UNIVERSITÉ DE MONTRÉAL

ÉVALUATION ET AMÉLIORATION DE LA QUALITÉ DE DBPEDIA POUR LA REPRÉSENTATION DE LA CONNAISSANCE DU DOMAINE

LUDOVIC FONT

DÉPARTEMENT DE GÉNIE INFORMATIQUE ET GÉNIE LOGICIEL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION

DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES

(GÉNIE INFORMATIQUE)

DÉCEMBRE 2016

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé :

ÉVALUATION ET AMÉLIORATION DE LA QUALITÉ DE DBPEDIA POUR LA REPRÉSENTATION DE LA CONNAISSANCE DU DOMAINE

présenté par : FONT Ludovic

en vue de l'obtention du diplôme de : Maîtrise ès sciences appliquées

a été dûment accepté par le jury d'examen constitué de :

M. ADAMS Bram, Doctorat, président

M. GAGNON Michel, Ph. D., membre et directeur de recherche

Mme ZOUAQ Amal, Ph. D., membre et codirectrice de recherche

Mme KOSSEIM Leila, Ph. D., membre externe

REMERCIEMENTS

Je tiens à remercier mes directeurs de recherche, Michel Gagnon et Amal Zouaq, pour leur encadrement lors de cette entrée dans le monde de la recherche, ainsi que pour leurs nombreux conseils m'ayant permis de m'améliorer tout au long de ces deux ans de projet. En particulier, je remercie Amal Zouaq pour avoir financé ce projet de recherche. Je remercie aussi l'École Nationale d'Informatique et de Mathématiques Appliquées de Grenoble pour m'avoir permis de mener à bien ce projet de double-diplôme en partenariat avec l'École Polytechnique de Montréal. Je remercie enfin mon père et Nicolas pour avoir courageusement relu et aidé à corriger ce mémoire.

RÉSUMÉ

L'évolution récente du Web sémantique, tant par la quantité d'information offerte que par la multiplicité des usages possibles, rend indispensable l'évaluation de la qualité des divers ensembles de données (datasets) disponibles. Le Web sémantique étant basé sur la syntaxe RDF, i.e. des triplets <sujet, relation, objet> (par exemple <Montréal, est une ville de, Québec>), on peut le voir comme un immense graphe, où un triplet relie un nœud « sujet » et un nœud « objet » par une arête « relation ». Chaque dataset représente ainsi un sous-graphe. Dans cette représentation, *DBpedia*, un des datasets majeurs du Web sémantique, en est souvent considéré comme le nœud central. En effet, DBpedia a pour vocation, à terme, de pouvoir représenter toute l'information présente dans Wikipedia, et couvre donc une très grande variété de sujets, permettant de faire le lien avec tous les autres datasets, incluant les plus spécialisés. C'est de cette multiplicité des sujets couverts qu'apparait un point fondamental de ce projet : la notion de « domaine ». Informellement, nous considérons un domaine comme étant un ensemble de sujets reliés par une thématique commune. Par exemple, le domaine *Mathématiques* contient plusieurs sujets, comme *algèbre*, *fonction* ou *addition*. Formellement, nous considérons un domaine comme un sous-graphe de DBpedia, où l'on ne conserve que les nœuds représentant des concepts liés à ce domaine.

En l'état actuel, les méthodes d'extraction de données de DBpedia sont généralement beaucoup moins efficaces lorsque le sujet est abstrait, conceptuel, que lorsqu'il s'agit d'une entité nommée, par exemple une personne, ville ou compagnie. Par conséquent, notre première hypothèse est que l'information disponible sur DBpedia liée à un domaine est souvent pauvre, car nos domaines sont essentiellement constitués de concepts abstraits. La première étape de ce travail de recherche fournit une évaluation de la qualité de l'information conceptuelle d'un ensemble de 17 domaines choisis semi-aléatoirement, et confirme cette hypothèse. Pour cela, nous identifions plusieurs axes permettant de chiffrer la « qualité » d'un domaine : 1 - nombre de liens entrants et sortants pour chaque concept, 2 - nombre de liens reliant deux concepts du domaine par rapport aux liens reliant le domaine au reste de DBpedia, 3 - nombre de concepts *typés* (i.e. représentant l'*instance* d'une *classe*, par exemple *Addition* est une instance de la classe *Opération mathématique* : le concept *Addition* est donc typé si la relation <addition, instance de, opération mathématique> apparait dans DBpedia). Nous arrivons à la conclusion que l'information conceptuelle contenue dans DBpedia est effectivement incomplète, et ce selon les trois axes.

