Vers une exploitation efficace de grandes bases de connaissances par des graphes de contexte

Nada Mimouni, Jean-Claude Moissinac

Vers une exploitation efficace de grandes bases de connaissances par des graphes de contexte

Nada Mimouni, Jean-Claude Moissinac

TÉLÉCOM PARISTECH, LTCI Paris, France
nada.mimouni@telecom-paristech.fr, jean-claude.moissinac@telecom-paristech.fr

Résumé : Un problème lié à l'exploitation de graphe de connaissances, en particulier lors de traitements avec des méthodes d'apprentissage automatique, est le passage à l'échelle. Nous proposons ici une méthode pour réduire significativement la taille des graphes utilisés pour se focaliser sur une partie utile dans un contexte d'usage donné. Nous définissons ainsi la notion de graphe de contexte comme un extrait d'une ou plusieurs bases de connaissances généralistes (tels que DBpedia, Wikidata, Yago) qui contient l'ensemble d'informations pertinentes pour un domaine spécifique tout en préservant les propriétés du graphe d'origine. Nous validons l'approche sur un extrait de DBpedia pour des entités en lien avec le projet Data&Musée et le jeu de référence KORE selon deux aspects : la couverture du graphe de contexte et la préservation de la similarité entre ses entités.
Mots-clés : Base de connaissances, Graphe de contexte, Similarité, DBPedia, Base Joconde

1 Introduction

Les développements du web sémantique et des données liées (le Linked Open Data ou LOD) au cours de la dernière décennie ont permis la publication et le liage de plusieurs données structurées sur le Web. Le *LOD cloud* [1] est passé de 12 à 1378 ensembles de données et de 500 millions à plus de 130 milliards de triplets RDF entre 2007 et 2019. Ces données concernent plusieurs domaines comme la culture, les sciences du vivant, les données de gouvernements ou les données géographiques. Ce développement a démontré l'intérêt de relier un jeu de données d'un domaine applicatif restreint avec un ensemble de données externes afin d'en tirer une meilleure compréhension et exploitation.

Dans le cadre du projet Data&Musée, un projet exploratoire visant à améliorer les systèmes d'information d'institutions culturelles, nous faisons l'hypothèse que la promotion du patrimoine culturel peut bénéficier des techniques récentes de représentation et d'exploration des connaissances. Pour cela, nous avons besoin d'enrichir des ensembles de données relatives au domaine du projet et d'en assurer l'interopérabilité pour parvenir à accroître la visibilité et l'accessibilité par un plus large public. Une des conséquences directes est l'amélioration des conditions financières des institutions culturelles afin d'assurer une meilleure préservation de ce patrimoine.

Une approche reconnue pour assurer ce type d'exploitation de données est l'utilisation des technologies du web sémantique, qui ont montré leur puissance pour le développement des connaissances dans divers domaines comme le tourisme (Soualah Alila *et al.*, 2016; Al-Ghossein *et al.*, 2018), les villes intelligentes (Consoli *et al.*, 2015; Gyrard *et al.*, 2016) ou la valorisation du patrimoine culturel (Lodi *et al.*, 2017). Ces techniques permettent d'assurer une représentation unifiée de données hétérogènes mais apparentées qui facilite le liage et l'enrichissement de ces données.

Les grandes bases de connaissances comme Yago, DBPedia, DBPedia-Fr et Wikidata sont des ressources très utiles car elles fournissent un stock de connaissances encyclopédiques semi-structurées sur les principes du LOD. Mais ces ressources posent plusieurs problèmes d'exploitation : problèmes d'accès, problèmes de performances, limites sur les usages, etc. qui sont principalement liés à leurs très grande taille. Nous nous intéressons ici aux problèmes d'échelle et de performance qui peuvent se poser lorsqu'on veut exploiter des liens vers ces grands graphes de connaissances.

1. https://lod-cloud.net/

Dans cet article nous proposons une représentation alternative simplifiée, fidèle et plus accessible d'une base de connaissance, a fortiori étendue, au travers d'un graphe de contexte. L'algorithme d'extraction que nous proposons construit le graphe de contexte pour un domaine défini par un ensemble d'entités représentatives. Le graphe construit se caractérise par la préservation, dans le cadre de ce domaine, des propriétés du graphe d'origine tout en limitant les problèmes de performance et de passage à l'échelle.

Nous évaluons les propriétés du graphe extrait selon deux critères : sa couverture du domaine et son impact sur les résultats d'un ensemble de mesures de similarité entre les entités extraites. En effet, évaluer la similarité entre les ressources est crucial pour plusieurs applications guidées par les données, telles que la découverte de liens, le *clustering* ou le classement.

Nous avons effectué une série de tests pour valider notre méthode sur des données d'institutions culturelles issues du projet Data&Musée. Les résultats montrent que l'utilisation de graphes de contexte rend l'exploitation de grandes bases de connaissances plus maniable et efficace tout en préservant des propriétés du graphe initial.

Dans ce qui suit, la section 2 présente un état de l'art sur l'utilisation des contextes avec des bases de connaissances. La section 3 rappelle les notions de base sur les graphes sémantiques, donne les définitions que nous utilisons dans notre approche et décrit le processus de construction du graphe de contexte. La section 4 fait une revue des mesures de similarité sur des graphes de connaissances et présente notre mesure définie pour la validation d'un graphe de contexte. Les sections 5 et 6 décrivent les expérimentations et les tests de validation effectués respectivement sur les données de Paris Musées et sur le jeu de données de référence KORE. La conclusion et les perspectives sont données dans la section 7.

2 Travaux connexes

La notion de contexte a été utilisée dans plusieurs travaux basés sur le web sémantique pour différentes applications comme le calcul de similarités entre entités ou entre documents, la découverte des liens d'identités pour le liage des données sur le LOD ou la transformation vectorielle des graphes pour application à des méthodes d'apprentissage automatique (Shen *et al.*, 2015; Raad *et al.*, 2017; Beek *et al.*, 2016; Benedetti *et al.*, 2019; Shi *et al.*, 2017; Luo *et al.*, 2015). Ces approches utilisent un extrait des bases de connaissances, appelé contexte, qui le considèrent comme une partie du grand graphe porteuse de sémantique pour une ou plusieurs ressources.

Sémantique du contexte

Dans (Hulpus *et al.*, 2013), les auteurs décrivent un concept d'intérêt (*concept of interest*) C dans DBpedia par un graphe appelé graphe de sens (*sense graph*) ayant comme racine C. Ils proposent une solution au problème d'étiquetage automatique de thèmes (*automatic topic labelling*) utilisant DBpedia. Les thèmes sont extrait par une méthode de *probabilistic topic modelling* (comme LDA). Pour chaque concept C_i associé à un terme d'un thème identifié, ils extraient un *sense graph* G_i en interrogeant tous les nœuds situés à au plus deux sauts (2-hop) de C_i en prenant en compte récursivement tous les liens de type `skos:broader`, `skos:broaderOf`, `rdfs:sub-ClassOf`, `rdfs:type` et `dcterms:subject`. Les graphes G_i sont ensuite fusionnés pour obtenir le graphe de thème (*topic graph*) G. Dans la même direction, les auteurs dans (Raad *et al.*, 2017; Beek *et al.*, 2016) montrent que l'utilisation de contextes permet de mieux décrire les entités pour les lier via des liens d'identité de type `owl:sameAs`. Un lien d'identité est valide dans un contexte, correspondant à un sous-ensemble de propriétés, si deux instances i_1 et i_2 ont les mêmes valeurs de ces propriétés. Ils postulent que deux instances similaires dans un contexte peuvent ne pas l'être dans un autre avec un sous-ensemble différent de propriétés. Ils montrent ainsi l'importance de la prise en compte du contexte pour le calcul de similarité.

