Atelier Web des Données

Organisateurs
Patricia Serrano Alvarado (LS2N, Université de Nantes),
Olivier Curé (LIGM, Université Paris-Est Marne la Vallée)

PRÉFACE

Les principes du Web des Données (Linked Data) visent à faire évoluer le Web vers un espace de données global où les données ne sont plus constituées en silos mais fortement interconnectées.

Depuis son apparition en 2006, l'attention reçue par le Web des Données a explosé. Il existait alors 12 jeux de données publiées en tant que données ouvertes liées (Linked Open Data) alors que 1205 sont disponibles en 2018. De plus, ceux-ci sont fortement connectés par 16,012 liens sémantiques (https://lod-cloud.net/, juin 2018). Ces données, publiées sous licences ouvertes, concernent des domaines variés : institutions publiques, sciences de la vie, publications scientifiques, média, géographie, réseaux sociaux, etc. Mais ces jeux de données ne sont que la pointe de l'iceberg. De plus en plus de données sont sémantisées et interconnectées dans le domaine du privé car les technologies du web sémantique ouvrent un large panorama d'opportunités pour l'exploitation sémantique des données.

Les avancées du Web des Données ont été nombreuses ces dernières années et elles portent sur des dimensions complémentaires : les bases de connaissances (DBpedia, YAGO, Wikidata, GeoNames, LinkedGeoData, MusicBrainz, Google Knowledge Graph, etc.), les vocabulaires (FOAF, DC, schema.org, SKOS, VoID, PROV, SSN, etc), la standardisation du langage de requêtes SPARQL, de syntaxes RDF (Turtle, TriG, JSON-LD et RDFa), du langage OWL pour décrire les connaissances sur les objets, de PROV pour capturer la provenance, de SKOS pour la description d'information taxonomique, de LDP pour les plateformes de données liées, etc. Ces avancées ont motivé le développement de nouveaux outils (parseurs, moteurs de requêtes, raisonneurs, extracteurs, etc.) qui stimulent les travaux sur l'interconnection automatique de jeux de données et la décentralisation du Web des Données, ainsi que les nouveaux défis qui émergent avec les "mégadonnées" et l'internet des objets.

L'atelier Web des Données vise à partager les connaissances et les expériences sur les nouvelles problématiques associées au développement du Web des Données.

Les contributions de cette première édition sont 2 articles (un long et un court) et 2 démonstrations logicielles. L'article long traite le problème de l'extraction automatique de schéma pour les données décrites par les langages proposées par le W3C, comme RDF, RDFS et OWL. L'article court introduit les travaux en cours du projet SEDELA où l'objectif est la conception et le développement d'une infrastructure décentralisée et sémantique pour l'apprentissage tout au long de la vie. Les deux démonstrations se focalisent sur la sémantisation de données. LODEX est un logiciel ouvert qui facilite la curation et la sémantisation de données brutes pour les connecter au web de données via les normes et les standards du web sémantique. La deuxième démonstration propose une approche d'enrichissement de données RDF intégrées à la volé où des règles d'inférence sont appliquées au mapping permettant de faire l'intégration. Cette contribution est une alternative aux approches de saturation de la base de connaissance ou de réécriture de requêtes.

Membres du comité de lecture

Bernd Amann, LIP6, Université Sorbonne
Khalid Belhajjame, LAMSADE, Université Paris-Dauphine
Jérôme David, Inria Rhône-Alpes, Université Grenoble Alpes
Emmanuel Desmontils, LS2N, Université de Nantes
Sébastien Ferré, IRISA, Université de Rennes 1
Alban Gaignard , Institut du Thorax
François Goasdoué, IRISA, Université de Rennes 1
Luis-Daniel Ibáñez, Université de Southampton
Clément Jonquet, LIRMM, Université de Montpellier
Myriam Lamolle, LIASD, Université Paris 8
Maxime Lefrançois, Laboratoire Hubert Curien, École des Mines de Saint-Étienne
Gabriela Montoya, Université Aalbork
Catherine Roussey, TSCF, Irstea
Fatiha Saïs, LRI, Université Paris
Hala Skaf-Molli, LS2N, Université de Nantes
Antoine Zimmermann, Institut Henri Fayol, École des Mines de Saint-Étienne

TABLE DES MATIÈRES

Extraction automatique de schéma pour des données massives

Redouane BOUHAMOUM, Zoubida KEDAD
Stéphane LOPES

DAVID - Université de Versailles Saint-Quentin-en-Yvelines
Versailles, France
prénom.nom@uvsq.fr

Résumé. Nous nous intéressons dans ce travail au problème d'extraction automatique de schéma pour les données du Web sémantique. La flexibilité des langages utilisés dans ce contexte, comme le langage RDF, peut rendre l'exploitation des données difficile. En effet, le schéma pour ces sources de données peut être incomplet ou manquant et, même s'il est présent, les données ne sont pas contraintes par ce schéma et ne sont pas tenues de le respecter. Nous avons proposé dans des travaux précédents une représentation condensée pour réduire la taille d'un jeu de données RDF en vue d'extraire son schéma. Cependant, pour certaines sources de données particulièrement hétérogènes, la taille de cette représentation demeure trop importante. Nous proposons dans ce papier SC-DBSCAN, un algorithme d'extraction de schéma inspiré de DBSCAN. La conception distribuée de notre algorithme le rend efficace sur de grandes sources de données RDF.

1 Introduction

Une quantité croissante de données sont disponibles sur le Web, décrites par les langages proposées par le W3C, comme RDF, RDFS et OWL. Ces langages permettent une description flexible des données, car ils n'imposent pas de structure à laquelle les instances doivent se conformer, contrairement aux bases de données relationnelles. Cette flexibilité lors de la création des données rend leur exploitation difficile. En effet, il peut exister des jeux de données où le schéma est incomplet ou absent. De plus, même si le schéma existe, les données ne sont pas contraintes de le respecter.

La découverte d'un schéma implicite décrivant les classes et les propriétés des entités du jeu de données est utile pour l'utilisation de ces sources de données. Certaines approches ont proposé l'utilisation d'algorithmes de clustering dans le but de regrouper les entités similaires en clusters représentant les classes du schéma. Cependant, l'utilisation de ces approches pour de grandes sources de données reste impossible à cause de la complexité des algorithmes utilisés.

Dans notre travail, nous abordons le problème du passage à l'échelle de la découverte de schéma pour les jeux de données RDF. Dans des travaux antérieurs, nous avons proposé une représentation condensée pour les données RDF sur laquelle un algorithme de clustering peut

être appliqué [Bouhamoum et al. (2018)]. Cependant, lorsque les entités sont décrites par des ensembles de propriétés très hétérogènes, la taille de la représentation condensée demeure trop importante.

Nous proposons dans ce travail SC-DBSCAN, un algorithme de clustering basé sur la densité et scalable. Cet algorithme est inspiré de l'algorithme de clustering DBSCAN [Ester et al. (1996)] et fournit les mêmes résultats que ce dernier. SC-DBSCAN est conçu et mis en œuvre dans un contexte Big Data pour assurer le passage à l'échelle. Il comprend les étapes suivantes : (i) les données sont partitionnées en fonction des propriétés décrivant les entités, (ii) les voisinages de chaque entité sont calculés au sein de chaque partition, (iii) des clusters partiels sont construits dans chaque partition, et (iv) les clusters partiels sont fusionnés.

La suite de cet article est structurée comme suit. La section 2 présente notre problématique. Notre approche est détaillée dans la section 3 et les expérimentations sont présentées dans la section 4. La section 5 présente les travaux existants sur l'extraction de schéma et sur le passage à l'échelle de DBSCAN. Une conclusion est présentée en section 6.

2 Problématique

Un jeu de données D est défini par un ensemble de triplets RDF, $D \subseteq (R \cup B) \times P \times (R \cup B \cup L)$ où R, B, P et L représentent respectivement des ressources, nœuds vides (ressources anonymes), propriétés et littéraux. Dans un jeu de données RDF, tout nœud excepté les littéraux représente une entité. Les données publiées au format RDF peuvent être décrites par un schéma exprimé en RDF, RDFS ou OWL. Cependant, le langage RDF n'impose pas de contrainte sur la conformité des données au schéma, ce qui complique considérablement la compréhension et l'utilisation de ces jeux de données.

Différents travaux portant sur l'extraction de schéma ont été publiés, où les auteurs proposent d'utiliser des algorithmes de clustering afin de grouper dans une même classe les entités présentant des propriétés similaires [Kellou-Menouer et Kedad (2015); Kellou-Menouer et Kedad (2016); Christodoulou et al. (2013)]. L'utilisation de ces approches sur de grands jeux de données est impossible à cause de la complexité des algorithmes de clustering utilisés.

Parmi les algorithmes proposés, le clustering basé sur la densité répond aux exigences de l'extraction de schéma sur des données RDF. Tout d'abord, il permet de former des clusters de forme arbitraire, ce qui est important dans ce contexte où les entités peuvent être décrites par des ensembles de propriétés hétérogènes bien qu'elles soient de même type. Ensuite, il n'exige pas de préciser à priori le nombre de clusters, ce qui est également important, car le nombre de classes n'est en général pas connu. Enfin, il fournit un résultat déterministe et détecte le bruit, i.e. les entités qui ne sont pas assez importantes pour former un cluster. Toutefois, la question du passage à l'échelle de ce type d'algorithmes et leur utilisation dans un contexte big data restent un défi.

