration, l'enrichissement et la maintenance des sources de données. Nous avons effectué une analyse exploratoire de la complétude des données de plusieurs catégories appartenant au jeu de données DBpedia (Bizer *et al.*, 2009). Nous avons défini d'abord une méthode d'évaluation de la complétude que nous avons ensuite appliquée à deux versions de DBpedia.

La suite de cet article est organisée comme suit : la section 2 résume les travaux connexes sur le sujet, tandis que la section 3 détaille la méthode de calcul de la complétude. La section 4 présente et analyse un ensemble d'expériences. Enfin, la section 5 tire des conclusions et donne de futures orientations.

2 Etat de l'art

La qualité de l'information a attiré de nombreux travaux de recherche pendant deux décennies. Des approches théoriques mais aussi pratiques ont conduit à la proposition de diverses méthodes centrées sur la qualité pour les systèmes d'information traditionnels (Batini *et al.*, 2009). Dans le contexte du web de données, la qualité est cependant une nouvelle problématique présentant de nombreux défis. Les chercheurs ont souligné le fait que rendre les données accessibles ne suffit pas, en particulier lorsque les utilisateurs sont des entreprises et des organismes gouvernementaux, et surtout quand l'usage concerne la sécurité des entreprises ou celle des pays. La crédibilité des sources de données repose en grande partie sur leur qualité.

Plusieurs propositions pourraient être analysées selon les 4 catégories proposées par l'approche TDQM [1] (Wang & Strong, 1996) à savoir : *la qualité Intrinsèque* (la précision, la réputation, la crédibilité et la provenance), la qualité *Représentative* (la compréhensibilité, la cohérence et la concision), l'*Accessibilité* (l'accessibilité et la sécurité), et la qualité *Contextuelle* (la quantité de données, la pertinence, la complétude et la rapidité). La qualité *Intrinsèque* s'appuie, lors de l'évaluation, sur les caractéristiques internes des données . Les travaux existants traitent surtout du problème de la provenance (Hartig, 2008). Du point de vue de la qualité externe, la qualité *Représentative* s'intéresse aux facteurs qui influencent les interprétations du point de vue des utilisateurs. On peut citer des travaux sur la compréhensibilité ou sur la concision de données (Zaveri *et al.*, 2013). En ce qui concerne l'accessibilité, un critère très important dans le contexte des données Web, les auteurs dans (Hogan *et al.*, 2010) ont analysé les erreurs communes lors du processus de publication et qui ont un impact sur l'accessibilité des données. Enfin, la qualité *Contextuelle* signifie que les données ne peuvent pas être considérées comme "bonnes" ou "mauvaises" sans tenir compte à la fois du contexte dans lequel elles ont été produites et dans celui dans lequel elles seront utilisées. La pertinence a été également évaluée en s'appuyant sur un processus de notation ou de classement des données (Eastman & Jansen, 2003; Herzig & Tran, 2012). Pour conclure, les approches existantes proposent peu de métriques objectives s'appuyant sur les caractéristiques internes des données. De plus, elles se concentrent sur l'évaluation et s'intéressent peu à l'évolution de la qualité une fois connue.

3 Calcul de la complétude : une approche basée sur la fouille de données

La mesure de la complétude, au niveau des données, s'appuie souvent sur les valeurs manquantes Pipino *et al.* (2002). Cette vision requiert, au vu des spécificités déjà citées du web de données, l'extraction du schéma à partir de la source de données elle-même. Cependant, il n'est pas pertinent de considérer, pour un sous-ensemble d'instances (appartenant à un jeu de données), le schéma comme l'union de toutes les propriétés utilisées dans leur description. En

1. *Total Data Quality Management* : http ://web.mit.edu/tdqm/

effet, cette vision néglige le fait que les valeurs manquantes pouvaient exprimer l'inapplicabilité. Cette dernière survient lorsque la propriété ne s'applique pas ou n'a pas de sens pour l'instance ou l'objet en question. Ceci nous permet de conclure que toutes les propriétés décrivant les instances d'une source ne sont pas d'égales importances.

Pour intégrer cet aspect, nous proposons une approche qui calcule la complétude en la formulant comme un problème d'exploration d'itemsets fréquents. Ainsi, seules les données qui appartiennent réellement à un jeu de données seront considérées dans l'inférence de son schéma. Notre approche comprend deux étapes :

1. **Extraction du schéma de la source :** Soit la source de données $\mathcal{D}$, nous représentons tout d'abord l'ensemble des propriétés qui décrivent les instances $\mathcal{D}$, comme un vecteur de transactions. Nous appliquons ensuite l'algorithme FP-growth (Han *et al.*, 2004) afin d'extraire les itemsets les plus fréquents (nous avons choisi l'algorithme FP-growth pour des raisons de performance des calculs. Un autre algorithme aurait tout à fait pu être utilisé). Nous ne retenons au final qu'un sous-ensemble, dit "Maximal", de ces ensembles les plus fréquents (Grahne & Zhu, 2003). Ce choix est motivé, d'une part, par le fait que nous accordons une importance à *l'expression* du pattern fréquent et, d'autre part, par le fait que le nombre de patterns les plus fréquents peut être exponentiel lorsque la taille du vecteur de transactions est importante (voir Section 3.1 pour plus de détails).
2. **Evaluation de la complétude :** Une fois que l'itemset le plus fréquent "Maximal" $\mathcal{MFP}$ aura été généré, nous exploitons la fréquence d'occurrence des items (propriétés) dans $\mathcal{MFP}$, pour assigner à chacun de ses items un poids qui reflète l'importance de la propriété sous-jacente dans la description des instances. Les poids ainsi définis serviront à l'évaluation de la complétude de chaque transaction (en tenant compte de la présence ou de l'absence des propriétés dans la transaction). Par la suite, la complétude de l'ensemble de la source de données est calculée.

Nous détaillons ci-après, chacune des étapes.

3.1 Extraction du schéma de la source

Soit une source de données $\mathcal{D}(C, I_C, P)$, où C est l'ensemble des catégories (ex. *Film*, *Organisation*), I_C est l'ensemble des instances des catégories C (ex. *Godfather* est une instance de la catégorie *Film*), et $P = \{p_1, p_2, ..., p_n\}$ est l'ensemble des propriétés telles que le nom ou le réalisateur (ex. $director(Film, Person)$).
Soit $\mathcal{T} = \{t_1^{i_1}, t_2^{i_2}, ..., t_m^{i_m}\}$ l'ensemble des transactions où $\forall k, 1 \leq k \leq m : t_k^{i_k} \subseteq P$, et $E(t_k^{i_k})$ est l'ensemble des items d'une transaction $t_k^{i_k}$. Chaque transaction est l'ensemble des propriétés utilisées dans la description des instances composant le sous-ensemble $\mathcal{I}' = \{i_1, i_2, ..., i_m\}$ avec $\mathcal{I}' \subseteq I_C$ (ex. les propriétés utilisées pour décrire l'instance *Godfather* sont : *name*, *director*, *producer*, *musicComposer*, .etc).
Initialement, $\mathcal{T} = \phi$, $\mathcal{MFP} = \phi$. Pour chaque $i \in \mathcal{I}'$ nous générons une transaction t. En effet, chaque instance i est liée à des valeurs (des ressources ou des littéraux) à travers un ensemble de propriétés. Par conséquent, une transaction t_k d'une instance i_k est un ensemble de propriétés tel que $t_k \subseteq P$. Les transactions générées ainsi pour toutes les instances de $\mathcal{I}'$ sont alors ajoutées à l'ensemble $\mathcal{T}$.