La notion d'extrait d'une base de connaissance a été également étudiée dans des domaines plus spécifiques comme l'IoT ou l'environnement pour réduire la complexité de manipulation des données dans ces domaines. (Gyrard *et al.*, 2016) propose le système LOV4IoT pour la

construction d'applications sémantiques de web d'objets utilisant des ontologies du domaine afin de réduire les espaces de recherche et faciliter l'interrogation. Les résultats dans (Wannous *et al.*, 2017) montrent l'impact positif des optimisations, telles que des contraintes de domaine et des raffinements de voisinage, sur la réduction de la complexité du mécanisme d'inférence sur des bases de connaissance de comportements d'animaux. Ces optimisations ont permis de réduire à moitié le temps de calcul et d'améliorer ainsi le passage à l'échelle.

Similarité dans un contexte

La plupart des méthodes qui comparent des ressources, par exemple en terme de similarité, dans le web sémantique se basent sur un ensemble pré-sélectionné de triplets (Colucci *et al.*, 2016). Pour leur méthode de définition et de calcul du LCS (*Least Common Subsumer* : l'ancêtre taxonomique le plus spécifique qui subsume deux ressources) dans des graphes RDF, les auteurs montrent qu'il est important d'expliciter le sous-graphe du web sémantique qui sert de contexte au calcul de LCS pour une ressource r. Le contexte de r, appelé *rooted r-graph*, est constitué d'un ensemble T_r de triplets tel que toutes les ressources dans T_r sont connectées à r par un chemin dans le graphe RDF. Dans (Cheniki *et al.*, 2016), une mesure de similarité entre entités basée sur le LOD est définie sur un contexte (voisins à une profondeur N) extrait à partir de l'ensemble des données disponibles. Pour le calcul de similarité entre documents, les auteurs dans (Benedetti *et al.*, 2019) définissent le contexte sémantique d'analyse extrait d'une base de connaissances comme DBpedia. A partir de ce contexte, ils créent un vecteur sémantique de contexte qui permet de surpasser les méthodes classiques de similarité inter-documents. Dans (Bhatt *et al.*, 2019), les auteurs décrivent un algorithme de détection et de caractérisation de communautés basé sur les graphes de connaissances. Ils abordent le problème de trouver le contexte qui résume le mieux les nœuds des communautés. L'algorithme utilise une mesure de similarité qui intègre les attributs des nœuds décrits dans des graphes de connaissances hiérarchiques (HKG) spécifiques à un domaine. Ces graphes fournissent des informations pertinentes pour un groupe d'objets du monde réel.

Apprentissage sur contexte

L'utilisation des graphes de connaissances avec des méthodes d'apprentissage automatique a été principalement favorisée par le développement de techniques de transformation vectorielle de graphes (*graph embedding*). Cette transformation préserve les propriétés pertinentes du graphe d'origine comme la topologie (proximité entre voisins) ou la sémantique. Dans ce cadre, (Shi *et al.*, 2017) et (Luo *et al.*, 2015) proposent un *embedding* de graphe de connaissances qui crée des vecteurs mieux représentatifs des entités. L'approche tient compte des contextes explicites (liens entrants et sortants et chemins entre paires d'entités) et implicites (motifs de connectivité contextuelle) entre entités non connectées dans ce graphe. Un contexte implicite est construit à partir de l'hypothèse que les entités connectées à un même noeud sont généralement implicitement liées les unes aux autres, même si elles ne sont pas directement liées dans le graphe.

Taille du contexte

La taille du contexte est un paramètre qui a été discuté dans plusieurs travaux. Dans leur travail sur l'étiquetage automatique de thèmes avec DBpedia (Hulpus *et al.*, 2013), les auteurs utilisent une distance de 2 sauts à partir du nœud de départ. Cette distance a été choisie suite à une série de tests sur l'expansion de nœuds qui a montré qu'à partir de 3 sauts, l'expansion produit des graphes très larges et introduit beaucoup de bruit. Pour la définition d'une mesure de similarité entre entités basée sur le LOD (Cheniki *et al.*, 2016), les auteurs se limitent à des chemins de longueur 2 pour récupérer toutes les ressources équivalentes possibles et enrichir l'espace d'instantiation d'une ressource dans le LOD.

Ces travaux mettent l'accent sur l'intérêt d'utiliser des contextes. Cependant, dans ces approches, l'intégralité de la base est considérée pour calculer le contexte à la volée au moment

de l'utilisation des ressources ce qui pose des problèmes d'accès liés à la taille de la base. Dans notre approche, nous proposons de construire *a-priori* un graphe de contexte, unique pour toutes les ressources d'un domaine, qui servira comme point d'accès optimisé pour les différents traitements dans une application donnée.

3 Graphe de contexte guidé par le domaine

3.1 Rappels sur les graphes sémantiques

Notre approche s'appuie sur des bases de connaissances décrites par une ontologie en OWL et des données représentées en RDF. Une base de connaissance correspond à un schéma conceptuel et un ensemble de faits (déclarations).

Définition. Ontologie. Une ontologie $\mathcal{O}$ correspond à la partie conceptuelle de la base (schéma) qui structure les connaissances dans un domaine donné. Elle peut être représentée par un triplet $\mathcal{O} = (C, P_r, A)$ où C est l'ensemble de classes (concepts d'un domaine), P_r est l'ensemble de propriétés de classes et A est l'ensemble d'axiomes, qui précisent des contraintes sur les propriétés d'une classe. Dans la suite, nous parlerons aussi de ***T-Box*** pour désigner cette partie conceptuelle des connaissances (voir figure 1).

Définition. Faits et graphe de connaissances. Un graphe de connaissances $\mathcal{KG}$ est défini par un ensemble de faits. Un fait est représenté par un triplet de la forme $\langle sujet, predicat, objet \rangle$. $sujet$ désigne un élément sur lequel on veut affirmer une connaissance ; $predicat$ désigne une propriété qu'on veut associer au $sujet$; $objet$ est la valeur que prend la propriété pour ce $sujet$. L'ensemble des faits constitue la ***A-Box*** d'un graphe (voir figure 1).

Cette définition fait de $\mathcal{KG}$ un graphe étiqueté orienté où $\mathcal{V}$ est l'ensemble de nœuds (sommets) et $\mathcal{E}$ est l'ensemble des liens entre deux nœuds, liens étiquetés par un prédicat (arête). Un fait décrit par $\langle sujet, predicat, objet \rangle \in \mathcal{E}$ est tel que $sujet, objet \in \mathcal{V}$ et $predicat \in \mathcal{P}$, un ensemble de prédicats, par exemple choisi dans un ensemble de propriétés P_r définis dans une ontologie.
$\mathcal{V}$ est l'union de trois ensembles disjoints :

$$\mathcal{V} = \{v \mid v \in \mathcal{U} \cup \mathcal{B} \cup \mathcal{L}\}$$

où $\mathcal{U}$ = ensemble des URIs (identifiants uniques de ressources),
$\mathcal{B}$ = ensemble des *blank nodes*, des sommets qui ont un rôle technique pour regrouper des propriétés sans les associer à une URI,
$\mathcal{L}$ = ensemble des valeurs littérales ; il s'agit de valeurs typées : chaînes de caractères, valeurs numériques, dates, etc.
Un prédicat lie deux URIs ou *blank nodes* ou une URI ou un *blank node* avec un littéral.

$$\mathcal{E} = \{(v_1, e, v_2) \mid v_1 \in \mathcal{U} \cup \mathcal{B}, v_2 \in \mathcal{U} \cup \mathcal{B} \cup \mathcal{L}, e \in \mathcal{P}\}$$

Dans nos expérimentations, nous utilisons la version française de DBpedia comme base de connaissance généraliste, en raison de sa large couverture et de l'abondance et la diversité des liens qu'elle contient et du fait quelle est reliée à de nombreuses autres bases. La figure 1 montre un exemple d'un sous-graphe $\mathcal{KG}$ de DBpedia.