Notre problème peut donc être énoncé comme suit : étant donné un grand jeu de données RDF, comment regrouper les entités structurellement similaires pour former des classes et produire un schéma décrivant les données ? Des entités sont structurellement similaires si elles ont un certain nombre de propriétés en commun. Les mesures de similarité sont donc basées sur le nombre de propriétés partagées entre deux entités comme par exemple la similarité de Jaccard [Wikipedia].

3 SC-DBSCAN

Nous décrivons dans cette section notre approche de découverte de schéma pour les grands jeux de données RDF. Les différentes étapes de notre proposition sont présentées dans la figure 1.

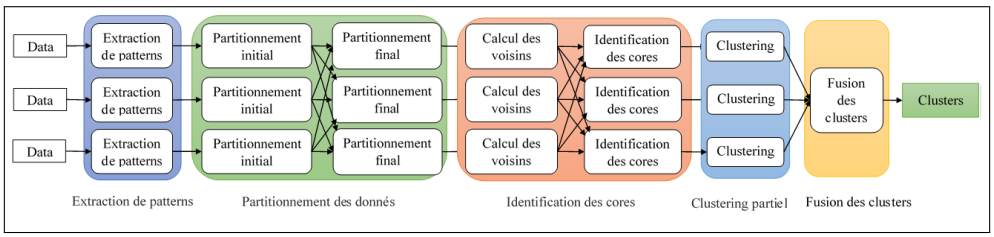


FIG. 1 – *Les étapes de SC-DBSCAN.*

Notre approche consiste tout d'abord à extraire une représentation condensée de l'ensemble de données RDF. Les étapes suivantes de la découverte de schéma seront effectuées sur cette représentation condensée au lieu du jeu de données initial. Cette étape consiste à extraire un ensemble de *patterns* représentant la structure des entités du jeu de données.

Définition 1. *Un* pattern Pt *est un ensemble de propriétés distinctes tel qu'il existe au moins une entité décrite par l'ensemble de propriétés* Pt*.*

L'extraction de patterns d'un jeu de données produit en sortie toutes les structures (ensembles de propriétés) décrivant les entités du jeu de données. À chaque pattern est associé le nombre d'entités ayant la même structure que ce pattern. Le clustering est ensuite appliqué sur les patterns au lieu des entités pour permettre une exécution plus rapide tout en conservant la même qualité du schéma obtenu.

SC-DBSCAN est un algorithme de clustering distribué et déterministe basé sur la densité, inspiré de DBSCAN. Il permet de calculer efficacement les classes d'un jeu de données RDF et fournit les mêmes résultats que DBSCAN. SC-DBSCAN partitionne d'abord les données, identifie les *core patterns* (patterns ayant un nombre de voisins supérieur à un certain seuil), construit les clusters en parallèle dans chaque partition et enfin, fusionne les clusters partiels produits dans chaque partition pour fournir le résultat final.

Le partitionnement des données est fondé sur l'idée que des patterns similaires partagent au moins une propriété. Les partitions regroupent des patterns ayant des propriétés en commun en garantissant que les patterns similaires seront comparés au moins une fois.

En raison du partitionnement des patterns, le voisinage d'un pattern peut être réparti sur différentes partitions, empêchant ainsi l'identification des *core patterns*. Pour résoudre ce problème, SC-DBSCAN calcule le voisinage de chaque pattern avant l'étape du clustering, en assurant ainsi l'attribution à chaque pattern du rôle approprié (core, bordure ou bruit). Cette étape est effectuée en parallèle dans chaque partition, puis les voisins locaux découverts sur chaque partition sont regroupés par pattern, enfin les core patterns sont identifiés.

En utilisant les core patterns, des clusters partiels sont calculés dans chaque partition en parallèle, sans échange d'informations entre les nœuds de calcul. Les clusters finaux sont formés en fusionnant les clusters partiels qui ont des patterns en commun.

SC-DBSCAN est implémenté en utilisant Spark, un framework de calcul distribué adapté au traitement de grands ensembles de données. Le reste de cette section détaille notre proposition.

3.1 Extraction de patterns

Notre approche pour réduire la taille du jeu de données initial consiste à extraire un ensemble de patterns formant une représentation condensée des données.

Tout d'abord, le jeu de données est divisé et distribué sur les nœuds de calcul. À partir des triples RDF, l'identifiant du sujet et les propriétés des entités sont extraits pour former des paires de la forme $(entityID, property)$. Toutes les propriétés d'une même entité sont ensuite regroupées pour composer les entités et produire les paires $(entityID, \{p_1, p_2, p_3, \ldots\})$.

Les patterns sont ensuite extraits pour constituer un ensemble de paires $(pattern, nb)$, nb étant le nombre d'entités décrites par le pattern. Pour cela, les paires $(entityID, \{p1, p2, p3, \ldots\})$ sont lues et le résultat $(pattern, 1)$ est généré, le pattern constituant l'ensemble des propriétés d'une entité. Le nombre 1 indique qu'une entité correspondant à ce modèle a été trouvée.

Enfin, le nombre d'entités décrites par un pattern est calculé en regroupant toutes les paires $(pattern, 1)$ ayant la même clé. À la fin de cette étape, la liste des patterns et le nombre d'entités pour chacun sont obtenus.

Étant donné que le clustering est basé sur la structure des entités et que la similarité est évaluée en fonction des propriétés les décrivant, le clustering de l'ensemble de patterns fournit le même schéma que celui produit par le clustering sur l'ensemble des entités.

La figure 2 représente un jeu de données RDF et les patterns correspondant aux entités du jeu de données. Par exemple, le pattern pt_1 représente les entités e_1 et e_2 décrites par le même ensemble de propriétés $\{b, c\}$.

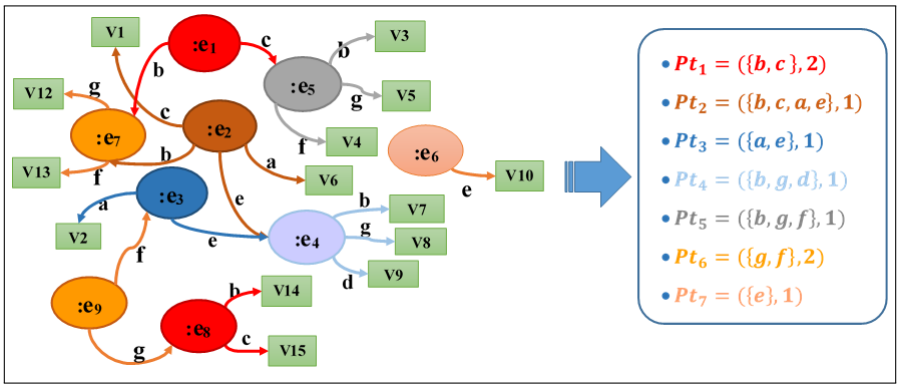


FIG. 2 – *Example d'un jeu données RDF et les patterns correspondant.*

3.2 Partitionnement des données

Le partitionnement des données joue un rôle important dans le traitement efficace des grands jeux de données. Cela permet de distribuer correctement les calculs sur les nœuds d'un

cluster. Dans notre contexte, il assure la division du jeu de données initial en sous-ensembles pour générer des tâches de clustering pouvant être traitées en parallèle. Lors du calcul des clusters, un partitionnement adéquat limite également les coûts de communication entre les partitions : le clustering des patterns dans une partition ne nécessite aucune donnée située dans une autre partition et il n'y a donc pas de transfert de données entre les noeuds de calcul. Enfin, le partitionnement des données fournit assez d'informations pour fusionner les clusters partiels. Notre méthode de partitionnement génère des partitions non disjointes et les patterns dupliqués sont utilisés pour fusionner les clusters partiels.

Dans notre approche, une partition est créée pour chaque propriété impliquée dans un pattern et contient tous les patterns décrits par cette propriété. De cette façon, tous les patterns qui pourraient être similaires sont regroupés dans la même partition. Les patterns qui ne se trouvent jamais dans la même partition ne partagent aucune propriété et leur comparaison est donc inutile.

Définition 2. *Une partition est un sous-ensemble des patterns obtenus à partir d'un jeu de données, choisis en fonction d'une propriété donnée. Partitionner un ensemble de patterns produit l'ensemble de partitions* $partitionSet = \{part_{p_x} \mid p_x \in P\}$ *où* P *est l'ensemble de toutes les propriétés du jeu de données et où* $part_{p_x}$ *contient tous les patterns décrits par la propriété* p_x.

Si on considère l'ensemble de patterns obtenus dans l'exemple précédant, le partitionnement produit les partitions présentées dans la figure 2.