Exemple 1
Considérons l'exemple du tableau 1, soit $\mathcal{I}'$ un sous-ensemble d'instances tel que : $\mathcal{I}' =$ {The_Godfather, Goodfellas, True_Lies}*. L'ensemble des transactions $\mathcal{T}$ serait alors :*

$$\mathcal{T} = \{\{director, musicComposer\}, \{director, editing\}, \{director, editing, musicComposer\}\}$$

TABLE 1 – Echantillon de triplets DBpedia et leurs transactions

Sujet	Prédicat	Objet
The_Godfather	director	Francis_Ford_Coppola
The_Godfather	musicComposer	Nino_Rota
Goodfellas	director	Martin_Scorsese
Goodfellas	editing	Thelma_Schoonmaker
True_Lies	director	James_Cameron
True_Lies	editing	Conrad_Buff_IV
True_Lies	musicComposer	Brad_Fiedel

Ressource	Transaction
The_Godfather	{director, musicComposer}
Goodfellas	{director, editing}
True_Lies	{director, editing, musicComposer}

L'objectif est alors de calculer les ensembles fréquents de propriétés co-occurrentes, dits patterns fréquents $\mathcal{FP}$, à partir du vecteur de transaction $\mathcal{T}$.

Définition 1 (Pattern)
Soit $\mathcal{T}$ un ensemble de transactions. Un pattern $\hat{P}$ est une séquence de propriétés présentes dans une ou plusieurs transactions t de $\mathcal{T}$.

Pour chaque pattern $\hat{P}$, soit $E(\hat{P})$ l'ensemble des items qui le composent et qui correspondent dans notre cas aux propriétés, et $T(\hat{P}) = \{t \in \mathcal{T} \mid E(\hat{P}) \subseteq E(t)\}$ l'ensemble des transactions correspondant. $E(\hat{P})$ désigne l' *expression* de $\hat{P}$, et $|T(\hat{P})|$ son *support*. Un pattern $\hat{P}$ est fréquent si $\frac{1}{|\mathcal{T}|}|T(\hat{P})| \geq \xi$, où ξ désigne un seuil spécifié par l'utilisateur.

Exemple 2
Considérons le tableau 1, soit $\hat{P} = \{director, musicComposer\}$ et $\xi = 60\%$. $\hat{P}$ est fréquent puisque son support qui vaut ($66,7\%$) est plus élevé que ξ.

Pour trouver tous les patterns fréquents $\mathcal{FP}$, nous avons utilisé, comme expliqué précédemment, l'algorithme FP-growth pour l'extraction d'itemsets fréquents. Cependant, et à cause de la taille du vecteur de transactions, l'algorithme FP-growth risque de générer un très grand ensemble $\mathcal{FP}$. Mais comme notre but est de mesurer à quel point une transaction (ou la description d'une instance) est *complète* au regard d'un ensemble de propriétés, nous privilégions *l'expression* du pattern (donc les items qu'il contient) plutôt que son *support*.

Concernant le calcul de la complétude, nous devons choisir un seul pattern qui servira de schéma de référence à la source. Ce pattern devra préserver le juste équilibre entre la fréquence et l'expressivité. Dans l'extraction des itemsets, le concept d'itemset fréquent "Maximal" fournit un tel pattern. Par conséquent, pour réduire l'ensemble $\mathcal{FP}$, nous générons un sous-ensemble contenant uniquement les patterns maximaux.

Définition 2 ($\mathcal{MFP}$)
Soit $\hat{P}$ un pattern fréquent. $\hat{P}$ est dit maximal si aucun sur-ensemble pouvant être composé à partir de ses propriétés n'est fréquent. Nous définissons l'ensemble des Patterns Fréquents

Maximaux $\mathcal{MFP}$ comme suit :

$$\mathcal{MFP} = \{\hat{P} \in \mathcal{FP} \mid \forall \hat{P}' \supsetneq \hat{P} : \frac{|T(\hat{P}')|}{|\mathcal{T}|} < \xi\}$$

où ξ désigne un seuil spécifié par l'utilisateur

Exemple 3
*En considérant le tableau 1, soit $\xi = 60\%$ et l'ensemble des patterns fréquents $\mathcal{FP}$ = {{*director*}, {*musicComposer*},{*editing*}, {*director,musicComposer*} {*director,editing*}}. L'ensemble $\mathcal{MFP}$ est alors : $\mathcal{MFP}$={{*director,musicComposer*},{*director, editing*}}*

3.2 Calcul de la complétude

Dans cette étape, nous réalisons pour chaque transaction, une comparaison entre ses propriétés et chaque pattern de l'ensemble $\mathcal{MFP}$ (en vérifiant la présence ou l'absence d'un pattern) que nous considérons comme le schéma inféré. Une moyenne est donc calculée pour obtenir la complétude de chaque transaction $t \in \mathcal{T}$. Enfin, la complétude de l'ensemble de $\mathcal{T}$ sera la moyenne de toutes les valeurs de complétude calculées pour chaque transaction.

Définition 3
*(*Complétude*) Soit $\mathcal{I}'$ un sous-ensemble d'instances, $\mathcal{T}$ l'ensemble de transactions construites à partir de $\mathcal{I}'$, et $\mathcal{MFP}$ l'ensemble des Patterns Fréquents Maximaux. La complétude de $\mathcal{I}'$ correspond à la complétude de son vecteur de transactions $\mathcal{T}$. Cette complétude est calculée comme la moyenne des complétudes de chacune des transactions au regard de $\mathcal{MFP}$. Par conséquent, nous définissons la complétude $\mathcal{CP}$ d'un sous-ensemble d'instances $\mathcal{I}'$ comme suit :*

$$\mathcal{CP}(\mathcal{I}') = \frac{1}{|\mathcal{T}|} \sum_{k=1}^{|\mathcal{T}|} \sum_{j=1}^{|\mathcal{MFP}|} \frac{\delta(E(t_k), \hat{P}_j)}{|\mathcal{MFP}|} \tag{1}$$

$$\textit{tel que : } \hat{P}_j \in \mathcal{MFP}, \textit{ et } \delta(E(t_k), \hat{P}_j) = \begin{cases} 1 & \textit{si } \hat{P}_j \subset E(t_k) \\ 0 & \textit{sinon} \end{cases}$$

Exemple 4
*Admettons que ξ soit fixé à 60%. La complétude du sous-ensemble d'instances du tableau 1 au regard de $\mathcal{MFP}$ = {{*director, musicComposer*}, {*director, editing*}}, est calculé par :*

$$\mathcal{CP}(\mathcal{I}') = (2 * (1/2) + (2/2))/3 = 0.67$$

Cette valeur correspond à la complétude moyenne de l'ensemble de données au regard du schéma inféré $\mathcal{MFP}$. Si l'on calculait la complétude par rapport à un schéma tel que {{*editing, musicComposer*}}, qui ne fait partie du $\mathcal{MFP}$, la valeur de complétude par rapport à ce schéma est de 0.33, qui est très en dessous du seuil de 60% exigé par l'utilisateur.

TABLE 2 – Nombre d'instances/catégorie

	Film	Organisation	PopulatedPlace	Scientist
v3.6(2013)	53 619	147 889	340 443	9 726
v2015-04	90 060	187 731	455 398	20 301

4 Expérimentations

Les expérimentations ont été réalisées sur DBpedia qui est une base de connaissances issue d'un effort communautaire dont les données sont dérivées de Wikipedia. Elle contient actuellement 4,58 millions d'entités.