Définition. Chemin. Un chemin $\mathcal{C}$ de longueur N, $\mathcal{C} = (e_i)_{1 \leq i \leq N}$, est une suite finie non vide de liens de $\mathcal{E}$, avec $N \in \mathbb{N}$, tel que deux liens consécutifs sont adjacents. Deux liens l_1 et l_2 sont adjacents lorsqu'ils partagent un nœud n destination pour l_1 et origine pour l_2.

Définition. Prédicat exclu. On définit un ensemble de prédicats qui seront exclus des chemins construits sur un graphe $\mathcal{KG}$; on note cet ensemble par $\overline{\mathcal{P}}$ tel que $\overline{\mathcal{P}} \subset \mathcal{P}$. L'ensemble

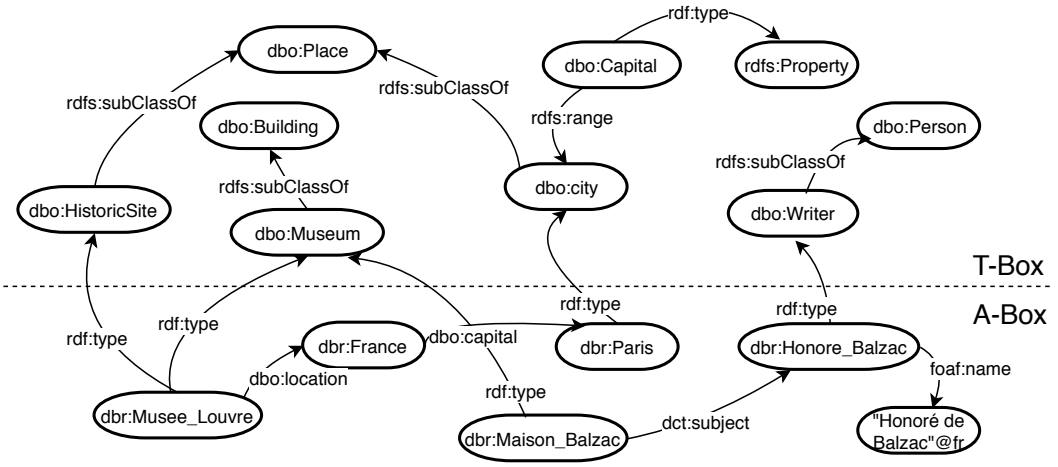

FIGURE *1 – Exemple d'un sous-graphe $\mathcal{KG}$ de DBpedia.*

des liens e étiquetés par des prédicats $p \in \overline{\mathcal{P}}$ est noté par $\overline{\mathcal{E}}$ tel que $\overline{\mathcal{E}} \subset \mathcal{E}$.

Définition. Nœud terminal. Un nœud terminal est un nœud sur lequel on impose l'arrêt de la construction d'un chemin (comme si ce nœud n'avait aucun lien sortant). Un chemin $\mathcal{C}$ construit sur un graphe $\mathcal{KG}$ s'arrête s'il rencontre un nœud terminal ; on note cet ensemble de nœuds par $\overline{\mathcal{V}}$ tel que $\overline{\mathcal{V}} \subset \mathcal{V}$.

3.2 Graphe de contexte

Étant donné un domaine d et un graphe $\mathcal{KG}$, notre objectif est d'extraire un sous-graphe de $\mathcal{KG}$ contenant toutes les informations sur le domaine d qu'on note $\mathcal{CG}(d)$. Notre méthode repose sur les parties *T-Box* et *A-Box* de $\mathcal{KG}$ en considérant des nœuds $v \in (\mathcal{U} \cup \mathcal{L})$ et des liens $e \in (\mathcal{E} \setminus \overline{\mathcal{E}})$.

3.2.1 Définition des paramètres

Pour réduire la taille du graphe de contexte, nous tenons compte d'observations faites sur $\mathcal{KG}$ et de connaissances expertes, quand elles sont disponibles, quant à l'utilité ou l'inutilité de certains nœuds et prédicats.

Plus précisément, la liste $\overline{\mathcal{V}}$ est définie à partir de $\mathcal{V}$ par l'exclusion automatique des nœuds qui appartiennent à la *T-Box* du fait de leur caractère très général (ex. dans DBpedia : `dbo:Building`, `dbo:Place` ou `owl:Thing`) et les nœuds de structuration (ex. mise en forme des pages de DBpedia, `<http://fr.dbpedia.org/resource/Modèle:P.>`).

Parallèlement, la liste $\overline{\mathcal{E}}$ est définie à partir de $\mathcal{E}$ par l'exclusion des liens étiquetés par deux types de prédicats : les prédicats considérés comme peu ou pas informatifs pour le domaine d (liste alimentée par un expert) et les prédicats de structuration de la base $\mathcal{KG}$ (ex. `<http://dbpedia.org/ontology/wikiPageRevisionID>`).

Ces nœuds et prédicats introduisent du bruit sans apporter d'information pertinente pour le domaine considéré. A titre d'exemple, dans nos expérimentations sur DBpedia, nous avons constaté 3486 nœuds construits sur `<http://fr.dbpedia.org/resource/Modele:????>` donnant lieu a 2692515 liens et 101235 liens vers `<http://www.w3.org/2004/02/skos/core#Concept>`. Dans le cadre de notre

application, nous avons construit une liste de nœuds et prédicats exclus qui peut être réutilisée pour des applications dans d'autres domaines. La liste est disponible ici [2].

3.2.2 Processus de construction

L'extraction d'un graphe de contexte $\mathcal{CG}(d)$ de dimension N est un processus récursif qui s'arrête quand la limite N est atteinte. Les étapes du processus sont :

— Identification des germes : une liste de germes $\mathcal{S}(d)$ est définie pour un domaine d'application d. Dans certains domaines cette liste est évidente comme pour le cas des musées, hôtels ou restaurants. Dans le cas général, la pratique commune consiste à se référer à un jeu de données de référence (comme IMDB pour le domaine du cinéma).
— Construction des contextes voisins : un contexte voisin $\mathcal{CV}(a)$ est généré pour toute entité a de $\mathcal{S}(d)$ et la liste des germes $\mathcal{S}(d)$ est mise à jour avec les voisins récoltés.
— Construction du graphe de contexte : le graphe de contexte $\mathcal{CG}(d)$ est construit comme l'agrégation de tous les contextes voisins $\mathcal{CV}$ des germes.

1. Identification des germes. Les germes $\mathcal{S}(d)$ sont des nœuds de $\mathcal{KG}$ qui constituent les entités de départ pour la construction du graphe de contexte $\mathcal{CG}$, $\mathcal{S}(d) = \{v | \forall v \in \mathcal{S}(d), v \in (\mathcal{V} \setminus \overline{\mathcal{V}})\}$. La liste $\mathcal{S}(d)$ est définie pour un domaine d comme l'ensemble des instances de concepts représentatifs du domaine. Par exemple, dans notre cas (projet Data& Musée), les entités de départ correspondent à liste des musées et monuments de *Paris Musées* et du *Centre des Monuments Nationaux*.

Exemple. Sur la figure 1 : $\mathcal{S}(d)$ = { `dbr:Musee_Louvre`, `dbr:Maison_Balzac` }

2. Construction des contextes voisins. Un contexte voisin $\mathcal{CV}$ d'une entité est son voisinage direct (1-hop) dans $\mathcal{KG}$. C'est la structure locale qui interagit avec l'entité et reflète divers aspects de cette entité. Plus précisément, étant donné une entité $a \in \mathcal{S}(d)$, le contexte voisin de a est définit comme suit :

$$\mathcal{CV}(a) = \mathcal{CS}(a) \cup \mathcal{CE}(a)$$

où

$$\mathcal{CS}(a) = \{(a, p, o) \mid \forall (a, p, o) \in \mathcal{E}, \forall p \in (\mathcal{P} \setminus \overline{\mathcal{P}}), o \in (\mathcal{U} \cup \mathcal{L})\}$$
$$\mathcal{CE}(a) = \{(s, p, a) \mid \forall (s, p, a) \in \mathcal{E}, \forall p \in (\mathcal{P} \setminus \overline{\mathcal{P}}), s \in \mathcal{U}\}$$

avec $\mathcal{CS}$ est un ensemble de liens sortants de a tandis que $\mathcal{CE}$ est un ensemble de liens entrants de a et s ou o est le nœud voisin de a par le lien p. Notons ici que $a \notin \overline{\mathcal{V}}$, nous ne construisons pas de contextes voisins pour les nœuds terminaux. La liste des germes $\mathcal{S}(d)$ est par la suite mise à jour avec les voisins o et s récoltés tel que $o, s \notin \overline{\mathcal{V}}$.