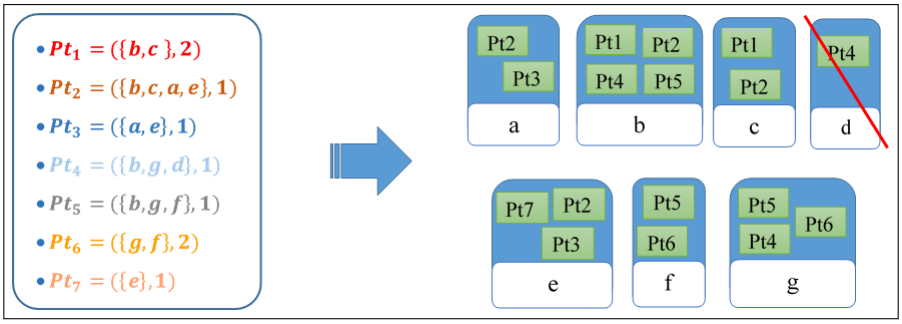


FIG. 3 – *Partitionement de l'ensemble de patterns.*

Le nombre d'éléments d'une partition $part_{p_x}$ peut être supérieur à la capacité d'un nœud de calcul. Cela rend le clustering de $part_{p_x}$ coûteux voire impossible. Dans ce cas, cette partition est subdivisée en fonction d'autres propriétés (autre que p_x). Ainsi, pour subdiviser $part_{p_x}$, une sous-partition est créée pour chaque propriété autre que p_x, et les patterns de $part_{p_x}$ sont distribuées sur les différentes sous-partitions par rapport aux propriétés les décrivant. Récursivement, les partitions dépassant la capacité de calcul sont subdivisées à nouveau jusqu'à ce que toutes aient un nombre d'éléments inférieur à la capacité. Dans notre exemple précédent, si la capacité d'un nœud est fixée à 3, la partition $part_b$ dépasse cette capacité. Elle est donc subdivisée en sous-partitions comme présenté dans la figure 4.

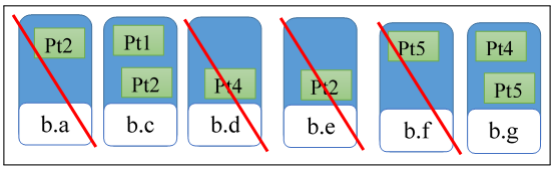


FIG. 4 – *Partitionnement de* $part_b$

3.3 Identification des cores

Dans SC-DBSCAN, le clustering est exécuté sur les patterns au lieu des entités. La définition d'un *core pattern* doit tenir compte du nombre d'entités représentées par ce pattern. Les autres patterns sont soit des *bordures*, i.e. voisins d'un core pattern, soit du bruit, i.e. n'appartenant à aucun cluster.

Définition 3. *Soient* ϵ *un seuil de similarité et* $minPts$ *un nombre d'entités. Un pattern est un* core pattern *si la somme de son nombre d'entités et du nombre d'entités des patterns dans son* ϵ*-voisinage est supérieure au seuil* $minPts$*.*

Tout d'abord, pour chaque pattern, la liste de ses voisins dans chaque partition est calculée en parallèle. Ensuite, pour chaque pattern, tous les voisins trouvés dans chaque partition sont regroupés pour constituer la liste complète de ses voisins. Enfin, les patterns ayant une somme d'entités supérieure ou égale à $minPts$ sont définis comme étant des core patterns. Ce processus garantit que les rôles attribués à chaque pattern sont les mêmes que ceux qui auraient été attribués sans partitionner les données.

Si $\epsilon = 0.5$ et $minPts = 3$ dans notre exemple, les core patterns identifiés sont pt_2 et pt_4.

3.4 Clustering partiel au sein des partitions

Les core patterns fournissent suffisamment d'informations pour former les clusters partiels dans chaque partition. Seuls les core patterns produisent un cluster en ajoutant leurs voisins en tant qu'éléments du cluster. Les autres patterns sont soit des bordures dans le voisinage d'un core qui seront affectées au cluster, soit du bruit.

Pour chaque core pattern pt_i, un cluster c_i contenant pt_i et ses voisins est créé. Ensuite, parmi les patterns ajoutés au cluster, les cores sont sélectionnés et leurs voisins ajoutés au cluster c_i. Les clusters partiels sont identifiés en répétant ce processus de manière récursive sur les patterns nouvellement ajoutés jusqu'à ce qu'un pattern bordure soit trouvé. Tous les patterns non affectés à un cluster sont considérés comme du bruit.

La figure 5 présente les clusters calculés sur chaque partition.

3.5 Fusion des clusters partiels

La fusion des clusters partiels vise à identifier les clusters qui s'étendent sur plusieurs partitions et à les fusionner.

Un pattern pt_x est attribué à un cluster c_i si pt_x est *density-reachable* à partir d'un core pattern de c_i.

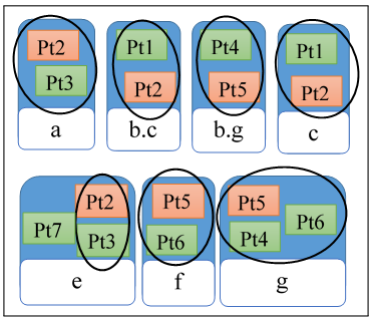


FIG. 5 – *Les clusters partiel obtenu sur chaque partition*

Définition 4. *(Density-reachable) Un pattern* pt_d *est* $density - reachable$ *à partir d'un pattern* pt_s *s'il existe une chaîne de patterns* $pt_1, \ldots, pt_z$ *avec* $pt_1 = pt_s$, $pt_z = pt_d$ *tel que* pt_{i+1} *se trouve dans le voisinage de* pt_i.

Si ce même pattern pt_x est affecté à un autre cluster c_j, cela signifie qu'il est density-reachable à partir d'un core pattern de c_j. Si pt_x est un core, cela constituerait un pont entre les clusters c_i et c_j. Ainsi, ces patterns doivent être affectés au même cluster et donc c_i et c_j doivent être fusionnés.

L'étape de fusion identifie les clusters qui s'étendent sur différentes partitions en recherchant les clusters locaux qui ont un core pattern en commun et en fusionnant ces clusters pour obtenir le résultat final.

Si un pattern bordure est affecté à différents clusters au cours de la phase du clustering, il sera attribué de manière aléatoire à l'un de ces clusters lors de la fusion.

SC-DBSCAN garantit un résultat de clustering identique à l'utilisation du DBSCAN séquentiel.

Considérant les clusters partiels obtenus dans notre exemple, les clusters des partitions $part_a$ et $part_{b.c}$ sont fusionnés puisqu'ils partagent le core pattern pt_2. Les clusters finaux sont présentés dans la figure 6 et représentent les classes du schéma ($Class1$ et $Class2$).

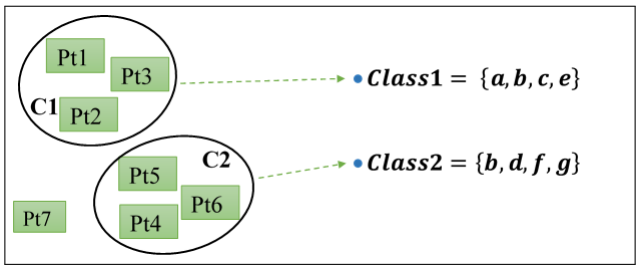


FIG. 6 – *Le resultat final du clustering.*

4 Expérimentations

Nous présentons dans cette section les évaluations de notre approche afin de valider son efficacité. Pour nos expérimentations, nous utilisons Apache Spark 2.3 installé sur un cluster de calculs de 5 nœuds équipés de 32Go de RAM.

Nous détaillons en premier lieu les évaluations effectuées sur la représentation condensée des données RDF pour montrer la rapidité du processus et le ratio de réduction. Par la suite, nous exposerons les résultats concernant notre algorithme de clustering SC-DBSCAN.

4.1 Réduction du jeux de données

Nous avons évalué notre représentation condensée sur différents jeux de données RDF réels : DBpedia [1] qui est une extraction de Wikipedia au format RDF ; DBLP [2] qui contient des données sur plus de 1.8 million de publications scientifiques ; Katrina et Charley [3] qui représentent des données d'observations de cyclones et tornades aux États-Unis.

Le tableau 1 montre, pour chaque jeu de données, le nombre de triplets RDF qui constituent les données, le nombre d'entités, le nombre de patterns extraits, et le temps nécessaire pour l'extraction des patterns.

Source	**Triples**	**Entités**	**Patterns**	**Temps (s)**
DBpedia	9 500 000 000	66 195 296	1 918 480	750
DBLP	222 375 855	16 086 516	351	163
Katrina	203 386 049	3 409	37	100
Charley	101 956 760	3 353	52	50

TAB. 1 – *Évaluations de la construction d'une représentation condensée.*

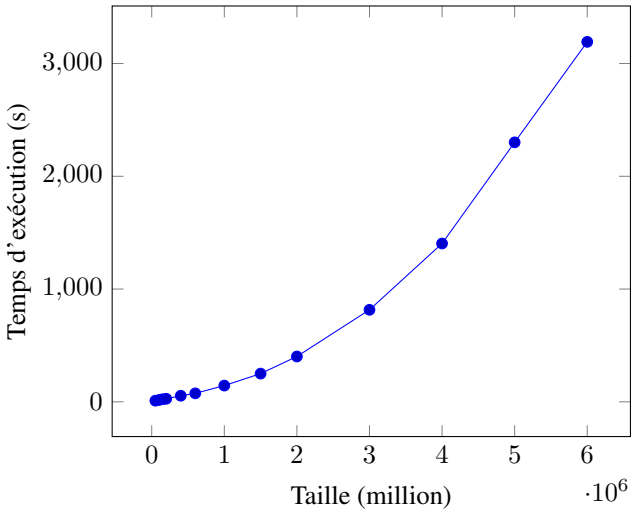
Le nombre de patterns qui constituent la représentation condensée est lié à l'hétérogénéité des entités : plus elles sont décrites par des ensembles de propriétés hétérogènes, plus le nombre de patterns est grand. Si on considère DBpedia, le nombre élevé de patterns s'explique par le fait que la source contient des entités très hétérogènes, contrairement à DBLP, Katrina et Charley qui sont moins hétérogènes et produisent un nombre réduit de patterns.