Pour évaluer la complétude des différentes versions de DBpedia, nous avons choisi deux versions relativement éloignées. La première (v3.6) a été générée en Mars/Avril 2013 et la seconde (v2015-04) en Février/Mars 2015. Pour chaque ensemble de données, nous avons choisi des catégories de différentes natures. Nous avons étudié la complétude des ressources (instances) qui ont pour types : $C = \{$*Film*, *Organisation*, *Scientist*, *PopulatedPlace*$\}$.
Pour les propriétés utilisées dans les descriptions des ressources, nous avons choisi les ensembles de données DBpedia "mapping-based properties", "instance types" et "labels" (les versions anglaises). La taille de chaque catégorie (le nombre d'instances de même type) est donnée dans le tableau 2.

Dans la première étape des expérimentations, nous avons construit un vecteur de transactions $\mathcal{T}$ qui est constitué de séquences de propriétés déduites des instances appartenant à une seule catégorie (par exemple l'ensemble des *Film* dans DBpedia). Ce vecteur est ensuite utilisé comme entrée pour extraire les patterns fréquents et pour calculer la complétude. Les expérimentations ont été exécutées sur un Dell XPS 27 avec un processeur Intel Core i7-4770S et 16GB de DDR3. Le temps d'exécution de chaque expérimentation est autour de 2 minutes.

La méthode d'évaluation consiste à calculer au regard du même schéma inféré (c.-à-d. le même $\mathcal{MFP}$), la complétude des différentes versions. Nous avons choisi, comme schéma de référence pour nos expériences, celui déduit de l'ancienne version (2013). Ainsi, nous pourrons observer l'évolution des nouvelles versions.

La figure 1 montre les résultats de complétude obtenus pour les catégories choisies de DBpedia v3.6 et v2015-04 en variant le support ξ. La complétude est calculée pour les deux versions au regard du même $\mathcal{MFP}$ déduit de la version v3.6.

Nous observons pour les catégories *Scientist* et *Organisation* (à l'exception de la complétude pour ξ compris entre 30% et 50%), que les valeurs de complétude sont presque les mêmes. Toutefois, pour *Films* et surtout pour *PopulatedPlace* il y a une nette différence dans les valeurs de complétude. Pour la catégorie *Films*, les valeurs de complétude de la version v2015-04 sont inférieures à celles de la version v3.6. Cela signifie que les descriptions des ressources (instances) dans la nouvelle version sont moins complètes que celles de l'ancienne version. Cela pourrait être dû à deux raisons ; les nouvelles ressources ajoutées à DBpedia v2015-04 ne disposent pas dans leurs descriptions de toutes les propriétés du schéma de référence, ou certaines propriétés (du schéma de référence) appartenant à des ressources existantes ont été enlevées suite à leurs mises à jour.

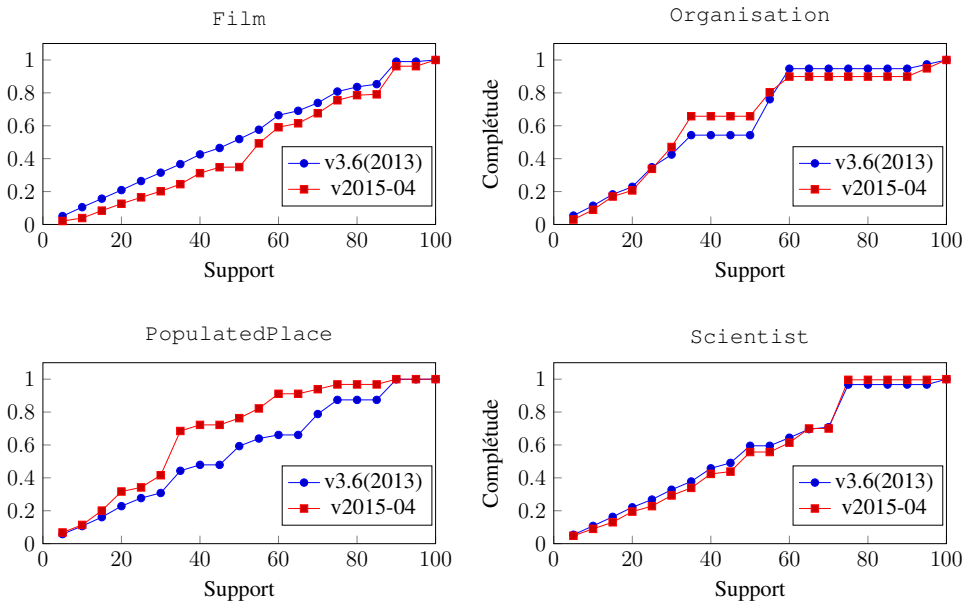

FIGURE 1 – La complétude de DBpedia v3.6 et v2015-04

En effet, nous savons que, outre le fait que de nouvelles ressources sont ajoutées à chaque nouvelle version de DBpedia (cf. tableau 2), les ressources existantes pourraient également être mises à jour. Nous prenons comme exemple le film "Driving et Miss Daisy", qui a dans sa description dans la version v3.6, des propriétés telles que *type*, *label*, *name*, *director*, *producer*, etc., et dans la version v2015-04 seulement *type*, *label* et *homepage*.

En ce qui concerne la catégorie *PopulatedPlace*, les résultats montrent que, contrairement à *Films*, les valeurs de complétude sont plus élevées. Là aussi, les raisons sont les mêmes, mais dans ce cas, la complétude des ressources existantes a été améliorée et/ou les nouvelles ressources sont plus complètes au regard du schéma de référence. Par exemple, la ressource "Haryana" a comme propriétés dans sa description dans la version v3.6 seulement *type* et *label*. Cependant, dans la version v2015-04 il y a vingt propriétés supplémentaires (par exemple *country*, *areaTotal*, *populationTotal*, *leaderName*, etc.) dans sa description.

Pour mieux interpréter ces résultats, nous avons effectué une deuxième série de tests en calculant cette fois-ci la complétude uniquement sur des ressources équivalentes appartenant aux deux versions. Ceci permet de vérifier si la variation de la complétude est induite par les modifications apportées aux ressources existantes, ou par les nouvelles ressources ajoutées. Les résultats de cette nouvelle expérience sont illustrés dans la figure 2.

La figure 2 montre que les courbes des catégories *Films*, *Organisation*, et *PopulatedPlace* sont à peu près les mêmes pour les deux versions. Ainsi, nous pouvons conclure que pour ces catégories, la variation de la complétude est principalement due aux modifications apportées aux ressources existantes. Pour la catégorie *Scientist*, les résultats montrent que, même si les descriptions des ressources existantes ont été améliorées, les nouvelles ressources (celles du v2015-04) détériorent les valeurs de complétude de la version v2015-04. Nous pouvons donc conclure que, dans ce cas, la variation est principalement due aux nouvelles ressources.