Exemple. Sur la figure 1 :

$\mathcal{CV}$(`dbr:Musee_Louvre`) = { (`dbr:Musee_Louvre, rdf:type, dbo:HistoricSite`), (`dbr:Musee_Louvre, rdf:type, dbo:Museum`), (`dbr:Musee_Louvre, dbo:location, dbr:France`)}

$\mathcal{CV}$(`dbr:Maison_Balzac`) = { (`dbr:Maison_Balzac, rdf:type, dbo:Museum`), (`dbr:Maison_Balzac, dct:subject, dbr:Honore_Balzac`)}

3. Construction du graphe de contexte. La construction d'un graphe de contexte est un processus récursif, sur la base de l'étape suivante :

2. `https://gitlab.com/snippets/1844328`

$\mathcal{CG}(d)$ pour un domaine d est construit comme l'agrégation de tous les contextes voisin des nœuds germes $a \in \mathcal{S}(d)$ tel que $\mathcal{S}(d) \subset (\mathcal{V} \setminus \overline{\mathcal{V}})$,

$$\mathcal{CG}(d) = \bigcup_{a \in \mathcal{S}(d)} \mathcal{CV}(a)$$

A la fin d'une étape, la liste des germes $\mathcal{S}(d)$ est mise à jour avec les voisins v récoltés tel que $v \notin \overline{\mathcal{V}}$.

Le processus est répété N fois pour un graphe de contexte de profondeur N. Comme démontré par les travaux dans la section 2, $N = 2$ est la valeur la plus intéressante pour un contexte. En effet, la taille du graphe augmente exponentiellement en fonction de la profondeur (pour des entités e qui possèdent en moyenne x voisins, à 1-hop la taille est x, à 2-hop la taille est x^2, à 3-hop la taille est x^3, etc.), aller au delà de 2 augmente significativement l'espace et introduit beaucoup de bruit. Dans le cadre de notre expérimentation, nous arriverions au niveau 3 à un graphe du même ordre de grandeur que tout DBPedia.

A la fin du processus, $\mathcal{CG}(d)$ est enrichi avec la partie *T-Box* de $\mathcal{KG}$ (ici l'ontologie DBpedia [3]) et pour tout nœud dans $\mathcal{CG}(d)$, on assure qu'un lien de type `is-a` existe avec un concept de la partie *T-Box* (si ce lien existe dans le graphe d'origine $\mathcal{KG}$). Le **cœur du contexte** est le graphe obtenu au niveau $N - 1$. Les entités ajoutées au niveau N sont la **périphérie du contexte**.

3.3 CONTEXT : Algorithme de construction d'un graphe de contexte

L'algorithme `CONTEXT` (algorithme 1) permet la construction d'un graphe de contexte $contexte$ à partir d'un graphe de connaissances $\mathcal{KG}$ pour un domaine d. Pour un ensemble d'entités représentatives d'un domaine, les germes ($germesATraiter$), `ContexteVoisin`(g) extrait un contexte voisin C_v à partir d'un graphe de connaissances $\mathcal{KG}$ pour chaque germe g. Le contexte final, $contexte$, est enrichi par C_v. Une liste de nouveaux germes, $nouveauxGermes$, est mise à jour avec les nouvelles entités récoltées après filtrage des nœuds terminaux avec la méthode `EntitésFiltrées`. La profondeur d'exploration $niveau$ est incrémenté de 1 à chaque étape jusqu'à la limite $rayon$ souhaitée. A la fin du processus, le contexte résultant $contexte$ est enrichi par les classes de l'ensemble de ces entités extraites de $\mathcal{KG}$ par les méthodes `AjoutClasses` et `Entités`.

4 Validation de graphes de contexte par mesure de similarité

4.1 Hypothèse de pertinence d'un graphe de contexte

L'utilisation d'un graphe de contexte construit à partir d'un grand graphe de connaissances permet, comme évoqué plus haut, de gagner en performance (temps de calcul, espace mémoire, etc.). Ce gain ne doit pas par ailleurs pénaliser son utilisation par les méthodes habituellement basées sur la structure et le contenu du graphe d'origine. En effet, plusieurs algorithmes utilisent les graphes de connaissances comme structure de base ou comme source d'enrichissement sémantique pour effectuer plusieurs tâches dans différents domaines tels que l'analyse de réseaux sociaux (e.g. détection de communautés) (Bhatt *et al.*, 2019), la recommandation (e-commerce, tourisme, musique) (Oramas *et al.*, 2016), etc.

La plupart de ces méthodes se base sur la notion de similarité sémantique (*semantic similarity*) ou de proximité sémantique (*semantic relatedness*) entre entités (instances de classes) pour effectuer des traitements finaux sur les données d'origine. Le calcul de ces mesures a plusieurs applications directes et pertinentes pour le traitement automatique de langue (la désambiguïsation, l'annotation sémantique, la recherche d'information, etc.), la découverte

3. `https://wiki.dbpedia.org/services-resources/ontology`

Algorithme 1 : CONTEXT

Fonction *ContexteConstructeur(*, $germesATraiter$, $rayon$, $filtre$*)*
  Entrée : Un graphe de connaissances KG
  Une profondeur $rayon$ de voisinage à atteindre
  Un ensemble d'entités qui servent de germes $germesATraiter$
  Un ensemble d'entités qui ne doivent pas servir de germes $filtre$
  Sortie : Graphe de contexte $contexte$
  $niveau \leftarrow 0$
  $contexte \leftarrow \emptyset$
  tant que $niveau$ < $rayon$ **faire**
    $nouveauxGermes \leftarrow \emptyset$
    pour chaque $g \in germesATraiter$ **faire**
      $C_v \leftarrow$ `ContexteVoisin(`KG, g`)`
      $contexte \leftarrow contexte \cup C_v$
      $nouveauxGermes \leftarrow nouveauxGermes \cup$ `EntitésFiltrées(`C_v, $filtre$`)`
    fin
    $niveau \leftarrow niveau + 1$
    $germesATraiter \leftarrow nouveauxGermes$
  fin
  $contexte \leftarrow contexte \cup$ `AjoutClasses(`KG, `Entités(`$contexte$`))`
  retourner $contexte$
fin

de liens ou le classement.

Partant des cas d'utilisation cités plus haut, nous considérons qu'une mesure de similarité est une condition nécessaire pour évaluer si l'utilisation d'un graphe de contexte peut suffire pour satisfaire les besoins en calcul des tâches relatives aux méthodes appliquées aux graphes d'origine. Nous parlons alors de pertinence d'un graphe de contexte.

Définition. Graphe de contexte pertinent. Un graphe de contexte est dit pertinent pour un domaine donné s'il préserve des propriétés du graphe d'origine pour ce domaine évalués en terme de similarité entre entités.

Hypothèse. Nous faisons l'hypothèse que notre graphe de contexte est pertinent pour un domaine si les similarités relatives de deux entités par rapport à une troisième dans le graphe d'origine, sont préservées dans le graphe de contexte.