Nos évaluations montrent que pour certains jeux de données comme DBLP, Katrina et Charley, la taille initiale a été considérablement réduite, permettant une extraction de schéma en utilisant des algorithme de clustering existants. Cependant, pour des sources dont l'hétérogénéité est élevée comme DBpedia, l'extraction de schéma reste difficile.

En considérant le temps nécessaire pour extraire les patterns, les évaluations montrent que notre approche est capable de traiter de grandes sources de données dans un temps réduit (environ 12 minutes pour DBpedia qui est composée de plus de 9 milliards de triplets).

1. https ://old.datahub.io/dataset/dbpedia
2. https ://old.datahub.io/dataset/dblp
3. http ://wiki.knoesis.org/index.php/LinkedSensorData

FIG. 7 – *Scalabilité de SC-DBSCAN*



4.2 Clustering

Comme SC-DBSCAN produit exactement le même résultat que DBSCAN, nos évaluations porterons uniquement sur sa capacité à extraire un schéma sur de grands jeux de données.

Nous avons tout d'abord analysé le passage à l'échelle en utilisant des jeux de données de tailles différentes. Ensuite, nous avons étudié l'influence du nombre de propriétés décrivant les patterns.

Pour mener à bien ces différentes évaluations en contrôlant les paramètres, nous avons utilisé des jeux de données synthétiques générés en utilisant le générateur "IBM Quest Synthetic Data Generator" [IBM].

Dans ce qui suit, nous avons fixé $minPts$ à 3 et ϵ à 0.8 et nous utilisons, pour évaluer la similarité entre deux patterns, la formule de Jaccard (rapport entre le cardinal de l'intersection des propriétés de deux patterns et le cardinal de l'union de ces propriétés.

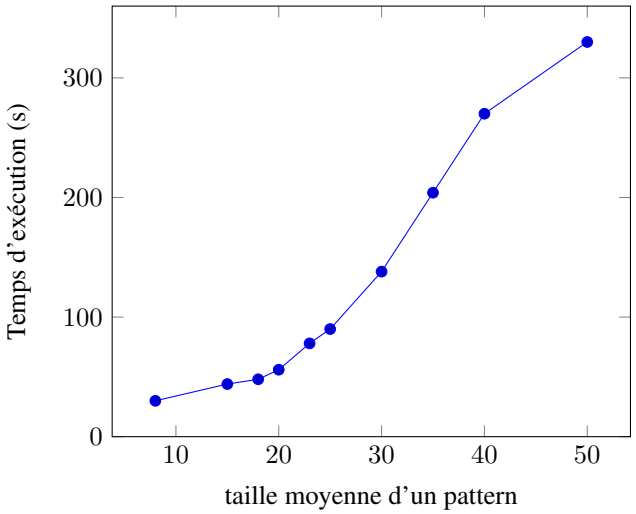
$$J(P1, P2) = \frac{|Prop(P1) \cap Prop(P2)|}{|Prop(P1) \cup Prop(P2)|}$$

Nous avons exécuté SC-DBSCAN sur des jeux de données de tailles différentes, chacun d'eux contenant des entités décrites en moyenne par 15 propriétés, i.e. la dimension moyenne d'une entité est de 15. La figure 2 présente les temps d'exécution de SC-DBSCAN.

Les résultats montrent l'efficacité de notre algorithme pour traiter de grands jeux de données. Le clustering d'un jeux de données contenant 5 millions d'entités prend environ 38 minutes. En comparaison, l'application de NG-DBSCAN sur un jeu de données de la même taille nécessite 30 minutes mais fournit un résultat approximatif [Lulli et al. (2016)].

Ces résultats s'expliquent par le fait que la première étape de SC-DBSCAN génère des partitions contenant un nombre de patterns pouvant être traité par un seul nœud du cluster :

FIG. 8 – *Influence de la dimensions des données.*



chaque nœud doit traiter un nombre de patterns inférieur à sa capacité dans une tâche. En outre, SC-DBSCAN ignore certaines comparaisons inutiles lors de la recherche du voisinage de chaque pattern, car les patterns ne sont comparés que s'ils partagent au moins une propriété.

Comme nous l'avons expliqué, notre approche de partitionnement est fondée sur les propriétés décrivant les données. Par conséquent, le nombre de propriétés influe sur le comportement de notre algorithme. La figure 3 montre les performances de SC-DBSCAN lorsque le nombre de propriétés varie. Pour cela, nous avons effectué des évaluations sur un jeu de données de 150000 patterns avec une capacité de 2000 en faisant varier la moyenne du nombre de propriétés décrivant un pattern (la dimension d'un pattern).

Le partitionnement distribue les patterns par rapport aux propriétés les décrivant. Plus un pattern est décrit un nombre important de propriétés, plus il sera distribué sur plusieurs partitions augmentant ainsi la charge des partitions, ce qui nécessite des divisions hiérarchiques sur plusieurs niveaux. Cela ce traduit par la génération d'un grand nombre de partitions, ce qui explique l'évolution de la courbe en augmentant la taille des patterns.

5 État de l'art

La découverte de schéma sur les jeux de données RDF a été abordée dans plusieurs travaux de recherche. Certain travaux proposent d'utiliser des algorithme de clustering afin de grouper les entités similaires dans des clusters représentant les classes du schéma. L'approche proposée dans [Kellou-Menouer et Kedad (2015); Kellou-Menouer et Kedad (2016)] utilise DBSCAN afin de regrouper les entités similaires représentant les differents type inclus dans un jeux de données RDF. Dans [Christodoulou et al. (2013)], les auteurs proposent d'utiliser le clustering hiérarchique pour la découverte des classes composant le schéma. Cependant, l'utilisation de ces approches sur de grandes masses de données est impossible à cause de la complexité des

algorithmes de clustering. D'autres approches ont été proposées pour le traitement de grandes masses de données RDF et ont été implémentées en utilisant des technologies big data [Ruiz et al. (2015); Baazizi et al. (2017)]. Ces approches utilisent les déclarations de type existantes dans le jeu de données RDF pour définir les entités similaires et ne s'appliquent pas dans les cas où ces déclarations sont absentes.

L'algorithme DBSCAN a été largement utilisé et étendu pour assurer sa scalabilité en proposant des versions parallèles. Dans [Patwary1 et al. (2012)], les données sont partitionnées de manière aléatoire et le clustering est ensuite appliqué sur chaque partition en parallèle en comparant les entités d'une partition avec l'ensemble du jeu de données. Dans [Luo et al. (2016)], S-DBSCAN partitionne les données aléatoirement, puis calcule les clusters dans chaque partition. Les clusters dont les centres sont proches les uns des autres sont ensuite fusionnés. L'approche proposée dans [Savvas et Tselios (2016)] est assez similaire à S-DBSCAN, mais fusionne les clusters dont l'intersection est non vide. Après le partitionnement et le calcul des clusters partiels dans chaque partition, [Han et al. (2016)] définit un intervalle pour chaque partition et considère les points hors de cet intervalle comme base pour fusionner les clusters partiels. MR-DBSCAN partitionne les données à l'aide du partitionnement binaire de l'espace, duplique les frontières de chaque partition dans les partitions voisines et calcule les clusters [HE et al. (2013)]. Les clusters sont finalement fusionnés s'ils partagent certains points. NG-DBSCAN procède en deux étapes [Lulli et al. (2016)] : tout d'abord, l'ϵ-graphe est calculé en comparant chaque point avec k points choisis aléatoirement et un arc est ajouté entre les plus proches. Ensuite, les sommets ayant le plus grand nombre de voisins sont considérés comme racine du cluster et tous les éléments connectés à cette racine constituent un même cluster.

Les versions existantes de DBSCAN scalable présentent certaines limitations : (i) PDS-DBSCAN compare une partition à l'ensemble de données, ce qui nécessite de dupliquer la totalité du jeux de données dans tous les nœuds de calcul, (ii) NG-DBSCAN est un algorithme probabiliste et ne fournit pas le même résultat que le DBSCAN séquentiel ; la même limitation existe avec S-DBSCAN et l'approche proposée dans [Savvas et Tselios (2016)], qui repose sur les centres pour fusionner les clusters partiels, (iii) il n'existe pas d'ordre relatif sur les ensembles de données Web, comme requis dans [Han et al. (2016)], (iv) MR-DBSCAN utilise le partitionnement binaire de l'espace, qui n'est pas adapté aux données de grande dimensionnalité telles que les jeux de données RDF.

6 Conclusion

Nous avons proposé SC-DBSCAN, une approche d'extraction de schéma à partir des données du web sémantique et applicable sur de grandes masses de données.

Afin d'atteindre cet objectif, nous avons proposé en un premier lieu de condenser les données RDF afin de réduire leur taille pour le clustering. Ensuite, nous avons proposé un partitionnement des données permettant de concevoir un algorithme distribué de clustering basé sur la densité. Nos expérimentations ont montré que SC-DBSCAN permet d'extraire un schéma à partir d'un grand jeu de données RDF.