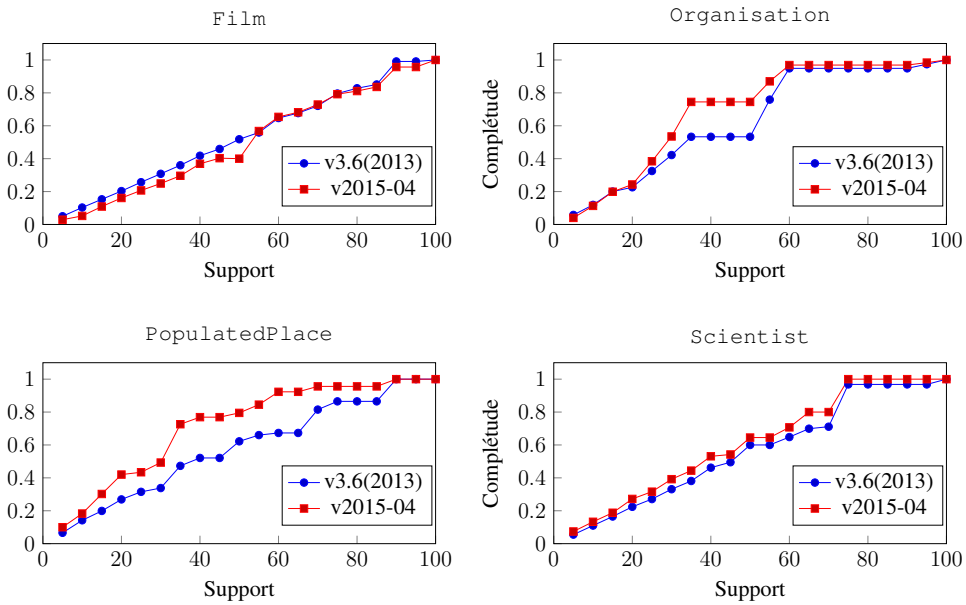

FIGURE 2 – La complétude des ressources équivalentes dans DBpedia v3.6 et v2015-04

5 Conclusion

Cet article est une étude exploratoire sur l'évolution de la complétude de DBpedia. Nous avons d'abord présenté une approche de calcul de la complétude. Nous avons ensuite mené une série d'expérimentations sur deux versions relativement éloignées de DBpedia. Ces expérimentations ont révélé que la variation dans le temps, de la complétude d'un jeu de données pourrait être la conséquence de modifications apportées aux données existantes ou d'ajouts de nouvelles données. Nous avons également remarqué que cette évolution ne tire souvent pas profit du nettoyage initial des données, car les propriétés décrivant les ressources continuent à évoluer au fil du temps.
Notre approche pourrait être utile pour les fournisseurs de sources de données pour améliorer, ou au moins, pour préserver une certaine complétude de leurs sources de données. Elle pourrait être particulièrement utile pour les ensembles de données construits de manière collaborative, en imposant aux contributeurs quelques règles quand ils veulent ajouter de nouvelles ressources ou mettre à jour les ressources existantes.

Comme perspectives de travail, nous prévoyons d'enrichir notre enquête en étudiant d'autres sources de données telles que Yago, IMDB, etc. Nous travaillons actuellement sur une approche d'amélioration de la complétude guidée par le contenu des jeux de données, par les valeurs de complétude, et par les liens *owl :sameAs*.

Références

BATINI C., CAPPIELLO C., FRANCALANCI C. & MAURINO A. (2009). Methodologies for data quality assessment and improvement. *ACM Computing Surveys (CSUR)*, **41**(3), 16.

BIZER C., LEHMANN J., KOBILAROV G., AUER S., BECKER C., CYGANIAK R. & HELLMANN S. (2009). Dbpedia-a crystallization point for the web of data. *Web Semantics : science, services and agents on the world wide web*, **7**(3), 154–165.

EASTMAN C. M. & JANSEN B. J. (2003). Coverage, relevance, and ranking : The impact of query operators on web search engine results. *ACM Transactions on Information Systems (TOIS)*, **21**(4), 383–411.

GRAHNE G. & ZHU J. (2003). Efficiently using prefix-trees in mining frequent itemsets. In *Proceedings of the ICDM 2003 Workshop on Frequent Itemset Mining Implementations, 19 December 2003, Melbourne, Florida, USA*.

HAN J., PEI J., YIN Y. & MAO R. (2004). Mining frequent patterns without candidate generation : A frequent-pattern tree approach. *Data Min. Knowl. Discov.*, **8**(1), 53–87.

HARTIG O. (2008). Trustworthiness of data on the web. In *Proceedings of the STI Berlin & CSW PhD Workshop* : Citeseer.

HEATH T., HAUSENBLAS M., BIZER C., CYGANIAK R. & HARTIG O. (2008). How to publish linked data on the web. In *Tutorial in the 7th International Semantic Web Conference, Karlsruhe, Germany*.

HERZIG D. M. & TRAN T. (2012). Heterogeneous web data search using relevance-based on the fly data integration. In *Proceedings of the 21st World Wide Web Conference 2012, WWW 2012, Lyon, France, April 16-20, 2012*, p. 141–150 : ACM.

HOGAN A., HARTH A., PASSANT A., DECKER S. & POLLERES A. (2010). Weaving the pedantic web. In *Proceedings of the WWW2010 Workshop on Linked Data on the Web, LDOW 2010, Raleigh, USA, April 27, 2010*.

PIPINO L. L., LEE Y. W. & WANG R. Y. (2002). Data quality assessment. *Communications of the ACM*, **45**(4), 211–218.

WANG R. Y. & STRONG D. M. (1996). Beyond accuracy : What data quality means to data consumers. *Journal of management information systems*, p. 5–33.

ZAVERI A., RULA A., MAURINO A., PIETROBON R., LEHMANN J., AUER S. & HITZLER P. (2013). Quality assessment methodologies for linked open data. *Submitted to Semantic Web Journal*.

Détection et Représentation des changements dans les sources de données RDF

Daniel Mercier[1], Nathalie Pernelle[1], Fatiha Saïs[1], Sujeeban Thuraisamy[1]

UNIVERSITÉ PARIS SUD, LABORATOIRE DE RECHERCHE EN INFORMATIQUE
91405 Orsay cedex, France
{pernelle, sais}@lri.fr, {DanielMercier,Sujeeban.Thuraisamy}@u-psud.fr

Résumé : De nombreuses sources de données RDF sont en évolution constante que ce soit au niveau des données ou du vocabulaire utilisé (ontologie). De nombreuses tâches d'intégration sont impactées par ces modifications qu'il s'agisse de synchroniser des données locales avec une source de données externe ou d'effectuer des traitements plus complexes comme le liage ou la fusion de données. Dans ce contexte, il est important de disposer d'outils permettant de détecter et de représenter ces changements de façon à ce que les tâches d'intégration impactées puissent mettre-à-jour leurs résultats sans devoir redémarrer un processus de zéro. De nombreux travaux se sont focalisés sur la détection, la représentation et le management des changements au niveau ontologique. Dans ce papier, nous présentons une approche permettant de détecter et de représenter des changements plus ou moins complexes que l'on peut détecter lorsque l'on s'intéresse aux seules données. Une première expérimentation a été menée sur différentes versions de DBPedia.

Mots-clés : Ontologies, Représentation des connaissances, Evolution des données et des connaissances

1 Introduction

Une des caractéristiques intrinsèque du Web des données (LOD) est la dynamicité et l'évolution permanente de son contenu. En effet, de nombreuses modifications sont apportées quotidiennement sur les données et les vocabulaires publiés sur le LOD. En 2012 (Käfer *et al.* (2012)), 76% des documents RDF du LOD ont subi au moins une modification. Ces modifications peuvent être de différents types (ajout, suppression et renommage) et peuvent intervenir à différents niveaux : au niveau ontologique (classes, propriétés, axiomes et liens avec les autres ontologies) et au niveau données (les entités, les propriétés, les valeurs des propriétés et les liens entre entités). De nombreuses tâches d'intégration de données sont impactées par ces modifications qu'il s'agisse de synchroniser des données locales avec une source de données externe ou d'effectuer des traitements plus complexes comme le liage ou la fusion de données.