Dans les graphes de connaissances, la sémantique qui décrit les ressources est codée selon différents aspects comme par exemple les voisins ou la hiérarchie de classes. Une grande majorité des mesures de similarité existantes considèrent les aspects de manière isolée ce qui, dans notre cas, ne permet pas de couvrir toutes les propriétés de ces ressources. Nous définissons dan ce qui suit une mesure de similarité composée (section 4.3) qui prend en compte l'aspect structurel (les liens de hiérarchie taxonomique) et sémantique (l'ensemble des prédicats) décrivant une entité. Ces mesures seront utilisées dans les sections 5 et 6 afin d'évaluer la pertinence d'un graphe de contexte.

4.2 Revue des mesures de similarité basées sur des graphes de connaissances

Dans la littérature, nous distinguons trois grandes familles de mesures de similarité qui se basent sur les ontologies, la partie T-Box d'une base de connaissance (pour une revue dé-

taillée voir (Sánchez *et al.*, 2012; Harispe *et al.*, 2014)).

Mesures basées sur les liens (*edge-counting*). Ces mesures utilisent le nombre de liens séparant les nœuds (relation transversale) comme critère de similarité. La mesure la plus directe est celle définie par (Rada *et al.*, 1989) qui calcule le chemin C le plus court entre deux entités dans un graphe $\mathcal{KG}$ en suivant les liens `is-a` : $dis(a, b) = min_i(N_i)$, N_i longueur de $C_i \in \mathcal{C}$, $\mathcal{C}$ est l'ensemble des chemins entre a et b. Plusieurs améliorations de cette mesure de base ont été proposées pour prendre en compte le profondeur des nœuds dans la hiérarchie (relations hiérarchiques) (Wu & Palmer, 1994; Li *et al.*, 2003; Paul *et al.*, 2016). La plupart de ces mesures se base sur le calcul du LCS qui a montré un intérêt pour des tâches d'extraction d'information du web de données : désambiguïsation et liage d'entités, détection de communautés de données RDF ou extraction automatique de propriétés partagées entre ressources (Colucci *et al.*, 2016).

Mesures basées sur les propriétés (*feature-based*). Ces mesures complètent les méthodes basées sur le chemin en considérant le degré de chevauchement entre les propriétés des entités comparées. La similarité est calculée comme fonction des propriétés en commun et des différences entre les entités. La mesure de base adoptée est celle définie par *Tversky* (Tversky, 1977) : $sim_t(a, b) = \alpha.f(P(a) \cap P(b)) - \beta.f(P(a) \setminus P(b)) - \gamma.f(P(b) \setminus P(a))$, avec $P(a)$ et $P(b)$ respectivement les propriétés des entités a et b. Plusieurs méthodes ont été proposées (Rodriguez & Egenhofer, 2003; Petrakis *et al.*, 2006) selon le choix de la nature des propriétés (e.g. dans WordNet, les *synsets* et *glosses* ont été utilisées) et le calcul des paramètres de pondération α, β et γ.

Mesures basées sur le contenu (*information content*). Ces mesures reposent sur des corpus de textes pour calculer des probabilités sur l'occurrence des mots et des thésaurus (e.g. WordNet) pour le calcul des hyponymes des concepts (Sánchez & Batet, 2011; Traverso-Ribón & Vidal, 2015), un aspect qui sort du cadre de cette étude.

Les mesures de similarité basées purement sur le graphe de connaissances (liens, propriétés) se caractérisent par leur simplicité et efficacité. Elles exploitent le réseau de sommets et de liens étiquetés, contrairement aux méthodes basées sur le contenu qui nécessitent des sources de données externes. Cependant, ces mesures considèrent les aspects des ressources en isolation et représentent moins l'intégralité de l'information autour de ces nœuds. Selon (Traverso *et al.*, 2016), une mesure de similarité qui combine différents aspects d'une entité donne une meilleure corrélation avec les valeurs de référence. Ces aspects sont les voisins, la hiérarchie et le degré d'un nœud ou sa spécificité. La définition de cette mesure se rapproche de notre objectif quant à la représentation des différents aspects des ressources dans un graphe. Toutefois, elle n'est pas utilisable dans notre cas comme, par construction du graphe de contexte, l'aspect degré d'un nœud (nombre de liens incidents) n'est pas conservé (notamment pour les nœuds terminaux et appartenant à la *T-Box*).

La validation du graphe dans notre approche se repose ainsi sur deux aspects en combinant deux types de mesures :

— les mesures basées sur les liens comme elles permettent de couvrir l'aspect structurel dans un graphe (structure locale d'un nœud) ;
— les mesures basées sur les propriétés comme elles exploitent plus de connaissances sémantiques en évaluant à la fois les points communs et les différences.

4.3 Une mesure de similarité pour la validation de graphes de contexte

Nous présentons dans cette section une nouvelle mesure de similarité qui repose sur les liens taxonomiques et les propriétés des entités dans un graphe de connaissances afin de valider l'hypothèse de la section 4.1. Cette mesure se compose de deux parties. La première partie sert à valider la structure du graphe en suivant les liens taxonomique (de type `is-a`) pour comparer deux entités. Nous utilisons pour ceci la mesure de *Wu et Palmer* (Wu & Palmer, 1994).

Définition. Similarité liens. Soient a et b deux entités dans le graphe, N_1 et N_2 respectivement le nombre de liens `is-a` à partir de a et b jusqu'à leur LCS, N_3 est le nombre de liens `is-a` du LCS à la racine du graphe (racine de la A-Box). La similarité $sim_l(a, b)$ entre a et b est calculée comme suit :

$$sim_l(a, b) = \frac{2 \times N_3}{N_1 + N_2 + 2 \times N_3} \qquad (1)$$

La deuxième partie est basée sur les propriétés et suit le principe proposé dans le modèle de *Tversky* (décrit dans la section 4.2) qui considère que la similarité entre deux entités est une fonction de leurs propriétés communes et distinctives. Nous considérons en deuxième lieu l'ensemble des couples (propriété,valeur). La même définition est utilisée pour les deux mesures avec l'ensemble des propriétés ou des couples (propriété,valeur).

Définition. Similarité propriétés. Soient a et b deux entités dans le graphe, $\mathcal{P}_a = \{e : (a, v) \mid v \in \mathcal{U} \cup \mathcal{L}\}$ et $\mathcal{P}_b = \{e : (b, v) \mid v \in \mathcal{U} \cup \mathcal{L}\}$ respectivement l'ensemble des propriétés de a et b. La similarité $sim_p(a, b)$ entre a et b est calculée en fonction de la cardinalité de leurs propriétés comme suit :

$$sim_p(a, b) = \frac{|\mathcal{P}_a \cap \mathcal{P}_b|}{|\mathcal{P}_a \setminus \mathcal{P}_b| + |\mathcal{P}_b \setminus \mathcal{P}_a| + |\mathcal{P}_a \cap \mathcal{P}_b|} \qquad (2)$$

Définition. Similarité propriété-valeur. Soient $\mathcal{KG}$ un graphe de connaissances et a et b deux entités dans $\mathcal{KG}$, $\Sigma_a = \{(e, v) \mid e \in \mathcal{P}_a, v \in \mathcal{U} \cup \mathcal{L}\}$ et $\Sigma_b = \{(e, v) \mid e \in \mathcal{P}_b, v \in \mathcal{U} \cup \mathcal{L}\}$ respectivement l'ensemble des couples propriété-valeur de a et b. La similarité $sim_{pv}(a, b)$ entre a et b est calculée en fonction de la cardinalité de l'ensemble des couples comme suit :

$$sim_{pv}(a, b) = \frac{|\Sigma_a \cap \Sigma_b|}{|\Sigma_a \setminus \Sigma_b| + |\Sigma_b \setminus \Sigma_a| + |\Sigma_a \cap \Sigma_b|} \qquad (3)$$

Définition. Mesure de similarité agrégée. Soient $\mathcal{KG}$ un graphe de connaissances et a et b deux entités dans $\mathcal{KG}$, la mesure de similarité agrégée est définie comme suit :

$$sim(a, b) = \top(sim_l(a, b), sim_p(a, b), sim_{pv}(a, b)), \qquad (4)$$

avec $\top$ est la moyenne des similarités précédentes. Toutes les mesures sont normalisées dans l'intervalle $[0, 1]$, où un score égale à 0 signifie que les ressources comparées sont dissemblables, et le score 1 signifie que les ressources sont identiques.