Nous travaillons sur l'optimisation de SC-DBSCAN en améliorant l'étape de partitionnement afin de produire un nombre minimum de partitions et d'éviter les comparaisons inutiles.

Références

Baazizi, M.-A., H. B. Lahmar, D. Colazzo, G. Ghelli, et C. Sartiani (2017). Schema inference for massive json datasets. *EDBT*.

Bouhamoum, R., K. K. Kellou-Menouer, S. Lopes, et Z. Kedad (2018). Scaling up schema discovery approaches. *International Conference on Data Engineering Workshops*.

Christodoulou, K., N. W. Paton, et A. A. Fernandes (2013). Structure inference for linked data sources using clustering. *EDBT/ICDT*.

Ester, M., H.-P. Kriegel, J. Sander, et X. Xu (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. *KDD*.

Han, D., A. Agrawal, W. Liao, et A. Choudhary (2016). A novel scalable dbscan algorithm with spark. *International Parallel and Distributed Processing Symposium Workshops*.

HE, Y., H. TAN, W. LUO, S. FENG, et J. FAN (2013). Mr-dbscan : a scalable mapreduce-based dbscan algorithmfor heavily skewed data. *International Parallel and Distributed Processing Symposium Workshops*.

IBM. Ibm quest synthetic data generator. `https://sourceforge.net/projects/ibmquestdatagen/files/latest/download`. Accessed : 2018-10-01.

Kellou-Menouer, K. et Z. Kedad (2015). Schema discovery in RDF data sources. In *Conceptual Modeling - 34th International Conference, ER*, pp. 481–495. Springer.

Kellou-Menouer, K. et Z. Kedad (2016). A self-adaptive and incremental approach for data profiling in the semantic web. *T. Large-Scale Data- and Knowledge-Centered Systems 29*, 108–133.

Lulli, A., M. Dell'Amico, P. Michiardi, et L. Ricci (2016). Ngdbscan :scalable density based clustering forarbitrary data. *VLDB*.

Luo, G., X. Luo, et T. F. Gooch (2016). A parallel dbscan algorithm based on spark. *BDCloud*.

Patwary1, M. M. A., D. Palsetia, A. Agrawal, W. k. Liao, F. Manne, et A. Choudhary (2012). A new scalable parallel dbscan algorithm using the disjoint-set data structure. *International Conference for High Performance Computing, Networking, Storage and Analysis*.

Ruiz, D. S., S. F. Morales, et J. G. Molina (2015). Inferring versioned schemas from nosql databases and its applications. *ER*.

Savvas, I. K. et D. Tselios (2016). Parallelizing dbscan algorithm using mpi. *International Conference on Enabling Technologies : Infrastructure for Collaborative Enterprises*.

Wikipedia. Jaccard index. `https://en.wikipedia.org/wiki/Jaccard_index`. Accessed : 2018-10-15.

Summary

The problem addressed in this paper is automatic schema discovery for semantic Web data. More and more datasets are published on the Web, in languages such as RDF. These languages provide a high flexibility: the schema describing the data can be incomplete or missing, and even if provided, the data is not constrained by this schema. Several approaches have been

proposed in order to discover the underlying schema for an RDF dataset; in our work, we focus on the scalability issues raised by such approaches. In previous works, we have proposed an approach for building a condensed representation of an RDF dataset prior to schema discovery. However, for some datasets, the size of this condensed representation remains too large. In this paper, we propose SC-DBSCAN, an approach for automatic schema discovery inspired from DBSCAN. SC-DBSCAN is suitable for large RDF dataset. We present an implementation of our approach using big data technology along with some experiments to show the effectiveness of our approach.

Infrastructure décentralisée et sémantique pour l'apprentissage tout au long de la vie

Hala Skaf-Molli* Patricia Serrano-Alvarado**
Sara El hassad**, Emmanuel Desmontils**, Pascal Molli**

LS2N – University of Nantes, France
* {Hala.Skaf}@univ-nantes.fr ** {Name.LastName@}univ-nantes.fr

Résumé. L'apprentissage tout au long de la vie joue un rôle fondamental pour le développement professionnel des personnes. Les environnements tels que les portfolios numériques ou les environnements d'apprentissages personnels permettent aux apprenants d'acquérir de l'autonomie, élément essentiel pour cet apprentissage. Cependant, les services proposés par ces environnements se limitent généralement à la simple collecte de faits ou de preuves à propos des apprentissages effectués. De plus, ces services ne permettent pas d'effectuer des activités collaboratives et sont très souvent peu pérennes. Il est donc nécessaire de disposer d'une infrastructure flexible d'apprentissage qui va au-delà de la simple collecte de faits en offrant une intégration continue des données d'apprentissage et qui permet une collaboration en confiance entre les apprenants. Dans ce document, nous montrons comment le Web sémantique pourra aider à créer une infrastructure d'apprentissage respectant ces critères.

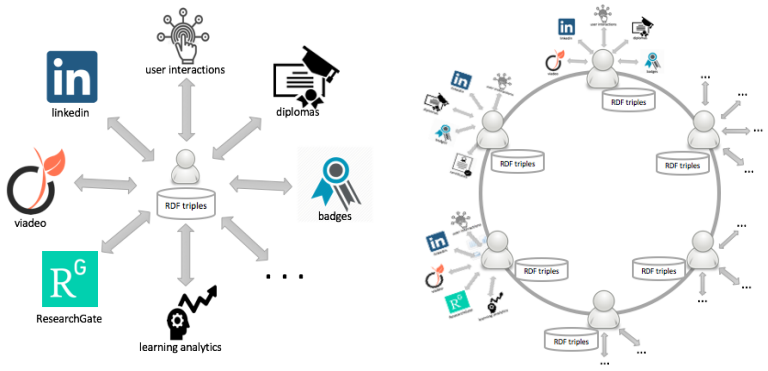
1 Introduction

De nos jours, l'apprentissage se déroule durant toute la vie d'une personne. Il commence lors de la formation initiale, de l'école jusqu'à l'université, et se poursuit pendant toute sa carrière, avec les différents emplois qu'il peut occuper. L'autonomisation des apprenants permet à chacun de définir son propre parcours d'apprentissage, ce qui est une condition préalable pour l'autonomie. Les dimensions identifiées de l'apprentissage tout au long de la vie dans les processus d'apprentissage autonome sont : la capitalisation des expériences, la reconnaissance, la gestion des objectifs, la gestion de l'apprentissage personnel et la collaboration. Les différentes infrastructures personnelles d'apprentissage (Van Harmelen, 2006) et les portfolios électroniques permettent de soutenir l'apprentissage autonome. Cependant, ils devraient être fonctionnellement étendus et synchronisés pour rassembler et exploiter les informations pertinentes pour les différentes dimensions de l'apprentissage tout au long de la vie.

Il n'existe actuellement aucune infrastructure fournissant un environnement de confiance pour l'apprentissage à long terme, la capitalisation des données et la gestion de l'apprentissage personnel (El Mawas et al., 2017). Nous croyons que les technologies du Web sémantique profiteraient à l'apprentissage autonome tout au long de la vie

sous plusieurs aspects. Le modèle de données RDF améliorera l'échange de données d'apprentissage sur le Web. Les ontologies des apprenants (Yang et al., 2006; Rezgui et al., 2017; Evangelos Katis et Vassilakis, 2018) permettront une intégration fluide des données d'apprentissage personnelles provenant de sources de données multiples et hétérogènes. Le langage de requête SPARQL permettra d'interroger les données personnelles d'apprentissage et d'autres données grâce aux requêtes fédérées.

Nous croyons que l'apprentissage autonome tout au long de la vie nécessite une infrastructure flexible et décentralisée. La décentralisation permet à chaque apprenant de garder le contrôle de ses données et elle élimine la nécessité d'un tiers qui doit être de confiance. Chaque apprenant doit pouvoir gérer ses données personnelles comme il le souhaite. La construction d'une infrastructure en réseau pair-à-pair dans le contexte de l'éducation a été proposée par Nejdl et al. (2002). Toutefois, ces travaux ne considèrent pas le large éventail des données collectées, le contexte d'une formation continue et d'une collaboration en confiance.



(a) Informations personnelles (b) Réseau social d'apprentissage

FIG. 1: Infrastructure d'apprentissage décentralisée et sémantique

2 Infrastructure décentralisée et sémantique

Nous envisageons une *infrastructure d'apprentissage décentralisée et sémantique* où chaque apprenant intègre ses données d'apprentissage personnelles dans un espace privé et collabore d'une manière personnalisée avec d'autres apprenants en partageant des données par le biais d'un ensemble de services fiables. La figure 1 montre une vision globale de l'infrastructure. L'espace privé présenté dans la figure 1a est composé de l'ensemble des informations pertinentes collectées par un apprenant. Chaque apprenant peut composer son espace privé différemment des autres et contrôler totalement les informations qui seront partagées avec les autres apprenants du fait de l'architecture décentralisée, comme illustré dans la figure 1b.