La seconde partie de ce travail de recherche est de tenter de répondre au problème posé dans la première partie. Pour cela, nous proposons deux approches possibles. La première permet de fournir des classes potentielles, répondant en partie à la problématique de la quantité de concepts typés. La seconde utilise des systèmes d'extraction de relations à partir de texte (ORE – Open Relation Extraction) sur l'*abstract* (i.e. premier paragraphe de la page Wikipedia) de chaque concept. En classifiant les relations extraites, cela nous permet 1) de proposer des relations inédites entre concepts d'un domaine, 2) de proposer des classes potentielles, comme dans la première approche. Ces deux approches ne sont, en l'état, qu'un début de solution, mais nos résultats préliminaires sont très encourageants, et indiquent qu'il s'agit sans aucun doute de solutions pertinentes pour aider à corriger les problèmes démontrés dans la première partie.

ABSTRACT

In the current state of the semantic web, the quantity of available data and the multiplicity of its uses impose the continuous evaluation of the quality of this data, on the various Linked Open Data (LOD) datasets. These datasets are based on the RDF syntax, i.e. <subject, relation, object> triples, such as <Montréal, is a city of, Québec>. As a consequence, the LOD cloud can be represented as a huge graph, where every triple links the two nodes "subject" and "object", by an edge "relation". In this representation, each dataset is a sub-graph. *DBpedia*, one of the major datasets, is colloquially considered to be the central hub of this cloud. Indeed, the ultimate purpose of DBpedia is to provide all the information present in Wikipedia, "translated" into RDF, and therefore covers a wide range of domains, allowing a linkage with every other LOD dataset, including the most specialized. From this wide coverage arises one of the fundamental concepts of this project: the notion of "domain". Informally, a domain is a set of subjects with a common thematic. For instance, the domain *Mathematics* contains several subjects such as *algebra, function* or *addition*. More formally, a domain is a sub-graph of DBpedia, where the nodes represent domain-related concepts.

Currently, the automatic extraction methods for DBpedia are usually far less efficient when the target subject is conceptual than when it is a named entity (such as a person, city or company). Hence our first hypothesis: the domain-related information available on DBpedia is often poor, since domains are constituted of concepts. In the first part of this research project, we confirm this hypothesis by evaluating the quality of domain-related knowledge in DBpedia for 17 domains chosen semi-randomly. This evaluation is based on three numerical aspects of the "quality" of a domain: 1 – number of inbound and outbound links for each concepts, 2 – number of links between two domain concepts compared to the number of links between the domain and the rest of DBpedia, 3- number of *typed* concepts (i.e. representing the *instance* of a *class* : for example, *Addition* is an instance of the class *Mathematical operation* : the concept *Addition* is typed if the relation <addition, type, mathematical operation> appears in DBpedia). We reach the conclusion that the domain-related, conceptual information present in DBpedia is indeed poor on the three axis.

In the second half of this work, we give two solutions to the quality problem highlighted in the first half. The first one allows to propose potential classes that could be added in DBpedia, addressing the 3rd quality aspect: number of typed concepts. The second one uses an Open Relation Extraction (ORE) system that allows to detect relations in a text. By using this system on the *abstract* (i.e. the

first paragraph of the Wikipedia page) of each concept, and classifying the extracted relation depending on their semantic meaning, we can 1) propose novel relations between domain concepts, and 2) propose additional potential classes. These two methods currently only represent the first step, but the preliminary results we obtain are very encouraging, and seem to indicate that they are absolutely relevant to help correcting the issues highlighted in the first part.