5 Expérimentations et validation sur données de Data&Musée

5.1 Data&Musée

Ce travail est conduit dans le cadre du *projet Data & Musée* [4]. Ce projet vise à améliorer les capacités de différentes institutions culturelles en agrégeant et analysant des données en provenance de ces différentes institutions. Les données récoltées et traitées seront utilisées dans l'objectif de l'élargissement, la fidèlisation et la meilleure compréhension de leurs publics. Début 2019, les institutions partenaires sont les 14 musées de Paris Musées et les 84 monuments du Centre des Monuments Nationaux. Nous présentons dans la suite la constitution d'un graphe de contexte pour ces institutions.

4. Data & Musée, sélectionné dans le 23ième appel à projet du Fonds Unique Interministériel (FUI) et certifié par Cap Digital et Imaginove. `http://datamusee.fr/le-projet/`.

5.2 Création d'un graphe de contexte pour Data&Musée

Comme nous l'avons vu à la section 3, le graphe de contexte est construit à partir d'une liste d'entités, représentées par leurs URIs. La liste d'entité est soit choisie par un expert d'un domaine, soit constituée par des entités évidentes (par exemple, pour les musées de Paris Musées, nous cherchons à la main une entité de DBpedia-fr correspondant à chaque musée).

Le processus d'extraction du graphe de contexte est paramétré par la dimension N du graphe. C'est un processus récursif pour la collecte des nœuds voisins qui dépend du choix de N. Partant des travaux décrits dans la section 2 et en se basant sur les observations faites durant les expérimentations sur nos données, nous considérons que $N = 2$ est un bon choix pour la dimension d'un graphe de contexte.

Le contexte pour Paris Musées a été ainsi construit avec une profondeur de 2. Le cœur du contexte a donc une profondeur de 1. L'étude de l'impact du choix de cette profondeur sort du champs de cet article. Une *blacklist* a été créée comportant essentiellement tous les éléments de la *T-Box*, considérés comme nœuds terminaux. En effet, par exemple, si un nœud nous amenait à `owl:Thing` et qu'on suivait les liens à partir de là, nous ramènerions 1527645 entités pas nécessairement en rapport avec notre domaine.

Le tableau 1 donne une description d'un graphe de contexte extrait de DBpedia-fr pour la profondeur $N = 2$. Nous testons différents réglages pour la constitution d'un tel graphe. Les chiffres dans ce tableau ne sont donc qu'une indication sur un exemple.

	Contexte $\mathcal{CG}$	DBPedia-fr	%
Nœuds distincts	451653	10515624	4,29
Prédicats distincts	2310	20322	11,36
Liens	5150179	185404534	2,78
Liens par nœud (moyenne)	11,4	17,6	

TABLE *1 – Description d'un graphe de contexte $\mathcal{CG}$ extrait de DBPedia-fr avec $N = 2$*

Il est normal qu'il y ait moins de liens par nœuds dans $\mathcal{CG}$ que dans DBPedia-fr, puisque par construction nous avons éliminé certains liens peu porteurs d'information dans notre cadre applicatif comme expliqué plus haut.

Nous avons donc un nombre L de liens 36 fois inférieur et un nombre S de sommets 23 fois inférieur dans le $\mathcal{CG}$ que dans le $\mathcal{KG}$. Sur un algorithme qui est en $O(L + S)$ -comme le parcours en largeur (*Breadth-first search*)-, nous pouvons donc anticiper un gain d'un facteur de l'ordre 30, ce qui peut fortement contribuer à l'applicabilité de certaines méthodes. Les gains peuvent devenir considérables sur des algorithmes tels que ceux de recherche du plus court chemin entre deux nœuds si on souhaite donner un poids aux liens où on peut être en $O(S^2)$.

Nous devrons approfondir ces questions d'apport pour la scalabilité, mais ces premiers indices sont favorables. Nous allons voir dans la suite d'autres indicateurs qui plaident en faveur du graphe de contexte.

5.3 Validation du contexte obtenu

5.3.1 Couverture du domaine par le graphe de contexte

Pour évaluer la pertinence de notre contexte, nous avons voulu voir s'il conserve une bonne couverture des principaux éléments nous concernant dans la base de données Joconde.

La base Joconde est constituée de metadonnées concernant près de 600000 œuvres du patrimoine français et est disponible en Open Data. Chaque œuvre est principalement décrite par le lieu où elle se trouve (ville et institution), son ou ses créateurs, son titre, les techniques dont elle relève et des informations temporelles. Cette base est importante dans notre projet

puisqu'elle apporte un nombre considérable de données qui font autorité pour la description du patrimoine français.

Le tableau 2 illustre la couverture de Joconde par notre graphe de contexte. Nous avons pris les 10 villes associées au plus grand nombre d'œuvres dans la base Joconde et trouvées dans DBPedia-Fr et avons vérifié qu'elles sont bien dans notre graphe de contexte. Nous avons procédé de même pour les créateurs et les musées associés à des oeuvres.

	Liste	Dans $\mathcal{CG}$
Villes	Paris, Saint-Germain-en-Laye, Marseille, Strasbourg, Sèvres, Chantilly, Bordeaux, Montauban, Communauté urbaine Creusot-Montceau, Rennes	10/10
Domaines	Dessin, Archéologie, Peinture, Ethnologie, Estampe, Sculpture, Photographie, Céramique, Costume, Néolithique	10/10
Créateurs	A.Rodin, H.Chapu, E.Boudin, G.Moreau, JB.Barla, Y.Jean-Haffen, T.Chassériau, Manufacture nationale de Sèvres, JBC.Corot, E.Delacroix	9/10
Musées	Louvre, Musée d'Archéologie nationale, Cité de la céramique, Musée Rodin, Musée Condé, Musée Ingres, Musée des beaux-arts(Strasbourg), Musée des beaux-arts(Rennes), Musée des beaux-arts(Angers), Musée Gustave-Moreau	10/10

TABLE *2 – Couverture de la base Joconde par notre graphe de contexte*

On voit que la couverture est excellente. Un seul créateur n'a pas été trouvé, *J.B.Barla* : il s'agit d'un naturaliste qui a réalisé de nombreux dessins de plantes, mais n'est pas bien référencé par DBpedia-Fr à ce sujet et donc n'est pas reconnu comme un élément de notre domaine. Pour la couverture en nombre d'œuvres, on a, par exemple, 333114 œuvres associées aux villes mentionnées, soit plus de la moitié des œuvres capturées pour seulement 10 villes.

Cela démontre que notre graphe de contexte assure une bonne couverture de la base Joconde, qui est une des plus importantes pour le domaine du patrimoine culturel français.

5.3.2 Pertinence d'un graphe de contexte par mesure de similarité

Dans cette section nous montrons que des propriétés du graphe d'origine, importantes pour nos travaux, sont préservées dans le graphe de contexte construit.

Similarité liens

D'abord, notons que puisque nous récupérons les types (liens `is-a`) de toutes les entités présentes dans le $\mathcal{CG}$, par construction, le LCS de deux entités calculé sur le $\mathcal{KG}$ (DBpedia-Fr) et sur notre $\mathcal{CG}$ sont identiques pour toutes les entités. Cela constitue un premier indice de pertinence du $\mathcal{CG}$.