Dans ce contexte, il est important de disposer d'outils permettant de détecter et de représenter ces changements de façon à ce que les tâches d'intégration impactées puissent mettre à jour leurs résultats sans devoir ré-appliquer tout le processus sur toutes les données. Par ailleurs, compte tenu du volume important des données, les tâches d'intégration de données doivent être complètement automatiques. Par conséquent, pour avoir des résultats sûrs il est parfois important de faire appel à des experts humains pour valider les résultats. Pour éviter de solliciter de tels experts chaque fois que les données subissent des modifications, il faut pouvoir identifier les résultats qui restent valides malgré les mises-à-jour. De nombreux travaux se sont focalisés sur la détection, la représentation et la gestion des changements au niveau ontologique (voir F. Zablith et al (Zablith *et al.* (2015)) pour une vue d'ensemble). Dans ce type de travaux, le problème consiste à détecter et à représenter les changements au niveau conceptuel entre différentes

versions d'une même ontologie et de s'intéresser à la cohérence de cette dernière. Certains travaux (Dinh *et al.* (2014)) se sont focalisés sur la représentation des évolutions subies par une ontologie et sur l'adaptation d'un ensemble de correspondances (mappings) entre ontologies associées à des bases de connaissances. Quelques travaux récents (Papavasileiou *et al.* (2013)) se sont intéressés au problème de détection de changement dans les données et de leur représentation par des types plus abstraits. Cependant, ces derniers ne sont pas représentés dans une ontologie.

Dans cet article, nous présentons une approche permettant de détecter et de représenter des changements plus ou moins complexes que l'on peut identifier lorsque l'on s'intéresse aux seules données. Une ontologie nommée O^{DE} a été conçue pour représenter les changements subis par les données de façon plus expressive qu'une simple liste d'ajouts et de suppressions. Le but de cette ontologie est qu'elle puisse être facilement exploitée par des tâches d'intégration de données telles que le liage de données, la découverte automatique de clés et la fusion de données. Une approche permettant de peupler cette ontologie automatiquement à partir de deux versions d'une source de données RDF a été développée.

Une première expérimentation a été menée sur différentes versions de DBPedia, une ancienne et une récente et deux versions consécutives de DBPedia.

Dans la section 2, nous présentons l'approche de détection et de représentation de l'évolution des sources de données RDF. Nous présentons ensuite en section 3 quelques exemples de requêtes permettant d'exploiter l'ontologie O^{DE}. Nous présentons en section 4 les premiers résultats d'expérimentation sur les données de DBPedia. Enfin, nous concluons et dressons quelques perspectives en section 5.

2 Approche de détection et de représentation de l'évolution de sources de données RDF

Dans cette section nous présentons notre approche de détection et de représentation de changements dans deux versions d'une source de données.

Comme nous le montrons en Figure 1, étant données deux versions old et new d'une source de données, l'outil charge le contenu de chacune des versions dans un TDB –Jena Triples Database (Owens *et al.* (2008)). Ainsi, nous obtenons deux bases de données TDB_{old} et TDB_{new} correspondant respectivement au contenu de la version old et de la version new de la source considérée. Ensuite, une méthode qui permet de détecter les changements élémentaires entre deux versions d'une source de données RDF est appliquée. Plus précisément, cette méthode prend en entrée TDB_{old} et TDB_{new} et construit deux ensembles de triplets : l'ensemble des triplets de TDB_{new} qui n'existaient pas dans TDB_{old} (triplets ajoutés) et l'ensemble des triplets de TDB_{old} qui n'existent plus dans TDB_{new} (triplets supprimés).
A partir de ces ensembles de triplets ajoutés et supprimés, une étape de représentation sémantique des changements est réalisée. En effet, grâce à l'ontologie O^{DE} (voir la sous-section 2.1) que nous avons conçue, il est possible de représenter sémantiquement les types de changements survenus entre deux versions différentes d'une source de données. Au lieu de représenter les changements uniquement par deux ensembles d'assertions ajoutées et supprimées, nous avons défini des types de changements, et chaque type est représenté par une classe de l'ontologie O^{DE} (ajout ou suppression d'une propriété, enrichissement/apauvrissement de la description des instances, ...).

Dans cette étape de représentation sémantique des changements, il s'agit, d'une part,

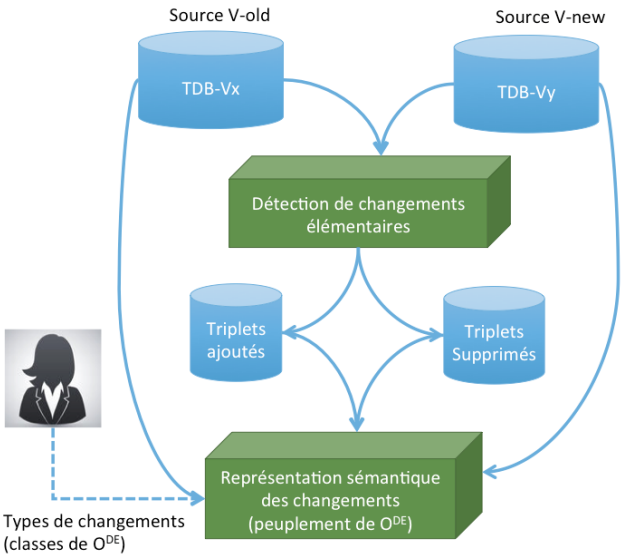

FIGURE 1 – Schéma global de l'approche de détection et de représentation de l'évolution de sources RDF

d'analyser les triplets pour détecter un ensemble de changements mais également garder trace des triplets qui sont à l'origine du type de changement détecté. Aussi, chaque triplet supprimé ou ajouté est associé à un ou plusieurs types de changements définis dans cette ontologie. Plus précisément, dans cette dernière étape de représentation de changements, étant données les deux ensembles d'assertions ajoutées et supprimées et les deux versions old et new de la source de données, l'outil détecte les types sémantiques (classes de l'ontologie O^{DE}) à associer à chaque assertion ajoutée ou supprimée. A l'issue de cette étape, on obtient une ontologie O^{DE} peuplée avec les triplets réifiés avec les types de changements qu'ils représentent. On note que l'on peut associer plusieurs types (i.e. classes de l'ontologie O^{DE}) à un même triplet.

L'ontologie peuplée peut ensuite être exploitée pour répondre à des requêtes d'experts souhaitant obtenir des informations sur l'évolution d'une source de données. Il est également possible de répondre à des requêtes plus complexes, comme par exemple : la liste des nouvelles propriétés et les objets associés, la liste des changements concernant une URI donnée et la liste des propriétés fonctionnelles ayant subit des changements de valeurs.

Nous présentons en sous-section 2.1, l'ontologie O^{DE} puis la méthode de peuplement de cette dernière en sous-section 2.2.

2.1 Ontologie d'évolution (O^{DE})

O^{DE} est une ontologie qui permet de représenter sémantiquement es évolutions qui peuvent surgir entre deux versions d'une source de données. En effet, moyennant une classification plus fine des types de changements au niveau des données, il est possible de représenter avec plus de précision les changements au niveau des données mais également de déduire des changements possibles au niveau du schéma (ontologie). Par ailleurs, avec cette représentation plus riche, il est possible pour un expert de domaine de poser des requêtes simples (e.g. le nombre

d'instances supprimées) ou plus complexes (e.g. les propriétés fonctionnelles dont la valeur a été modifiée). Enfin, à travers l'ontologie O^{DE}, il est possible de fournir aux outils dédiés aux tâches d'intégration de données l'ensemble des triplets impliqués dans des changements qui sont susceptibles d'impacter leur résultats. Cela permettra à ces outils de recalculer leurs résultats uniquement sur la partie des données qui a été mise-à-jour.