TABLE DES MATIÈRES

LISTE DES TABLEAUX

LISTE DES FIGURES

LISTE DES ANNEXES

LISTE DES SIGLES ET ABRÉVIATIONS

DL : Descriptive Logic. Il s'agit de l'un des groupes de prédicats que nous définissons dans le premier article.

LOD : Linked Open Data. Représente l'ensemble des datasets respectant les standards du Linked Data et dont les données sont accessibles à tous.

MCCov : Macro Concept Coverage Ratio. Métrique que nous définissons dans le premier article et utilisons pour évaluer DBpedia.

mCCov : micro Concept Coverage Ratio. Métrique que nous définissons dans le premier article et utilisons pour évaluer DBpedia.

NLP : Natural Language Processing, ou traitement des langues naturelles. Regroupe tous les outils permettant d'analyser du texte afin d'en extraire de l'information.

ORE : Open Relation Extraction. Système permettant d'extraire des relations à partir d'un texte sans aucune information extérieure au texte lui-même.

RDF : Resource Description Framework. Modèle permettant de décrire de l'information de façon formelle par des triplets <sujet, prédicat, objet>. Il s'agit du langage de base du Web sémantique.

SPARQL : SPARQL Protocol and RDF Query Language. Langage de requête, similaire au SQL, permettant d'interroger et de manipuler des données RDF présentes n'importe où sur le Web.

TCov : Triple Coverage ratio. Métrique que nous définissons dans le premier article et utilisons pour évaluer DBpedia.

URI : Uniform Resource Identifier. Chaine de caractères permettant d'identifier de manière unique une ressource (typiquement, une page web). Dans le cadre du Web sémantique, tout (entité, prédicat, classe…) est représenté par une URI.

CHAPITRE 1 INTRODUCTION

1.1 Contexte

Depuis sa création en 1989, le World Wide Web est l'une des technologies ayant le plus radicalement évolué, tant au niveau de la quantité de contenu disponible que de son format et du nombre d'applications possibles. Aujourd'hui, la quantité phénoménale d'information à disposition exige des outils d'indexage et de recherche particulièrement performants. Ce rôle est pour l'instant rempli par des moteurs de recherche tels que Google, Bing ou Yahoo. Bien que leur efficacité ne soit pas à démontrer, ils restent limités par la manière dont l'information est présente sur le Web. En effet, le Web est avant tout destiné aux humains et n'a pas été conçu pour être parcouru facilement par une machine. Les machines, en l'occurrence les moteurs de recherche, doivent donc se contenter d'effectuer leurs recherches dans le texte des pages. Le développement des outils de traitement automatique des langues naturelles (TALN, ou NLP - Natural Language Processing) permet d'améliorer chaque jour les algorithmes d'exploration du Web. Cependant, il s'agit d'une solution de forme et non de fond. La tâche d'extraction d'information depuis le texte est très délicate, et le Web est de plus en plus orienté vers du contenu multimédia, composé d'images, de son et de vidéos, pour lesquels l'extraction d'information est encore plus difficile et moins aboutie.

C'est ici qu'apparait le paradigme du Web Sémantique. Il s'agit d'un ensemble de standards définis par le World Wide Web Consortium (W3C) ayant pour but de créer un « web de données », où l'information serait donc autant accessible que s'il s'agissait d'une gigantesque base de données. Ceci permet aux moteurs de recherche et aux machines de manière générale de parcourir le Web beaucoup plus rapidement et efficacement, car toutes les entités et tous les concepts sont interconnectés. Par exemple, dans une page web contenant un article de journal sur Barack Obama, ce dernier sera souvent référé par « Le président des États-Unis », « M. le président », « Il » … Il sera donc difficile et tortueux pour un moteur de recherche de récupérer toutes les informations reliées à Barack Obama dans cet article. Dans le monde du Web Sémantique, l'article dispose, en plus de son contenu textuel destiné aux humains, une surcouche destinée aux machines. Cette surcouche devrait théoriquement contenir la majeure partie de cette information, mais dans une structure différente, reliant par exemple tous les termes mentionnés précédemment à l'entité « Barack Obama ». Pour l'utilisateur, tout est invisible, mais pour la machine, la manière de

parcourir le web est radicalement changée et simplifiée, offrant une myriade de nouvelles possibilités.