Cette première propriété permet d'affirmer que, pour chaque paire d'entités de notre $\mathcal{CG}$, la mesure de similarité de *Wu-Palmer* (formule (1), section 4.3) obtenue sur notre $\mathcal{CG}$ est identique à celle obtenue sur le $\mathcal{KG}$, puisqu'elle ne dépend que

— des mesures de LCS qui sont identiques sur les deux graphes,
— de la distance du LCS à la racine de la *T-Box* utilisée, qui est identique pour les deux graphes puisque nous avons inclu dans notre $\mathcal{CG}$ la *T-Box* de DBPedia-Fr.

Nous pourrons donc utiliser cette mesure de similarité sur notre $\mathcal{CG}$ sans perte d'information. Cela constitue un deuxième indice de pertinence de notre $\mathcal{CG}$.

Similarité propriétés

Nous avons utilisé la mesure de similarité des propriétés des entités définie par la mesure de *Tversky* (formule (2), section 4.3). Comme toutes les propriétés ne sont pas conservées

dans notre $\mathcal{CG}$, cette mesure de similarité donne des résultats différents sur le $\mathcal{CG}$ et sur le $\mathcal{KG}$. Dans ce cas, utiliser une corrélation de rang comme métrique pour évaluer la relation entre deux variables est un bon indicateur de préservation de cette mesure. Nous définissons ainsi une propriété de corrélation de rang et nous avons vérifié sa validité sur $\mathcal{CG}$.

Propriété. Corrélation de rang. Soient a, b et c trois entités de $\mathcal{CG}$ tels que $a, b, c \notin \overline{\mathcal{V}}$. Une corrélation de rang existe entre les paires d'entités (a, b) et (a, c) si :

$$sim_p^{\mathcal{KG}}(a,b) > sim_p^{\mathcal{KG}}(a,c) \Rightarrow sim_p^{\mathcal{CG}}(a,b) > sim_p^{\mathcal{CG}}(a,c) \qquad (5)$$

avec $sim_p^{\mathcal{KG}}$ et $sim_p^{\mathcal{CG}}$ sont les mesures de similarité respectives sur $\mathcal{KG}$ et $\mathcal{CG}$.

Pour vérifier la propriété (5), nous avons appliqué l'algorithme suivant sur un ensemble d'entités a et e choisies aléatoirement parmi l'ensemble des nœuds centraux :

— choisir $\{a_1, ..a_m\} \notin \overline{\mathcal{V}}$ et $\{e_1, ..e_n\} \notin \overline{\mathcal{V}}$

— $\forall l \in \{1, 2, ..m\}, \forall i \in \{1, 2, ..n\}$, calculer $sim_p^{\mathcal{KG}}(a_l, e_i)$ et $sim_p^{\mathcal{CG}}(a_l, e_i)$

— $\forall (e_i, e_j) \mid i, j \in \{1, 2, ..n\}, i \neq j$, vérifier la condition :

$$sim_p^{\mathcal{KG}}(a_l, e_i) > sim_p^{\mathcal{KG}}(a_l, e_j) \Rightarrow sim_p^{\mathcal{CG}}(a_l, e_i) > sim_p^{\mathcal{CG}}(a_l, e_j)$$

et compter le nombre de fois où elle est vérifiée.

Nous avons choisi $m = 100$ et $n = 10$ puis $m = 20$ et $n = 20$ et nous avons effectué $\frac{n(n+1)}{2} \times m$ vérifications pour valider la mesure de corrélation de rang. Le tableau 3 montre les résultats d'une série de tests pour ces différentes valeurs de m et n :

m	n	Nb. vérifications	Nb. succès	%
100	10	5500	5070	92,18
100	10	5500	5137	93,40
20	20	4200	3778	89,95
20	20	4200	3873	92,21

TABLE *3 – Corrélation de rang sur $\mathcal{KG}$ et $\mathcal{CG}$*

Ainsi, lorsque nous utilisons cette similarité pour comparer des éléments et en proposer à un utilisateur, dans 90% des cas ou plus, la proposition que nous pourrons faire sera identique à celle que nous aurions faite sur le $\mathcal{KG}$ complet.

Coefficient de corrélation rang-ordre de *Spearman*. Nous avons également calculé le coefficient de corrélation de *Spearman* qui est la métrique classique utilisée dans la littérature pour évaluer les mesures de similarité. Cette corrélation évalue la relation monotone entre deux variables.

De la même manière, nous avons calculé ce coefficient pour les valeurs $sim_p^{\mathcal{KG}}(a, e)$ et $sim_p^{\mathcal{CG}}(a, e)$ sur un ensemble d'entités a et e choisies aléatoirement. Nous avons réitéré ce processus plusieurs fois et nous avons calculé la mesure globale $sim^{\mathcal{KG}}(a, e)$ et $sim^{\mathcal{CG}}(a, e)$. Les résultats, décrits dans le tableau 4, montrent de très bonnes valeurs de corrélation.

La similarité globale $sim(a, e)$ est calculée comme moyenne de $sim_l(a, e)$ et $sim_p(a, e)$. Les expérimentations sur la similarité $sim_{pv}(a, e)$ définie par la formule (3) donnent de moins bon résultats. Ceci est dû au fait que toutes les propriétés-valeurs ne sont pas conservées dans $\mathcal{CG}$ (comme décrit plus haut). Les premiers résultats nous paraissent très encourageants et nous incitent à poursuivre l'exploitation de graphes de contexte dans le projet Data&Musée.

m	n	Corrélation Spearman
20	10	0.944
20	20	0.949
20	10	0.964
20	20	0.934

TABLE *4 – Corrélation de Spearman pour mesure de similarité sur $\mathcal{KG}$ et $\mathcal{CG}$*

6 Expérimentations et validation sur données de la base KORE

Dans un cadre général, pour évaluer la similarité entre entités, des données de référence ($benchmark$) existent. Afin de comparer l'utilisation des graphes de contexte $\mathcal{CG}$ avec l'utilisation du graphe $\mathcal{KG}$, nous utilisons le jeu de données de référence KORE (Hoffart *et al.*, 2012). Il contient 21 entités principales dans 5 domaines différents : *IT companies, Hollywood celebrities, Television series, Video games* et *Chuck Norris*. Pour chacune des entités principales, il contient 20 entités classées par similarité par rapport à celle-ci, la plus similaire étant classée en premier. Ceci résulte en 420 paires d'entités classées du plus au moins similaire. Nous utilisons la corrélation de Spearman comme métrique d'évaluation.

Nous avons identifié semi-automatiquement l'ensemble des entités de KORE dans DBpedia. Pour chacun des 5 domaines de KORE nous avons créé un graphe de contexte utilisant comme germes l'ensemble de ses entités qu'on passe en entrée à l'algorithme `CONTEXT`. En sortie nous avons 5 graphes de contextes sur lesquels nous évaluons la similarité pour les paires d'entités du jeu de données. Nous avons effectué les calculs de similarité entre les entités de KORE sur ces graphes et sur DBPedia, afin de comparer les résultats obtenus.

Le tableau 5 donne les résultats de la corrélation de Spearman entre $\mathcal{KG}$ et $\mathcal{CG}$ sur le jeu de données de référence. Chaque ligne du tableau décrit les valeurs de corrélation pour la mesure de similarité correspondante. La dernière ligne correspond à la corrélation sur le classement de la mesure de similarité calculée comme moyenne des trois précédentes. La colonne 'Moyenne' décrit les valeurs de corrélation pour toutes les entités des domaines de KORE considérés (le traitement du domaine *Video Games* a du être différé pour raison technique).

Mesure	IT Companies	Hollywood Celebrities	Television Series	Chuck Norris	Moyenne
$sim_l(a,b)$	1.0	0.999	0.995	0.898	0.973
$sim_p(a,b)$	0.997	0.998	0.994	0.998	0.997
$sim_{pv}(a,b)$	0.590	0.807	0.646	0.806	0.712
$sim(a,b)$	0.994	0.996	0.957	0.986	0,983

TABLE *5 – Corrélation de Spearman pour mesures de similarité sur $\mathcal{KG}$ et $\mathcal{CG}$(KORE)*

Nous observons sur le tableau que les mesures $sim_l(a,b)$, $sim_p(a,b)$ et $sim(a,b)$ donnent de très bonnes valeurs de corrélation entre les classements obtenus sur $\mathcal{KG}$ et ceux sur $\mathcal{CG}$, ce qui est un élément de confirmation de plus en faveur de l'utilisation de graphes de contexte. Comme dans le cas des graphes de contexte de Paris Musées (section 5.3.2), sur les graphes de contexte de KORE la mesure $sim_{pv}(a,b)$ donne de moins bons résultats. Des tests sont en cours pour améliorer les résultats de cette mesure.