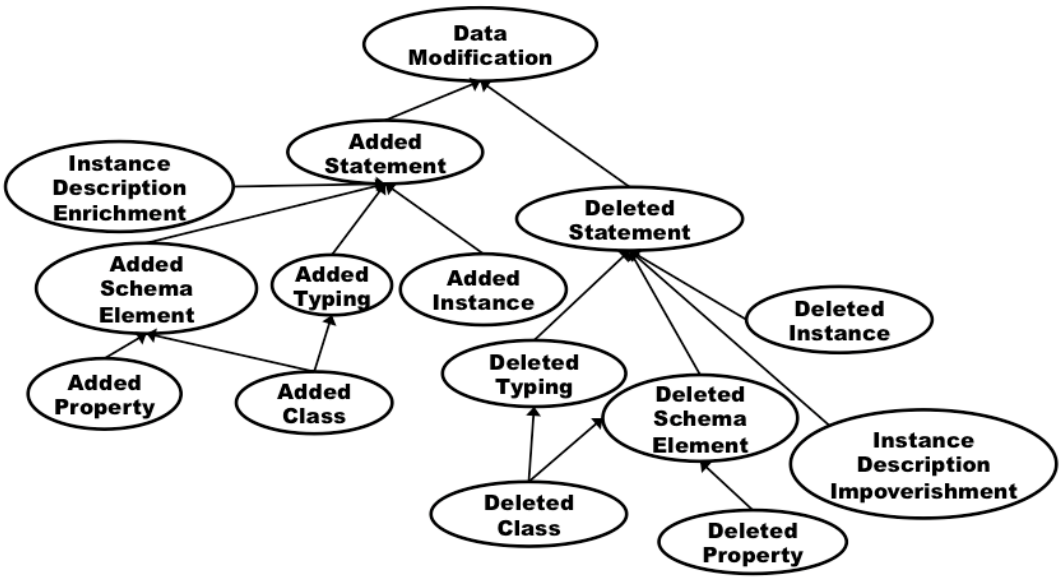

FIGURE 2 – Ontologie de changements dans les données RDF O^{DE}

Dans l'ontologie O^{DE} montrée en Figure 2, deux types généraux de changements sont distingués : *AddedStatement* et *DeletedStatement*. Pour chacun de ces types quatre sous-types sont associés. Pour le type *AddedStatement* on décrit :

- *AddedSchemaElement* qui représente les assertions impliqués dans l'introduction d'une nouvelle propriété (voir sous-type *AddedProperty*) ou d'une nouvelle classe (voir sous-type *AddedClass*) dans la nouvelle version de la source.
- *AddedTyping* qui décrit les assertions concernant l'introduction de nouveaux types pour des instances existantes (i.e. tous les nouveaux rdf :type). Parmi, ces assertions, certaines concernent la première utilisation d'une classe dans la source de données, d'où le lien de généralisation entre *AddedClass* et *AddedTyping* dans l'ontologie.
- *AddedInstance* qui représente l'ensemble des assertions décrivant des instances nouvelles dans la source de données.
- *InstanceDescriptionEnrichment* qui décrit l'ensemble des assertions qui viennent enrichir des instances existantes (assertions de la forme $<s\ p\ o>$ avec s existant dans l'ancienne source ou o existant dans l'ancienne source).

On note ici qu'une propriété *hasAddedInstance* de type *owl :ObjectProperty* est ajoutée à la classe *AddedInstance* pour garder trace des instances ayant été nouvellement introduites dans

la version new courante de la source (la classe étant représentée par un ensemble de triplets, il faut garder trace de l'URI de l'instance nouvelle sachant qu'il peut s'agir du sujet ou de l'objet des triplets). Par ailleurs, nous précisons que nous ne détaillerons pas la partie *DeletedStatement* de l'ontologie puisque celle-ci peut être décrite de façon strictement symétrique à celle de *AddedStatement*.

Les instances de l'ontologie O^{DE} sont des triplets réifiés par l'ajout d'un identifiant aux triplets. Ainsi, pour tout triplet <s, p, o> apparaissant dans l'ensemble de triplets ajoutés ou dans l'ensemble de triplets supprimés, on obtient la représentation réifiée suivante :

```
<tripleID-1 rdf:type rdf:Statement > .
<tripleID-1 rdf:subject s> .
<tripleID-1 rdf:predicate p> .
<tripleID-1 rdf:object o > .
<tripleID-1 rdf:type ClasseODE> .
```

2.2 Méthode de peuplement de O^{DE}

Après avoir détecté les changements élémentaires, en terme d'ajouts et de suppression d'assertions, entre deux versions old et new d'une source de données S, nous nous intéressons à la représentation sémantique des changements. Pour ce faire, nous avons développé deux algorithmes PODEA et PODES qui permettent de peupler l'ontologie O^{DE} en exploitant à la fois le fichier d'ajouts (resp. de suppressions) et le contenu de la version new (resp. old) de la source S.

Dans la suite nous ne présentons que l'algorithme PODEA (voir algorithme 1), puisque la description de l'algorithme PODES est analogue à celle de PODEA.

Dans l'algorithme PODEA, la fonction $existeA(s, p, o)$ vérifie l'existence du triplet (s, p, o) dans la version TDB_{old} de la source S. Une fonction $existeS(s, p, o)$ analogue est utilisée dans l'algorithme PODES pour vérifier l'existence du triplet (s, p, o) dans la version TDB_{new} de la source S

La fonction $instancier(c, tr)$ permet d'instancier la classe c de O^{DE} par le triplet réifié tr.

3 Interrogation de l'ontologie d'évolution (O^{DE})

L'ontologie O^{DE} peuplée par les triplets réifiés impliqués dans les changements peut être interrogée par un expert de domaine par des requêtes simples ou complexes représentées en SPARQL. Les requêtes simples que l'on peut exécuter permettent d'obtenir les instances O^{DE}. Ainsi, un expert de domaine pourra analyser les triplets qui induisent certains types de changements au niveau conceptuel ou au niveau des données. Par exemple, il pourra être intéressé par l'ensemble des classes qui ne sont plus instanciées dans la source de données ou par la liste complète des instances dont les triplets supprimés ont conduit à ce qu'une classe ne soit plus instanciée (voir table 1) .

L'expert pourra également poser des requêtes plus complexes combinant différents types d'informations : différentes classes de l'ontologie O^{DE}, des connaissances du domaine (e.g. fonctionnalité des propriétés) et des triplets des deux versions de la source de données considérée. La requête présentée en table 2 permet par exemple de retrouver la liste des instances de la propriété fonctionnelle $adresse Principale$ dont la valeur a été modifiée (i.e. le triplet

Algorithme 1: PODEA – Peuplement de O^{DE} avec des changements de type ajouts –

Input :
– T_a : l'ensemble de triplets RDF ajoutés
– TDB_{old} : la version old de la source de données
– O^{DE} : l'ontologie d'évolution de données
Output : O^{DE}, l'ontologie d'évolution peuplée avec les changements T_a survenus dans TDB_{old}

```
for each (triplet t(subject, predicate, object) ∈ Ta) do
    if (predicate == rdf :type) then
        if (¬ existA( ?s, predicate, object)) then
            instantiate(AddedClass, reification(t))
        else
            instantiate(AddedTyping, reification(t))

    else
        if (¬ isLiteral(object) and (¬ (existA( ?s, ?p, object) or existA(object, ?p, ?o)))) then
            instantiate(AddedInstance, reification(t))
            instantiateHasAddedInstance(t, object)
        else if (¬ isLiteral(object)) then
            instantiate(InstanceDescriptionEnrichment, reification(t))
    if (¬ (existA( ?s, ?p, subject) or existA(subject, ?p, ?o))) then
        instantiate(AddedInstance, reification(t))
        instantiateHasAddedInstance(t, subject)
    else
        instantiate(InstanceDescriptionEnrichment, reification(t))
    if (¬ existA( ?s, predicate, ?o)) then
        instantiate(AddedPoperty, reification(t))
```

TABLE 1 – Requêtes simples exploitant la classe DeletedClass

/* Liste des classes supprimées

```
SELECT DISTINCT ?deletedClass
WHERE {
?node rdf:type   ode:DeletedClass
?node rdf:object ?deletedClass .
}
```

/* Liste des instances ayant conduit à une suppression de classe

```
SELECT ?s
WHERE {
?node rdf:type ode:DeletedClass .
?node rdf:subject ?s .
}
```

ayant même sujet pour le prédicat $adressePrincipale$ dans $ode : AddedStatement$ et dans $ode : DeletedStatement$).