Ceci représente cependant un certain nombre de défis. Pour l'instant, il ne s'agit pas, comme dit précédemment, d'une surcouche ajoutée aux sites « classiques », mais plutôt de nouveaux sites regroupant de l'information. Certains se spécialisent dans certains domaines, comme GeoNames pour la géographie ou MusicBrainz pour la musique, et d'autres sont plus généraux, comme DBpedia, Freebase ou Yago. Ces datasets doivent respecter quatre contraintes, nommées « Linked Data principles » et établies par le W3C[1] :

- Utiliser des URIs (par exemple http://dbpedia.org/page/Barack_Obama) en guise de noms ;
- Associer ces URIs à des pages http afin que les utilisateurs puissent obtenir plus d'information sur le sujet ;
- Fournir de l'information respectant certains standards sur ces pages ;
- Inclure dans cette information des liens vers d'autres URIs afin de permettre la découverte d'autres sujets.

L'information respectant ces contraintes est nommée Linked Data. L'ensemble du Linked Data constitue le Web Sémantique. Lorsque l'information est ouverte et disponible actuellement, on parle de Linked Open Data (LOD)[2].

Le dataset généralement considéré comme le nœud central du LOD est DBpedia. Il s'agit d'un dataset ayant pour vocation de fournir un équivalent à Wikipedia, qui respecte les standards susmentionnés. Il s'agit d'un nœud central car, comme Wikipedia, il couvre la plupart des domaines de connaissance, ce qui lui permet d'être connecté à la plupart des autres datasets, incluant les plus spécialisés. Dans sa version 2014, DBpedia décrit 4,58 millions d'entités. Ces entités sont extraites par une méthode automatique basée sur les infobox de Wikipedia, c'est-à-dire les encarts se trouvant à droite de la plupart des pages et regroupant une certaine quantité d'information structurée. Ces entités sont ensuite regroupées dans DBpedia autour d'une ontologie (i.e. un schéma représentant des classes, indiquant par exemple que *Mathématicien* est une classe, sous-classe de *Scientifique*, elle-même sous-classe de *Personne*). Contrairement aux entités,

[1] https://www.w3.org/wiki/LinkedData

[2] https://www.w3.org/wiki/SweoIG/TaskForces/CommunityProjects/LinkingOpenData

l'ontologie est créée manuellement par la communauté afin de s'assurer de sa qualité, et comprend à ce jour 754 classes[3].

1.2 Problématique et objectifs

L'utilité de DBpedia n'est plus à prouver. Son rôle central en fait un point de passage quasiment incontournable pour un outil explorant le Web sémantique. C'est pour cette raison qu'il est crucial d'évaluer continuellement la qualité de l'information contenue dans DBpedia. Les méthodes actuelles de construction, basées sur les infobox Wikipedia, assurent une certaine quantité de données. Cependant, ces méthodes présentent des faiblesses, notamment lorsque l'entité traitée représente un concept abstrait, où les infobox sont en général peu remplis voire même inexistants. C'est de cette constatation que nait l'hypothèse directrice de notre travail de recherche : DBpedia contient peu d'information sur les concepts. Ceci est particulièrement problématique lorsque l'on recherche de l'information sur un domaine en particulier. Ce mémoire décrit les méthodes mises en place pour évaluer la qualité de l'information conceptuelle pour un domaine de connaissances quelconque dans DBpedia, et propose des méthodes pour améliorer cette qualité. Nos questions de recherche sont les suivantes :

Q1 : comment définir exactement ce qu'est un « domaine de connaissance » ?