Nous proposons d'utiliser Solid [1] comme infrastructure de stockage. Chaque apprenant stocke ses données RDF dans une banque de données personnelle accessible sur le Web (Mansour et al., 2016). Les données sont collectées pendant les processus d'apprentissage, par exemple, les diplômes d'un apprenti récupérés de l'université et ses emplois collectés de LinkedIn et Viadeo. Pour gérer l'hétérogénéité des données collectées, nous utilisons les techniques d'intégration de données sémantiques. Pour la flexibilité, nous proposons deux services d'intégration de données : data warehouse et l'intégration virtuelle. Dans data warehouse, les données personnelles de chaque apprenant provenant des différentes sources de données sont chargées et matérialisées. Dans l'intégration virtuelle, les données restent dans les sources et elles sont accessibles à la demande au moment de la requête, par exemple via les API accessibles (Moreau et al., 2017). Elles seront transformées avec le schéma global par les wrappers et interrogées par les médiateurs. Nous proposons aussi un service d'intégration sémantique (Montoya et al., 2014) adapté à l'apprentissage tout au long de la vie, et qui permet d'intégrer facilement de nouvelles sources de données Web.

Les apprenants partageront leurs données via des requêtes fédérées combinant des données locales (personnelles) et des données distantes (des autres apprenants). Dans notre infrastructure, le traitement des requêtes fédérées nécessite que chaque apprenant expose un serveur, si possible peu coûteux (Verborgh et al., 2016), capable de traiter les requêtes SPARQL. Les services exécuteront des requêtes fédérées pour des finalités différentes, par exemple pour trouver *les cinq MOOC les plus suivis par les apprenants qui ont suivi le MOOC Web sémantique*. Pour trouver des données pertinentes, chaque apprenant tiendra une liste de serveurs fiables qui peuvent être organisés par un index guidé par des politiques d'usage. Les sources de données pertinentes peuvent également être découvertes d'une manière dynamique lors du traitement des requêtes grâce aux réseaux superposés sémantiques (Grall et al., 2018).

Avant de partager les données, l'apprenant associera différentes politiques d'usage (ou licences) à ses données, en fonction de ses préférences de confidentialité. Les politiques d'usage spécifient avec précision les conditions de réutilisation des données (autorisations, interdictions et obligations) et sont définies à l'aide de ODRL [2]. Par exemple, un apprenant peut décider que ses données collectées sur LinkedIn peuvent être *lues* par les autres apprenants avec l'interdiction de les *distribuer*. Les résultats de la requête seront protégés par une politique d'usage conforme à toutes les politiques d'usage des données contribuant aux résultats de la requête Moreau et al. (2018). Toutes ces politiques permettront un contrôle d'usage fort dans notre infrastructure.

3 Conclusion

Nous commençons l'expérimentation de ces services dans le contexte du projet Sedela (El Mawas et al., 2017). Des expériences préliminaires avec des étudiants du deuxième cycle en sciences de l'éducation viennent de commencer. L'objectif est d'explorer de nouvelles méthodes et de nouveaux outils pour renforcer l'autonomie, l'auto-apprentissage et le développement personnel.

1. `https://solid.mit.edu/`
2. `www.w3.org/TR/odrl-vocab/`

Remerciement Ce travail fait partie du projet multidisciplinaire SEDELA, financé par CominLabs.

Références

El Mawas, N., J.-M. Gilliot, S. Garlatti, P. Serrano-Alvarado, H. Skaf-Molli, J. Eneau, G. Lameul, J.-F. Marchandise, et H. Pentecouteau (2017). Towards a Self-Regulated Learning in a Lifelong Learning Perspective. In *CSEDU*.

Evangelos Katis, H. K. et K. Vassilakis (2018). Developing an Ontology for Curriculum & Syllabus. In *ESWC*. Poster paper.

Grall, A., P. Molli, et H. Skaf-Molli (2018). SPARQL Query Execution in Networks of Web Browsers. In *ISWC*.

Mansour, E., A. V. Sambra, S. Hawke, M. Zereba, S. Capadisli, A. Ghanem, A. Aboulnaga, et T. Berners-Lee (2016). A Demonstration of the Solid Platform for Social Web Applications. In *WWW*. Demo paper.

Montoya, G., L. D. Ibáñez, H. Skaf-Molli, P. Molli, et M.-E. Vidal (2014). SemLAV : Local-As-View Mediation for SPARQL. *TLSDKCS*.

Moreau, B., P. Serrano-Alvarado, et E. Desmontils (2018). CaLi : A Lattice-Based Model for License Classifications. In *BDA*.

Moreau, B., P. Serrano-Alvarado, E. Desmontils, et D. Thoumas (2017). Querying non-RDF Datasets using Triple Patterns. In *ISWC*. Demo paper.

Nejdl, W., B. Wolf, C. Qu, S. Decker, M. Sintek, A. Naeve, M. Nilsson, M. Palmér, et T. Risch (2002). EDUTELLA : a P2P Networking Infrastructure Based on RDF. In *WWW*.

Rezgui, K., H. Mhiri, et K. Ghédira (2017). Ontology-based e-Portfolio Modeling for Supporting Lifelong Competency Assessment and Development. *Computer Science*.

Van Harmelen, M. (2006). Personal Learning Environments. In *ICALT*.

Verborgh, R., M. Vander Sande, O. Hartig, J. Van Herwegen, L. De Vocht, B. De Meester, G. Haesendonck, et P. Colpaert (2016). Triple Pattern Fragments : A low-cost Knowledge Graph Interface for the Web. *J. Web Sem.*

Yang, S. J. et al. (2006). Context Aware Ubiquitous Learning Environments for Peer-to-Peer Collaborative Learning. *Educational Technology & Society*.

Summary

Lifelong learning is a crucial support for experienced workers in their professional development. Existing environments help learners to gain autonomy. However, they are usually restricted to collecting simple facts or evidence about personal learning. Moreover, they are not collaborative and lifelong available. In this vision paper, we show how the Semantic Web helps in building a flexible and decentralized lifelong learning infrastructure that enables trusted collaboration among learners.

LODEX : des données structurées au web sémantique

Stéphanie Gregorio, Alain Collignon
François Parmentier, Nicolas Thouvenin

Inist-CNRS, 2, Allée du Parc de Brabois, CS 10310, 54519 Vandœuvre-lès-Nancy
prenom.nom@inist.fr, https://www.inist.fr

Résumé. Lodex est un logiciel *open source* dédié à la valorisation de données structurées. Il facilite la curation et la sémantisation de données brutes pour les connecter au web de données via les normes et les standards du web sémantique. Il propose, en plus de la création automatique d'uri, la génération d'identifiants pérennes normalisés via le système des ark.

1 Introduction

Les bibliothèques produisent depuis longtemps dans leurs catalogues des données structurées et contrôlées, qu'elles exposent sur le web. Le web sémantique est présenté comme étant le web pour lequel les ordinateurs interprètent les métadonnées afin de mieux assister l'utilisateur dans sa recherche de l'information (Berners-Lee et al., 2001). L'Inist-cnrs a lancé une expérimentation visant à publier, selon les normes du web sémantique, des données extraites du fonds istex [1] (plus de 20 millions de publications scientifiques). Cette expérience a eu comme incidence le développement de lodex, outil permettant de mettre en ligne des jeux de données dans le respect des normes et standards du w3c.

Dans cet article nous présenterons brièvement l'archive istex puis nous développerons l'outil lodex qui a pour but de publier des données extraites de cette archive et ainsi faciliter l'accès et la diffusion des données acquises et produites. Cette publication est réalisée via un site dédié (`https://data.istex.fr/`) et un SPARQL endpoint (`https://data.istex.fr/sparql/`) contenant un graphe global des données istex.

2 L'archive ISTEX

Le projet istex a pour objectif de permettre à la communauté scientifique française d'accéder à une bibliothèque numérique pluridisciplinaire en texte intégral regroupant l'essentiel des publications scientifiques mondiales. Ce réservoir de publications scientifiques est bien entendu à destination des documentalistes et chercheurs ayant un besoin documentaire. C'est également une ressource unique pour tous les chercheurs gravitant autour des thématiques de la fouille de textes, du tal (Traitement Automatique de la

1. `https://www.istex.fr`

Langue), de la Recherche d'Information…La mise en ligne de ces informations en texte intégral structuré permet de développer des fonctionnalités d'extraction de connaissances basées sur les technologies de la fouille de textes.

Ces enjeux ont été un déclencheur pour proposer une documentation dynamique et interopérable du fonds ISTEX, et pour publier sous forme de jeux de données toutes les informations non présentes dans les documents. Ces derniers respectent les normes du web sémantique grâce à l'utilisation d'un outil dédié.

3 L'outil LODEX

3.1 Cadre de réflexion

Dans le contexte présenté ci-dessus nous avons identifié différents utilisateurs pouvant intervenir lors de ce processus de publication. L'internaute consulte les ressources sur la toile et peut prendre le rôle de *data consumer* lorsqu'il télécharge des informations. Le documentaliste *data manager* sélectionne, affine et publie des données en toute autonomie. L'informaticien ou le documentaliste joue la fonction d'administrateur *data administrator* du système.

Puis nous avons défini schématiquement un processus intellectuel de publication des jeux de données (Fabry et al., 2017). Pour l'établir, nous avons rapproché notre réalité de terrain avec les notions théoriques du web sémantique appliquées en milieu documentaire (Bermès et al., 2013). En particulier nous nous sommes penchés sur le caractère hétérogène des ressources et son incidence sur le protocole à mettre en œuvre.