Des requêtes SPARQL peuvent aussi être définies pour construire des fichiers de données représentant les différents types de changements pouvant avoir un impact sur les résultats d'un outil réalisant une tâche liées à l'intégration de données. Par exemple, pour le liage de données on pourrait définir une série de requêtes SPARQL dont les résultats permettraient de mettre à jour le résultat du liage en considérant la partie des données mises-à-jour susceptible d'impacter

TABLE 2 – Requête listant les modifications des valeurs d'une propriété fonctionnelle

```
SELECT ?subject  ?valueBefore ?valueAfter
WHERE {
?node      rdf:type      ode:AddedStatement .
?node      rdf:subject   ?subject .
?node      rdf:predicate <http://.../adressePrincipale> .
?node      rdf:object    ?valueAfter .
?othernode rdf:type      ode:DeletedStatement .
?othernode rdf:subject   ?subject .
?othernode rdf:predicate <http://.../adressePrincipale> .
?othernode rdf:object    ?valueBefore .
}
```

les résultats. La plupart des approche de liage (Saïs *et al.* (2009); Volz *et al.* (2009); Nikolov *et al.* (2012)) de données s'appuient sur des ensemble de propriétés discriminantes (Symeonidou *et al.* (2014)) pour détecter des liens d'identité entre données. Les approches peuvent être très couteuses. Ainsi, en utilisant cette approche il est possible de fournir les nouvelles instances et les modifications qu'ont subit les propriétés discriminantes pour les instances existantes.

4 Experimentations

L'objectif de ces expérimentations est de montrer que l'approche proposée permet de peupler l'ontologie des modifications O_{DE} lorsque qu'une source de données subit un nombre important de changements. Il s'agit également de montrer comment l'ontologie, une fois peuplée, peut être utilisée pour étudier les différents types de changements.

4.1 Description des données

Nous avons évalué notre approche en utilisant les versions 3.5, 3.8 et 3.9 de DBPedia [1]. Nous nous sommes intéressés aux données décrites dans la classe $Person$ (fichier PersonData) auxquelles ont été ajoutées toutes les données de typage concernant ces personnes. Le tableau 3 présente le nombre de triplets, le nombre d'instances de personnes, le nombre de propriétés ainsi que le nombre de types différents associés à ces personnes dans ces trois différentes versions.

TABLE 3 – Evolution des triplets décrivant les Personnes dans trois versions de DBPedia

	Version 3.5	Version 3.8	Version 3.9
#triplets	482 080	18 719 429	22 008 122
#instances	48 692	2 853 529	3 733 629
#propriétés	9	9	9
#types	71	348	434

1. http ://dbpedia.org/services-resources/datasets

Entre la version 3.5 et la version 3.9, le nombre de triplets de la classe $Person$ a été multiplié par 45, le nombre de classes typant ces instances a été multiplié par 6 mais le nombre de propriétés n'a pas été modifié. Le nombre de changements étant important, cela nous a semblé un bon exemple pour tester notre approche.

4.2 Résultats et discussion

Nous avons détecté les changements élémentaires entre deux versions successives de la classe $Person$ et obtenu les deux fichiers contenant pour l'un, les triplets ajoutés et pour l'autre les triplets supprimés (temps d'exécution inférieur à 10 mn). Ces deux fichiers ont été exploités pour peupler l'ontologie O_{DE} (temps d'exécution sur plus de 18 millions de triplets : 55 mn). Lorsque les versions v3.5 et v3.8 sont comparées, après peuplement de l'ontologie O_{DE}, plus de 18 millions d'assertions sont instances de la classe $DataModification$. Il faut bien sûr noter que leur réification entraîne un sur-coût en terme de représentation : ces 18 millions d'assertions sont représentées par plus de 155 millions de triplets.
La taille des données représentant les personnes ayant été multipliée par 39, presque toutes ces assertions sont de type $AddedStatement$ (tab. 4), mais près de la moitié des triplets existants ont été supprimés (classe $DeletedStatement$, tab.5). La table 4 (resp. 5) montre comment les assertions instances de la classe $AddedStatement$ (resp. $deletedstatement$) se répartissent dans les différentes classes de l'ontologie O_{DE}. Le temps d'exécution est également donné pour chaque type de changement.

TABLE 4 – Type et nombre de changements pour la classe $AddedStatement$

	#Added Statements	#Added SchemaElement	#Added Property	#Added Class	#Added Typing	#Added Instance
v3.5 -> v3.8	18 469 394 (12 :43 mn)	284 (5 :16 mn)	5 (3 :28 mn)	279 (1 :48 mn)	13 596 447 (6 :43 mn)	2 835 666 (7 :20 mn)
v3.8 ->v3.9	4 813 958 (01 :49 mn)	86 (14 s)	0 (2 s)	86 (12 s)	4 015 870 (1 :21 mn)	1 103 520 (45 s)

TABLE 5 – Type et nombre de changements pour la classe $DeletedStatement$

	#Deleted Statements	#Deleted SchemaElement	#Deleted Property	#Deleted Class	#Deleted Typing	#Deleted Instance
v3.5 -> v3.8	232 058 (12 :43 mn)	7 (29 s)	5 (9.5 s)	2 (3.5 s)	41 788 (6 s)	6 732 (8.6 s)
v3.8 ->v3.9	1 525 420 (35 s)	0 (3 s)	0 (2 s)	0 (2 s)	1 359 525 (23 s)	223 489 (25 s)

Les classes $DeletedInstance$ et $AddedInstance$ permettent d'observer comment les instances évoluent sans se limiter à l'évolution globale du nombre d'instances. Ainsi, si près de trois millions d'instances de $Person$ ont été ajoutées dans v3.8, 14% des instances de v3.5 ont été supprimées. De plus les classes $addedTyping$ et $deletedtyping$ montre que le typage des instances évolue (18% des triplets supprimés sont des typages d'instances). Une requête utilisant les instances supprimées/ajoutées et les types supprimées/ajoutées, montre que de nombreux typages ont évolués pour des instances qui sont présentes dans les deux versions

de la source (donc les ajouts ou suppressions de types ne pas uniquement dus à l'ajout ou suppression d'instances).

L'utilisation de cette approche pour étudier l'évolution des données permet également de détecter que les éléments de l'ontologie utilisés pour décrire ces personnes a évolué. Ainsi, entre v3.5 et v3.8, 284 classes sont apparues dans la description des personnes et deux ont disparu. Notre étude étant extensionnelle, les classes nouvelles correspondent soit à des classes qui ont été ajoutées dans l'ontologie, soit à des classes existantes dont l'extension ne comprenait pas d'instances de $Person$. De plus, les résultats montrent que cinq propriétés sont nouvellement utilisées pour décrire les personnes tandis que cinq autres ont disparu. En fait, les résultats des requêtes correspondantes montrent qu'il s'agit des mêmes propriétés dont l'URI a été modifiée (e.g. la propriété <http ://.../birth est devenu <http ://.../birthDate à partir de la version V5.8). Si l'on compare les deux versions suivantes, les propriétés n'ont pas évoluées. En revanche, 86 classes ont été ajoutées.