Q2 : quels aspects de la « qualité » d'un domaine pouvons-nous évaluer quantitativement, et quel est le résultat de cette évaluation ?

Q3 : par quelles méthodes pouvons-nous améliorer la qualité de l'information conceptuelle de DBpedia ?

1.3 Contenu du mémoire

Ce mémoire est essentiellement constitué de deux articles. Avant ceux-ci, le chapitre 2 présente une revue de littérature indiquant les principaux travaux effectués auparavant concernant l'évaluation et l'amélioration de DBpedia, ainsi que ceux évaluant certains outils que nous utilisons. Le chapitre 3 présente de manière détaillée la démarche globale que nous avons suivie lors de ce travail. Les chapitres 4 et 5 contiennent les deux articles rédigés au long de ce projet. Le

[3] Version 2016-04 : http://wiki.dbpedia.org/dbpedia-version-2016-04

chapitre 6 propose une discussion générale sur l'ensemble de ce projet. Enfin, le chapitre 7 présente la conclusion.

Plus en détails, le chapitre 4 contient l'article *Assessing the Quality of Domain Concepts Descriptions in DBpedia* [1], publié dans les actes de la conférence *11th International Conference on Signal Image Technology & Internet based Systems* (2015). Cet article présente une première série d'évaluations sur la qualité de l'information conceptuelle liée à un domaine présente dans DBpedia, concluant que DBpedia présente des faiblesses non négligeables sur cet aspect.

Le chapitre 5 présente l'article *Assessing and Improving Domain Knowledge Representation in DBpedia* [2], qui a été soumis à l'*Open Journal of Semantic Web*. Dans cet article, nous présentons, dans un premier temps, une extension du travail présenté dans le premier article (chapitre 4), en évaluant DBpedia sur un plus grand nombre de domaines, avec de nouvelles méthodes. Dans un second temps, nous y proposons des idées d'approches pour résoudre partiellement les problèmes identifiés, ainsi qu'une évaluation préliminaire de ces approches.

CHAPITRE 2 REVUE DE LITTÉRATURE

Ce chapitre présente les différents travaux effectués ayant servi de base à notre travail. La première section présente ceux concernant l'évaluation de la qualité du LOD en général. La seconde présente ceux concernant DBpedia en particulier. La troisième section présente les outils dont nous nous sommes servis dans ce travail.

2.1 Le Linked Open Data

Là où le Web classique est originellement à l'usage des humains, le Linked Open Data (LOD) est conçu pour être exploré automatiquement. L'information y est en effet stockée exclusivement sous forme de triplets RDF <sujet, relation, objet> [3]. Les machines n'ayant aucune capacité intrinsèque de vérification et de correction de l'information, il est fondamental de concevoir des outils pour évaluer la qualité du LOD automatiquement.

La première question à se poser est « comment définir si un dataset est de bonne qualité ? » La réponse usuelle à cette question est le « fitness for use » [4], en d'autres termes, est-ce que le dataset est approprié pour les tâches dans lesquelles il est utilisé. Dans le cas du LOD, les tâches confiées sont principalement l'exploration et la génération de connaissances. Cela permet d'extraire

plusieurs aspects de la qualité d'un dataset. Un article de Zaveri et al. [5] offre une catégorisation exhaustive des différents aspects de la qualité d'un dataset du LOD proposés dans la littérature, basée en grande partie sur les travaux de Flemming [6]. La Table 2.1 fournit l'organisation de ces aspects telle que présentée dans cet article [5]. Comme on peut le constater, ce sont essentiellement des caractéristiques de forme et non de fond. Notre travail visant à évaluer la qualité de la connaissance dans DBpedia, ces aspects, et les métriques en découlant, ne sont pas pertinents pour nous.