Prenant en compte la typologie des utilisateurs ainsi que notre processus de publication, nous avons souhaité disposer d'un outil permettant de :

— publier selon des normes du web sémantique des tableaux comportant des données brutes,
— faciliter la transformation en données structurées,
— aider à aligner les données à publier avec des données similaires ou connexes,
— explorer le jeu de données publié pour valoriser et référencer chaque ressource.

Dans un environnement professionnel en pleine mutation, ayant vu naître de nouvelles activités dans les bibliothèques (ou centres de documentation), la curation, la modélisation, la normalisation, le modèle RDF sont au cœur des préoccupations des *data managers*. Ceci a eu pour incidence l'émergence d'outils dédiés à ces activités comme par exemple LODRefine et Catmandu (Harlow, 2015). Datalift(Scharffe et al., 2012) en est un autre exemple. Le concept *élévation des données*, permettant de passer d'un fichier tabulé à un fichier RDF nous a fortement séduits. Cependant, la fonctionnalité d'exposition des données sur le web était peu satisfaisante. Plus près de nos préoccupations, le logiciel CubicWeb dédié aux techniques du web sémantique est utilisé dans le développement de l'application `data.bnf.fr` (Le Bœuf, 2013). Le logiciel CubicWeb, présente de nombreuses fonctionnalités pouvant nous être utiles, cependant l'usage de ce *framework* nécessite l'appui technique de la société Logilab, par conséquent, nous nous sommes orientés vers le développement d'une solution logicielle libre appelée LODEX.

Par rapport aux outils similaires, cet outil se concentre sur trois priorités : masquer la complexité des triplets au format RDF, donner envie de structurer son information en augmentant les données (visualisation, interconnexion, *etc.*) et faciliter la mise à jour ou l'ajout d'information sans refaire un long processus de publication. LODEX a été développé avec des technologies JavaScript. C'est un logiciel libre dont le code source est accessible sur GitHub [2] et sous licence CeCILL.

2. `https://github.com/Inist-CNRS/lodex`

3.2 Le back office

Son *back office* permet de réaliser toutes les fonctionnalités nécessaires au traitement ou *stylage* d'un jeu de données.

Après avoir importé un jeu de données dans un des formats acceptés (`.csv`, `.tsv`, `.xml`, `.json`, …), l'outil propose six grandes étapes permettant le processus de publication :

1. Informations générales.
2. Comment la valeur est créée.
3. Transformations appliquées à cette valeur.
4. Sémantiques.
5. Comment et où elle est affichée.
6. Recherche.

Nous allons détailler les singularités de LODEX, sans nous attarder sur l'ensemble du processus qui sera développé lors de la démonstration du logiciel.

Suite à l'import d'un fichier, l'outil génère automatiquement un URI (*Uniform Resource Identifier*), identifiant requis pour le web sémantique. Par défaut, LODEX crée un `uid://` (*Unique Identifier*). Si votre organisation a opté pour le système d'identification ARK [3], l'URI se génère automatiquement en fonction de la présence des paramètres `naan` et `subpublisher` dans le fichier de configuration.

Une attention particulière a été portée à la fonctionnalité « Transformations appliquées à cette valeur » car elle donne la possibilité au *data manager* de réaliser une curation automatisée de ses données. L'outil LODEX propose différents *transformers* permettant de standardiser le contenu du jeu de données. Par exemple, LODEX permet de transformer la valeur du champ en un booléen, de remplacer une chaîne de caractères par une autre ou bien encore d'ajouter une chaîne de caractères à la fin de la valeur du champ…

L'étape 4 « Sémantiques », permet de renseigner la propriété ou prédicat des triplets (un triplet est composé de trois parties : sujet - prédicat - objet). La saisie y est facilitée par autocomplétion avec les différentes ontologies présentes dans le *Linked Open Vocabularies* (LOV). LODEX exporte les structures nécessitant des nœuds blancs en leur créant des identifiants uniques. Nous avons identifié deux cas :

1. Annoter un autre champ : par exemple pour préciser la source d'une définition.
2. Composer ce champ : au sens du web sémantique, composer ce champ à partir de plusieurs champs. Par exemple, une adresse est composée d'un nom de rue, d'une ville, d'un pays.

Après curation, *sémantisation*, le jeu de données est publié via le *front office* (dans notre cas `https://data.istex.fr/`). Différents exports aux formats du web sémantique sont possibles (Turtle pour sa lisibilité ; N-Quads et N-Triple pour leur simplicité et JSON pour son application courante dans le web). Ces exports permettent d'alimenter un *triplestore* (`https://data.istex.fr/sparql/`).

Une documentation permettant la prise en main de l'outil ainsi que son utilisation est accessible à l'adresse suivante `https://user-doc.lodex.inist.fr/`. Des tutoriels viendront la compléter.

3. `http://www.bnf.fr/fr/professionnels/issn_isbn_autres_numeros/a.ark.html`

4 Conclusion

L'objectif principal de notre approche est de mettre à disposition un outil intuitif afin de valoriser un jeu de données via le web de données ou *Linked Open Data*. L'outil LODEX qui présente la caractéristique de publier des tableaux bruts selon des normes du web sémantique révèle les particularités suivantes :

— faciliter la transformation de données structurées en données sémantisées,
— aider à aligner les données à publier avec des données similaires ou connexes,
— exposer le jeu de données pour valoriser et référencer chaque ressource.

Dans le nouveau paradigme de la science ouverte et plus particulièrement celui des données ouvertes, l'outil LODEX peut être un excellent allié afin de publier des données selon les principes FAIR [4].

Références

Bermès, E., A. Isaac, et G. Poupeau (2013). *Le Web Sémantique en bibliothèque*. Electre-Ed. du Cercle de la Librairie, Paris.

Berners-Lee, T., J. Hendler, et O. Lassila (2001). The semantic web. *Scientific American*, p. 29–37.

Fabry, C., C. Roussel, C. A., M. E., P. F., et T. N. (2017). Publier des données liées et ouvertes en sept étapes. *I2D - Information, données & documents 54*, p. 12–14.

Harlow, C. (2015). Data munging tools in preparation for RDF: Catmandu and LODRefine. *Code{4}lib Journal 30*, p. 1–12.

Le Bœuf, P. (2013). Customized OPACs on Semantic Web: the OpenCat prototype. In *IFLA Satellite Meeting*, Singapore.

Scharffe, F., L. Bihanic, G. Képéklian, G. Atemezing, R. Troncy, F. Cotton, F. Gandon, S. Villata, J. Euzenat, Z. Fan, B. Bucher, F. Hamdi, P.-Y. Vandenbussche, et B. Vatant (2012). Enabling linked data publication with the datalift platform. In *Proc. AAAI workshop on semantic cities*.

Summary

LODEX is an open source software dedicated to the valuation of structured data. It facilitates the curation and semantisation of raw data to connect them to the web of data via standards of the semantic web. It offers, in addition to the automatic creation of URIs, the generation of standardized perennial identifiers via the ARK system.

4. Findable, Accessible, Interoperable and Reusable

Enrichissement de Données RDF Integrées à la Volée

Benjamin Moreau[*,**] Emmanuel Desmontils[*],
Patricia Serrano-Alvarado[*]

[*]GDD-LS2N – Nantes University, France
{Name.LastName@}univ-nantes.fr,
https://www.ls2n.fr
[**]OpenDataSoft
{Name.LastName}@opendatasoft.com
https://www.opendatasoft.com

Résumé. Les règles d'inférence sous-jacente à une ontologie sont des atouts majeurs du web des données. Cependant, mettre en place l'inférence est très couteux en temps d'exécution, stockage et maintenance. Certains producteurs de données décident de ne pas matérialiser leurs données en RDF ce qui rend compliqué l'enrichissement des données. Dans cette démonstration, nous présentons une approche pour bénéficier de l'enrichissement sémantique pendant l'exécution de requêtes SPARQL sur des données non-RDF accessibles à travers une API comme Twitter, Github et LinkedIn.

1 Introduction et Motivation

Les règles d'inférence sous-jacente à une ontologie sont des atouts majeurs du web des données. Concrètement, des triplets RDF implicites peuvent être déduits à partir des triplets explicites en exploitant les ontologies utilisées pour décrire les données. Cet enrichissement du jeu de données permet d'évaluer des nouvelles requêtes. Pour être exploitables, les triplets implicites peuvent être ajoutés a priori au graphe RDF ou être retournés pendant l'exécution de la requête.

Des travaux comme (Weaver et Hendler, 2009) et (Subercaze et al., 2016) proposent de matérialiser les triplets implicites. Cette approche augmente fortement le volume d'un jeux de données. D'autres travaux comme (Pérez-Urbina et al., 2009) et (Rosati et Almatelli, 2010) utilisent des techniques de réécriture de requêtes pour tenir compte des triplets implicites sans les matérialiser. Avec cette approche, le coût de stockage est limité, mais le temps d'exécution des requêtes est augmenté.

Pour diminuer les coûts de stockage et de maintenance, certains fournisseurs proposent un accès à leurs données au format RDF en utilisant des approches d'intégration virtuelle. Ces approches utilisent la réécriture de requêtes pour évaluer des requêtes SPARQL sur des données non-RDF. Un *mapping*, décrit par un langage de mapping RDF (Dimou et al., 2014; Lefrançois et al., 2017), est utilisé pour représenter les correspondances entre les données non-RDF et RDF. Dans ce contexte, (Michel et al., 2016) propose l'exécution de requêtes SPARQL sur des documents MongoDB. Dans notre travail, nous visons l'intégration au web des données

de n'importe quelle source de données non-RDF accessible à travers une API (Moreau et al., 2017). L'approche, nommée ODMTP (On-Demand Mapping using Triple Patterns), utilise TPF (Triple Pattern Fragments) (Verborgh et al., 2016) dont l'interface est simple. Seuls des triplets requêtes sont évaluées sur le serveur.