Une fois l'ontologie peuplée par les différentes assertions, celle-ci peut être utilisée pour exécuter d'autres requêtes qui se basent sur les classes définies dans O_{DE} . Ainsi, il est facile pour un expert d'étudier l'évolution de la description d'une resource en effectuant des requêtes sur la base de connaissances O_{DE}. Par exemple, nous avons utilisé une requête SPARQL pour rechercher les assertions de type $InstanceDescriptionEnrichment$ qui ont été ajoutées pour l'URI correspondant à Barack Obama (cf table 6). Les résultats ont montré que 7 assertions ont été ajoutées pour cette URI, telle que la propriété $description$ dont la valeur est dans v3.8 "American politician, 44th President of the United States"@en. De plus, en recherchant les assertions de la classe $AddedTyping$ le concernant, nous pouvons voir que cette URI est nouvellement typée par les classes $Agent$ et $OfficeHolder$.

TABLE 6 – Requête listant les nouveaux triplets enrichissant la description de B. Obama

```
SELECT ?property ?value
WHERE{
?node rdf:subject barrack_obama .
?node rdf:type ode::instanceDescriptionEnrichment
.
?node rdf:predicate ?property .
?node rdf:object ?value
.
```

D'autres requêtes ont été exécutées permettant par exemple de lister ou de compter les nouvelles instances d'une sous-classe de personne donnée (utilisation de la classe $addTyping$), ou de lister les instances qui ont été supprimées (utilisation de la classe $deleteTyping$). En executant la requête ci-dessous (voir table 7), nous avons pu voir que 57 595 artistes ont été ajoutés à la version 3.5.

Voici un extrait de la liste des artistes obtenus :
(?instance = <http ://dbpedia.org/resource/Bill$_Reid >)
($?instance =< http : //dbpedia.org/resource/John\_White\_(colonist_and_artist) >$)

$(?instance =< http://dbpedia.org/resource/Louis\_Aragon >)$
$(?instance =< http://dbpedia.org/resource/JeanPierre\_Rampal >)$
$(?instance =< http://dbpedia.org/resource/Pierre\_de\_Marivaux >)$
$(?instance =< http://dbpedia.org/resource/Jean\_Giraudoux >)$
$(?instance =< http://dbpedia.org/resource/Eugene\_Ionesco >)$
$(?instance =< http://dbpedia.org/resource/Eric\_Dolphy >)$
$(?instance =< http://dbpedia.org/resource/Jacob\_Riis >)$
$(?instance =< http://dbpedia.org/resource/Baby\_Gramps >)$
$(?instance =< http://dbpedia.org/resource/Mark\_Mothersbaugh >)$
$(?instance =< http://dbpedia.org/resource/Frank\_Miller_{(}comics) >)$
$(?instance =< http://dbpedia.org/resource/Rosa\_Bonheur >)$
$(?instance =< http://dbpedia.org/resource/Sandro\_Botticelli >)$
$(?instance =< http://dbpedia.org/resource/Thomas\_Lawrence >)$
$(?instance =< http://dbpedia.org/resource/Geoffrey\_A.\_Landis >)$
$(?instance =< http://dbpedia.org/resource/Princess\_(singer) >)$
$(?instance =< http://dbpedia.org/resource/James\_Abbott\_McNeill\_Whistler >)$

TABLE 7 – Requête listant les instances qui ont acquis le type Artiste

```
SELECT ?instance
WHERE{
?node rdf:subject ?instance .
?node rdf:type rdfEvolution:addedTyping .
?node rdf:predicate rdf:type .
?node rdf:object <http://dbpedia.org/ontology/Artist> .
}
```

5 Conclusion

Dans cet article, nous avons présenté une approche originale de détection et de représentation sémantique des changements dans une source RDF. Plus précisément, nous avons conçu une ontologie O^{DE} qui représente et structure les différentes types de changements au niveau des données. A travers des requêtes SPARQL, il est possible pour un expert de données ou à une application réalisant une tâche d'intégration d'interroger l'ontologie peuplée par les triplets ajoutés ou supprimés.
Nous avons mené une expérimentation sur deux versions consécutives de DBPedia ainsi que sur une version ancienne et une versions plus récente. Cette expérimentation a montré que cette approche permet d'étudier l'évolution des données sur différents aspects. Les requêtes simples permettent de lister ou de compter les assertions instances de chacune des classes de O_{DE}. Des requêtes plus complexes, qui utilisent différentes classes de cette ontologie, peuvent être facilement définies pour s'adapter aux besoins d'un expert ou d'une application. De plus, malgré le processus de réification qui induit une représentation assez couteuse des changements, les temps d'execution des requêtes restent acceptables même pour une source qui a subit un nombre très volumineux de changements.

En perspectives, nous souhaitons étendre ce travail de façon à pouvoir détecter des changements dans des sources externes qui seraient liées à une source via des lien SameAs. Ainsi, un expert ou une application pourront être informée que des changements ont été réalisés concernant des resources décrites dans des sources de données externes mais pour lesquelles un lien d'identité a été déclaré (SameAs). Les changements au niveau d'une source pourront ainsi être propagés dans le Web de données en exploitant les liens SameAs entre les données. Nous envisageons, par ailleurs, d'étudier la possibilité d'inférer des changements au niveau des axiomes d'une ontologie,

par exemple la connaissance sur les propriétés fonctionnelles ou encore les cardinalité maximum et minimum de certaines propriétés. Enfin, nous souhaitons intégrer cette nouvelle approche de détection et de représentation des changements dans un processus d'intégration de données plus global où les tâches d'intégration exploiteront directement l'ontologie O^{DE} peuplée par notre approche pour réaliser des tâches d'intégration incrémentales.

Références

DINH D., DOS REIS J. C., PRUSKI C., SILVEIRA M. D. & REYNAUD-DELAÎTRE C. (2014). Identifying relevant concept attributes to support mapping maintenance under ontology evolution. *J. Web Sem.*, **29**, 53–66.

KÄFER T., UMBRICH J., HOGAN A. & POLLERES A. (2012). Dyldo : Towards a dynamic linked data observatory. In *WWW2012 Workshop on Linked Data on the Web, Lyon, France, 16 April, 2012*.

NIKOLOV A., D'AQUIN M. & MOTTA E. (2012). Unsupervised learning of link discovery configuration. In *ESWC*, p. 119–133.

OWENS A., SEABORNE A., GIBBINS N. & MC SCHRAEFEL (2008). Clustered tdb : A clustered triple store for jena.

PAPAVASILEIOU V., FLOURIS G., FUNDULAKI I., KOTZINOS D. & CHRISTOPHIDES V. (2013). High-level change detection in rdf(s) kbs. *ACM Trans. Database Syst.*, **38**(1), 1 :1–1 :42.

SAÏS F., PERNELLE N. & ROUSSET M.-C. (2009). Combining a logical and a numerical method for data reconciliation. *Journal on Data Semantics*, **12**, 66–94.

SYMEONIDOU D., ARMANT V., PERNELLE N. & SAÏS F. (2014). Sakey : Scalable almost key discovery in RDF data. In *The Semantic Web - ISWC 2014 - 13th International Semantic Web Conference, Riva del Garda, Italy, October 19-23, 2014. Proceedings, Part I*, p. 33–49.

VOLZ J., BIZER C., GAEDKE M. & KOBILAROV G. (2009). Discovering and maintaining links on the web of data. In *ISWC*, p. 650–665.

ZABLITH F., ANTONIOU G., D'AQUIN M., FLOURIS G., KONDYLAKIS H., MOTTA E., PLEXOUSAKIS D. & SABOU M. (2015). Ontology evolution : a process-centric survey. *Knowledge Eng. Review*, **30**(1), 45–75.