Un article de Bizer et al. [7] présente une autre classification des métriques, cette fois pour évaluer spécifiquement la pertinence de l'information présente quelque part sur le Web (non restreint au LOD). Les métriques y sont classées selon trois catégories :

- Métriques basées sur le contenu
- Métriques basées sur le contexte
- Métriques basées sur une notation par un agent extérieur

Tableau 2.1 : Aspects de la qualité d'un dataset du LOD tels que présentées dans l'article de Zaveri et al.

Dimension	Aspect
Disponibilité	Accessibilité du terminal et serveur SPARQL
	Accessibilité des données RDF
	Validité des URI
	Aucune erreur dans le type du contenu
	Validité des liens sortants
License	Indication d'une licence lisible par la machine
	Indication d'une licence lisible par l'humain
	Validité de la licence
Interliens	Détection d'interliens de bonne qualité
	Existence de liens vers des datasets externes
	Validité des liens entrants
Securité	Utilisation de signatures digitales
	Authenticité du dataset
Performance	Usage d'URIs - slash
	Latence faible
	Haut débit
	Évolutivité du dataset

Dans le cas du LOD, qui a une structure de graphe, le contexte et le contenu représentent la même information, i.e. le voisinage de chaque nœud. Par ailleurs, en l'état, les datasets du LOD n'offrent

pas à notre connaissance de possibilité de noter leur qualité par des agents extérieurs. Le framework proposé dans cet article, basé en grande partie sur une telle notation par un agent extérieur, ne nous est donc pas utile.

L'article de Hogan et al. [8] se rapproche également de notre sujet, car il présente une évaluation de plusieurs datasets du LOD selon deux aspects : respect des paradigmes du LOD et score PageRank. Une conclusion notable est que DBpedia fait partie des datasets les mieux notés sur les deux aspects. Cependant, aucune évaluation n'est faite sur la pertinence du contenu de ces datasets.

Les travaux de Debattista et al. [9] et Mendes et al. [10] incorporent la notion de domaine, fondamentale dans notre travail. Ils fournissent deux méthodes proposant un processus général pour évaluer finement, par une personne non experte en informatique mais spécialiste dans un domaine de connaissances, la qualité d'un dataset selon la tâche à effectuer. Dans les deux cas, aucune métrique n'est fournie, et les travaux sont orientés vers une utilisation pour un domaine en particulier, alors que notre but est d'évaluer la connaissance de DBpedia dans tous les domaines.

2.2 DBpedia

DBpedia est généralement considéré comme le point central du LOD. En effet, ce dataset a pour but de convertir le contenu de Wikipedia aux standards du LOD, afin de permettre d'effectuer des requêtes complexes, tout en reliant les autres datasets du LOD, ce qui ouvre des myriades de possibilités d'applications [11].

L'une des difficultés majeures est d'effectuer cette conversion depuis le format principalement textuel de Wikipedia vers le format RDF [3] utilisé par le LOD. Ce format requiert de convertir les données en triplets <*sujet, relation, objet*>, et par conséquent d'identifier de telles relations dans le contenu de Wikipedia. Une telle identification dans du texte brut est particulièrement difficile : afin de limiter les erreurs et de garantir une certaine qualité, DBpedia contient essentiellement de l'information extraite des infobox de Wikipedia. Ces infobox sont les encadrés présents sur la plupart des pages Wikipedia, qui regroupent de l'information de manière plus structurée, ce qui permet de la convertir en RDF plus facilement [12].

Cette approche a des faiblesses notables, particulièrement dans le cas de connaissances abstraites, que nous étudions dans ce travail. En effet, de manière générale, les concepts (tels qu'*Arithmétique* ou *Addition*) ont une infobox fournissant très peu d'information, ou pas d'infobox du tout. De cette

constatation découle l'hypothèse qui a guidé notre travail : DBpedia offre peu d'information sur les concepts. Étant donné que DBpedia a pour but de représenter toute l'information contenue dans Wikipédia, il s'agit d'une faiblesse importante.