Supporter la déduction de triplets avec les approches d'intégration RDF à la volée peut s'avérer difficile, car les triplets implicites ne peuvent pas être déduits sans les triplets explicites. De plus, les techniques de réécriture de requêtes augmentent le temps d'exécution des requêtes, lequel est déjà détérioré par l'intégration des données RDF à la volée. Nous proposons donc une approche simple qui consiste à étendre le mapping RDF pour permettre l'exécution d'un domaine plus large de requêtes SPARQL sur des données non-RDF. Notre objectif est de limiter les surcoûts en terme de stockage et de temps d'exécution des requêtes SPARQL.

Considérons un exemple d'intégration au web des données des données de Twitter. L'API Twitter permet d'accéder au contenu d'un tweet, aux liens présents dans le tweet, aux hashtags et aux différentes méta-données. Le Listing 4 représente un extrait de l'ontologie qui peut être utilisée pour décrire ces données en RDF. Nous utilisons cette ontologie pour décrire, en RML, la transformation des données Twitter en RDF. Le Listing 1 représente un extrait de ce mapping.

```
...
<#Tweets>
  rr:subjectMap [ rr:template "https://twitter.com/statuses/{$.id_str}";
    rr:class schema:SocialMediaPosting; ];
  rr:predicateObjectMap [ rr:predicate it:includedHashtag;
    rr:objectMap [xrr:reference "$.entities.hashtags";]; ]
```

Listing 1 – *Extrait d'un mapping.*

Ce mapping permet à ODMTP d'interroger l'API Twitter pour répondre à des requêtes SPARQL comme celle du Listing 2 où on recherche les tweets (*schema:SocialMediaPosting*) qui correspondent à un Hashtag en particulier. Mais l'ontologie nous indique qu'un tweet est aussi un *schema:Article*. Afin d'élargir le domaine des requêtes possibles, nous utilisons le mapping et l'ontologie pour déduire les triplets implicites.

Le résultat de cette inférence nous donne le mapping étendu du Listing 5. Ce mapping permet d'exécuter des requêtes plus générales comme celle du Listing 3 ce que le mapping du Listing 1 ne peut pas faire. En effet, ce dernier ne décrit pas les triplets correspondants aux triplets de la requête.

```
SELECT ?tweet
WHERE { ?tweet a schema:SocialMediaPosting .
        ?tweet it:includedHashtag "SemanticWeb" }
```

Listing 2 – *Une requête SPARQL exécutable sur les triplets explicites.*

```
SELECT ?article
WHERE { ?article a schema:Article .
        ?article it:includedHashtag "SemanticWeb" }
```

Listing 3 – *Une requête SPARQL exécutable sur les triplets implicites.*

```
...
schema:SocialMediaPosting rdfs:subClassOf schema:Article;
            rdfs:subClassOf schema:CreativeWork;
            rdfs:subClassOf schema:Thing.
it:includedHashtag rdfs:domain schema:SocialMediaPosting;
            rdfs:range xsd:String.
```

Listing 4 – *Extrait de l'ontologie.*

```
...
<#Tweets>
  rr:subjectMap [ rr:template "https://twitter.com..."
    rr:class schema:SocialMediaPosting;
    rr:class schema:Article;
    rr:class schema:CreativeWork;
    rr:class schema:Thing; ];
  rr:predicateObjectMap [rr:predicate it:includedHashtag;
    rr:objectMap [xrr:reference "$.entities.hashtags";];]
```

Listing 5 – *Extrait d'un mapping étendu.*

Dans cette démonstration, nous montrons l'efficacité de notre approche en proposant une extension de ODMTP supportant l'inférence. Cette extension étend le mapping RDF pour permettre l'exécution de requêtes SPARQL sur les triplets explicites et implicites sans avoir besoin de les matérialiser.

2 Intégration à la demande avec mapping étendu

La Figure 1 illustre le mécanisme de raisonnement sur le mapping intégré à ODMTP. Au déploiement de ODMTP, le mapping est étendu en utilisant un raisonneur sémantique. Le raisonneur sémantique déduit les triplets implicites à partir du mapping, de l'ontologie utilisée pour décrire le mapping et des règles définies par RDFS ou OWL LD (Glimm et al., 2012).

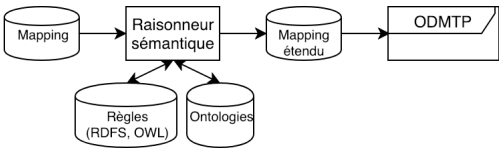


FIG. 1 – *ODMTP muni d'un raisonneur sémantique.*

Le mapping étant étendu au préalable, le temps d'exécution de la requête n'est pas détérioré par rapport à l'approche ODMTP classique. De plus, le temps de stockage est limité, car le nombre de triplets ajoutés dans le mapping est petit par rapport au nombre de triplets implicites d'un jeu de données RDF.

Cependant, toutes les règles de déduction ne sont pas applicables sur un mapping. L'extension d'un mapping RDF dépend de la catégorie de l'ontologie utilisée : RDFS, OWL Lite, OWL LD, etc. A chaque catégorie correspond des règles différentes. Par exemple, parmi les règles associées à RDFS [1], la règle *rdfs9* permet d'étendre un mapping. En effet, l'extension à partir des règles n'est possible que pour les règles s'appliquant aux concepts et aux propriétés manipulés dans le mapping. A l'inverse, les règles prenant en compte seulement les individus ne sont pas applicables car les individus ne sont pas matérialisés. Comme par exemple, dans la règle de transitivité sur les propriétés *prp-trp* de l'ontologie OWL LD. Une liste des règles implémentées est disponible sur le dépot Github [2].

3 Démonstration

Dans cette démonstration, nous utilisons une implémentation de ODMTP pour Twitter, Github et LinkedIn [3]. Les participants exécuteront des requêtes SPARQL sur les API de Twitter, Github et LinkedIn. Ils constateront que ODMTP supportant l'inférence peut exécuter plus de requêtes, que le temps d'exécution de la requête n'est pas détérioré et que le coût supplémentaire pour stocker le mapping étendu est minime.

1. https://www.w3.org/TR/rdf11-mt/#rdfs-entailment
2. https://github.com/benjimor/odmtp-tpf#supported-rules
3. https://github.com/benjimor/odmtp-tpf

La limite de cette approche est l'impossibilité d'appliquer certaines règles d'inférences se reportant aux individus. Il serait possible de les appliquer au moment de la matérialisation des données RDF. Cependant cela détériorerait le temps d'exécution des requêtes.

Références

Dimou, A., M. Vander Sande, P. Colpaert, R. Verborgh, E. Mannens, et R. Van de Walle (2014). RML: A Generic Language for Integrated RDF Mappings of Heterogeneous Data. In *LDOW in World Wide Web Conf (WWWC)*.

Glimm, B., A. Hogan, M. Krötzsch, et A. Polleres (2012). OWL: Yet to Arrive on the Web of Data ? In *LDOW in World Wide Web Conf (WWWC)*.

Lefrançois, M., A. Zimmermann, et N. Bakerally (2017). A SPARQL Extension for Generating RDF from Heterogeneous Formats. In *European Semantic Web Conf (ESWC)*.

Michel, F., C. Faron-Zucker, et J. Montagnat (2016). A Mapping-based Method to Query MongoDB Documents with SPARQL. In *Database and Expert Systems Applications (DEXA)*.

Moreau, B., P. Serrano-Alvarado, E. Desmontils, et D. Thoumas (2017). Querying non-RDF Datasets using Triple Patterns. *Demo in International Semantic Web Conf (ISWC)*.

Pérez-Urbina, H., I. Horrocks, et B. Motik (2009). Efficient Query Answering for OWL 2. In *International Semantic Web Conf (ISWC)*.

Rosati, R. et A. Almatelli (2010). Improving Query Answering over DL-Lite Ontologies. In *Principles of Knowledge Representation and Reasoning (KR)*.

Subercaze, J., C. Gravier, J. Chevalier, et F. Laforest (2016). Inferray: Fast In-memory RDF Inference. *VLDB Endowment*.

Verborgh, R., M. Vander Sande, O. Hartig, J. Van Herwegen, L. De Vocht, B. De Meester, G. Haesendonck, et P. Colpaert (2016). Triple Pattern Fragments: A Low-cost Knowledge Graph Interface for the Web. *Journal of Web Semantics. 37*.

Weaver, J. et J. A. Hendler (2009). Parallel Materialization of the Finite RDFS Closure for Hundreds of Millions of Triples. In *International Semantic Web Conf (ISWC)*.

Summary

Inference rules using ontologies is one of the major assets of the web of data. However, implementing inference is very expensive in terms of execution time, storage and maintenance. Some data producers decide not to materialize their data in RDF which makes it difficult to enrich the data. In this demonstration, we present an approach to benefit from the semantic enrichment during the execution of SPARQL queries on non-RDF data accessible through an API like Twitter, Github and LinkedIn.

Index