7 Conclusion et perspectives

Dans cet article nous avons présenté la notion de graphe de contexte pour un domaine. Nous le définissons comme un extrait d'un plus grand graphe et qui cible les connaissances sur ce domaine. Nous avons montré que ce graphe peut être construit simplement en partant de quelques entités importantes du domaine utilisant un jeu de données de départ ou avec peu de connaissances expertes si elles sont disponibles. Nous avons aussi montré que le graphe obtenu présente des caractéristiques qui permettent de le substituer au grand graphe pour des exploitations classiques du graphe de connaissance (étude sur la similarité entre des éléments). Dans un proche avenir, nous comptons appliquer cette technique à d'autres domaines et exploiter les graphes de contexte obtenu pour leur appliquer des techniques d'apprentissage sur les graphes.

Références

AL-GHOSSEIN M., ABDESSALEM T. & BARRÉ A. (2018). Open data in the hotel industry : leveraging forthcoming events for hotel recommendation. *J. of IT & Tourism*, **20**(1-4), 191–216.

BEEK W., SCHLOBACH S. & VAN HARMELEN F. (2016). A contextualised semantics for owl :sameas. In *Proceedings of the 13th European Semantic Web Conference*, volume 9678 of *LNCS*, p. 405–419 : Springer.

BENEDETTI F., BENEVENTANO D., BERGAMASCHI S. & SIMONINI G. (2019). Computing inter-document similarity with context semantic analysis. *Information Systems*, **80**, 136 – 147.

BHATT S., PADHEE S., SHETH A., CHEN K., SHALIN V., DORAN D. & MINNERY B. (2019). Knowledge graph enhanced community detection and characterization. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, WSDM '19, p. 51–59.

CHENIKI N., BELKHIR A., SAM Y. & MESSAI N. (2016). Lods : A linked open data based similarity measure. In *2016 IEEE 25th International Conference on Enabling Technologies : Infrastructure for Collaborative Enterprises (WETICE)*, p. 229–234.

COLUCCI S., DONINI F. M., GIANNINI S. & SCIASCIO E. D. (2016). Defining and computing least common subsumers in rdf. *J. Web Semant.*, **39**, 62–80.

CONSOLI S., MONGIOVÌ M., NUZZOLESE A. G., PERONI S., PRESUTTI V., RECUPERO D. R. & SPAMPINATO D. (2015). A smart city data model based on semantics best practice and principles. In *WWW*.

GYRARD A., ATEMEZING G., BONNET C., BOUDAOUD K. & SERRANO M. (2016). Reusing and unifying background knowledge for internet of things with lov4iot. In *2016 IEEE 4th International Conference on Future Internet of Things and Cloud (FiCloud)*, p. 262–269.

HARISPE S., SÁNCHEZ D., RANWEZ S., JANAQI S. & MONTMAIN J. (2014). A framework for unifying ontology-based semantic similarity measures : A study in the biomedical domain. *Journal of Biomedical Informatics*, **48**, 38 – 53.

HOFFART J., SEUFERT S., BA NGUYEN D., THEOBALD M. & WEIKUM G. (2012). Kore : Keyphrase overlap relatedness for entity disambiguation.

HULPUS I., HAYES C., KARNSTEDT M. & GREENE D. (2013). Unsupervised graph-based topic labelling using dbpedia. In *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*, WSDM '13, p. 465–474.

LI Y., BANDAR Z. A. & MCLEAN D. (2003). An approach for measuring semantic similarity between words using multiple information sources. *IEEE Transactions on Knowledge and Data Engineering*, **15**(4), 871–882.

LODI G., ASPRINO L., NUZZOLESE A. G., PRESUTTI V., GANGEMI A., RECUPERO D. R., VENINATA C. & ORSINI A. (2017). *Semantic Web for Cultural Heritage Valorisation*, In *Data Analytics in Digital Humanities*, p. 3–37. Springer International Publishing : Cham.

LUO Y., WANG Q., WANG B. & GUO L. (2015). Context-dependent knowledge graph embedding. p. 1656–1661.

ORAMAS S., OSTUNI V., DI NOIA T., SERRA X. & DI SCIASCIO E. (2016). Sound and music recommendation with knowledge graphs. *ACM Transactions on Intelligent Systems and Technology (TIST)*, **8**, 1–21.

PAUL C., RETTINGER A., MOGADALA A., KNOBLOCK C. A. & SZEKELY P. A. (2016). Efficient graph-based document similarity. In *International Semantic Web Conference (ISWC) 2016*.

PETRAKIS E. G. M., VARELAS G., HLIAOUTAKIS A. & RAFTOPOULOU P. (2006). X-similarity : Computing semantic similarity between concepts from different ontologies. *Journal of Digital Information Management (JDIM*, **4**.

RAAD J., PERNELLE N. & SAÏS F. (2017). Détection de liens d'identité contextuels dans une base de connaissances. In *IC 2017 - 28es Journées francophones d'Ingénierie des Connaissances*, p. 56–67, Caen, France.

RADA R., MILI H., BICKNELL E. & BLETTNER M. (1989). Development and application of a metric on semantic nets. *IEEE Transactions on Systems, Man, and Cybernetics*, p. 17–30.

RODRIGUEZ M. A. & EGENHOFER M. J. (2003). Determining semantic similarity among entity classes from different ontologies. *IEEE Transactions on Knowledge and Data Engineering*, **15**(2), 442–456.

SÁNCHEZ D., BATET M., ISERN D. & VALLS A. (2012). Ontology-based semantic similarity : A new feature-based approach. *Expert Syst. Appl.*, **39**, 7718–7728.

SHEN W., WANG J. & HAN J. (2015). Entity linking with a knowledge base : Issues, techniques, and solutions. *IEEE Transactions on Knowledge and Data Engineering*, **27**(2), 443–460.

SHI J., GAO H., QI G. & ZHOU Z. (2017). Knowledge graph embedding with triple context. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, CIKM '17, p. 2299–2302.

SOUALAH ALILA F., COUSTATY M., REMPULSKI N. & DOUCET A. (2016). Datatourism : designing an architecture to process tourism data. In *IFITT and ENTER 2016 Conferences*.

SÁNCHEZ D. & BATET M. (2011). Semantic similarity estimation in the biomedical domain : An ontology-based information-theoretic perspective. *Journal of Biomedical Informatics*, **44**(5), 749 – 759.

TRAVERSO I., VIDAL M.-E., KÄMPGEN B. & SURE-VETTER Y. (2016). Gades : A graph-based semantic similarity measure. In *Proceedings of the 12th International Conference on Semantic Systems*, SEMANTiCS 2016, p. 101–104.

TRAVERSO-RIBÓN I. & VIDAL M. (2015). Exploiting information content and semantics to accurately compute similarity of go-based annotated entities. In *2015 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*, p. 1–8.

TVERSKY A. (1977). Features of similarity. *Psychological Review*, **84**(4), 327–352.

WANNOUS R., MALKI J., BOUJU A. & VINCENT C. (2017). Trajectory ontology inference considering domain and temporal dimensions—application to marine mammals. *Future Generation Computer Systems*, **68**, 491 – 499.

WU Z. & PALMER M. (1994). Verbs semantics and lexical selection. In *Proceedings of the 32Nd Annual Meeting on Association for Computational Linguistics*, ACL '94, p. 133–138.