Plusieurs études ont été effectuées pour évaluer la qualité générale de DBpedia. L'étude menée par Zaveri et al. [13] est basée notamment sur un retour d'évaluateurs humains, et identifie de nombreux problèmes de forme, avec 11.93% des triplets analysées erronés. L'évaluation de la validité d'un triplet est basée sur les métriques catégorisées dans leur précédent travail [5].

Une autre étude, menée par Kontokostas et al. [14], analyse plusieurs datasets du LOD selon la richesse du schéma (les propriétés ont-elles un *rdfs:domain / rdfs:range* ?) et le respect des contraintes imposées par ce schéma (si le schéma indique que la propriété *dbo:spouse* doit relier deux personnes, est-ce qu'il y a des cas où le sujet ou l'objet de cette propriété n'est pas une personne ?) Les aspects considérés sont la complétude (il manque de l'information), la consistance (des contradictions apparaissent lorsque l'on regarde le dataset dans son ensemble) et la concision (trop de triplets inutiles noient l'information utile). Dans cette évaluation, 817 millions de triplets de DBpedia sont évalués, et 63 millions contiennent au moins une erreur.

Certains travaux proposent des solutions pour les corriger, à la fois pour DBpedia et pour le LOD en général, comme l'approche basée sur le crowdsourcing proposée par Acosta et al. [15]. Cependant, comme précédemment indiqué, aucun travail n'évalue à notre connaissance l'aspect de fond qui nous intéresse, soit celui de la qualité de l'information conceptuelle, liée à un domaine, dans DBpedia. La notion de *domaine* est importante car Wikipédia représente une base de connaissance couvrant la majorité des domaines : on y trouve de l'information précise aussi bien sur la médecine que sur les sports ou les mathématiques. Par conséquent, nous nous intéressons plus spécifiquement à savoir si DBpedia contient suffisamment d'information conceptuelle au sein d'un domaine de connaissances donné.

2.3 Outils utilisés

Au cours de ce projet, nous avons à plusieurs reprises fait appel à des outils externes afin d'effectuer des tâches complexes dont seul le résultat, et non la méthode, est pertinent. Nous utilisons deux catégories d'outils : les annotateurs sémantiques et les extracteurs de relations.

Annotation sémantique : l'annotation sémantique a pour but d'identifier, dans un texte ou corpus de textes donné en entrée, les mots ou groupes de mots importants, et de leur apposer des métadonnées, comme par exemple le type d'entité grammaticale (sujet, verbe…). Ce procédé général inclut plusieurs opérations : reconnaissance des entités nommées (NER : Named Entity Recognition) [16], identification de concepts [17] ou analyse d'opinion ou de sentiment de l'auteur [18]. De nombreux outils existent, aux performances variables [19], [20]. Leur efficacité peut dépendre de nombreux facteurs, comme le type du texte (article de journal, extrait de forum internet, retranscription de discussion…), la tâche à effectuer, le nombre de textes présents dans le corpus, ou de manière plus générale la taille du texte, et donc la quantité d'information disponible.

Dans cette recherche, nous utilisons l'annotation sémantique uniquement dans le second article, afin d'affiner notre méthode d'extraction de concepts liés à un domaine (pour plus de détail, voir la section 3.2 dans ce mémoire). En effet, notre but est d'identifier, dans l'abstract d'un concept, tous les autres concepts importants, afin de les ajouter au domaine initial. Par la suite, nous proposons une méthode d'identification de nouvelles relations et classes pour DBpedia basée sur l'extraction de relations.

Extraction de relations : l'extraction de relation est un procédé consistant à identifier dans un texte donné des relations entre entités. Souvent, ces méthodes sont basées sur de l'apprentissage machine, où un noyau (« kernel ») est construit à l'aide d'un corpus d'entraînement, et utilisé par la suite pour analyser le texte voulu [21], [22]. De nombreux outils utilisant cette approche ont été développés, comme RelExt [23], RelEx [24], ou encore le framework de Weston et al. [25].

Dans ce travail, nous nous intéressons plus spécifiquement à un sous-ensemble de l'extraction de relation nommé « Open Relation Extraction » (ORE). Il s'agit d'un dérivé du paradigme introduit par Banko et al. [26], l'« Open Information Extraction », consistant à extraire de l'information d'un texte sans aucune intervention humaine. Ceci permet d'avoir un système totalement automatique et dont l'efficacité ne dépend pas ou peu de la taille de l'entrée ou du domaine. L'intérêt de l'Open Information Extraction par rapport à l'Information Extraction classique est discuté par Banko et Etzioni [27]. Ils concluent que le rappel de l'OIE est notablement inférieur, mais que l'avantage est de ne pas avoir à disposer de corpus d'entraînement, et que l'on n'a pas à connaître a priori la quantité et la nature des relations que l'on compte extraire.

Cette approche a récemment subi des évolutions notables par l'ajout de nouvelles méthodes, comme l'utilisation de sources externes sur le Web [28] ou l' « inférence jointe » [29]. Il existe plusieurs extracteurs de relations respectant ce paradigme [30]–[32], parmi lesquels nous avons choisi le système ReVerb [30]. D'autres systèmes existent, comme Ollie [33], qui résout un des problèmes majeurs de ReVerb, qui ne considère que des relations axées autour d'un verbe. Cependant, ReVerb est un système reconnu, notamment considéré comme l'état de l'art et utilisé comme référence par Min et al. [34].

L'intérêt principal de ReVerb par rapport aux autres systèmes est d'imposer des contraintes sur le format du *prédicat* de la relation (l'élément central de la relation extraite <*sujet, prédicat, objet*>). En effet, afin d'éviter 1) les relations incohérentes (comme *contains omits* ou *was central torpedo*) et 2) les relations non informatives (<*Faust, made, a deal*> au lieu de <*Faust, made a deal with, the devil*>), le prédicat de la relation peut être soit un simple verbe, soit un verbe suivi par une préposition (*located in*), soit un verbe suivi par des noms, adjectifs ou adverbes, puis une proposition (*has atomic weight of*). Pour éviter d'obtenir des prédicats inutilement longs (*is offering only modest greenhouse gas reduction targets at*), ReVerb impose également qu'un prédicat donné doit apparaitre plusieurs fois dans le corpus, ce qui est très improbable dans le cas de l'exemple donné.

Plus spécifiquement, ReVerb prend en entrée un texte morpho-syntaxiquement annoté (i.e. où chaque mot est annoté pour indiquer sa nature grammaticale : s'il s'agit d'un verbe, et à quel temps, ou d'un pronom, etc.) et où les syntagmes nominaux (noun phrases ou NPs) sont identifiés, c'est-à-dire où les blocs de la phrase pouvant être remplacés par un nom sont identifiés. Par exemple, la phrase *We saw the yellow dog* contient deux blocs NP: *we* et *the yellow dog*. Ceci permet de ne plus être limité aux noms mais aux blocs représentant une seule entité.

Par la suite, ReVerb essaye de construire autour de chaque verbe v un prédicat r_v commençant par v et respectant les contraintes précédentes. Ensuite, il récupère les blocs NP les plus proches à gauche et à droite, et ceux-ci deviennent le sujet et l'objet de la relation. Enfin, un score de confiance est calculé, basé sur une table de propriétés (par exemple, le score augmente dramatiquement si la relation couvre tous les mots d'une phrase, et diminue s'il y a un autre bloc NP à gauche du sujet dans la phrase).

Le résultat final est un ensemble de relations respectant un certain format : le sujet et l'objet sont des blocs NP, et le prédicat doit commencer par un verbe, éventuellement accompagné d'une préposition ou d'autres informations. Chaque relation dispose également d'un score de confiance permettant d'évaluer rapidement sa qualité potentielle. Dans notre travail, nous utilisons ce système pour faire le lien entre l'information présente dans Wikipedia, qui est sous forme textuelle, et celle présente dans DBpedia, sous forme de triplets RDF formels.

CHAPITRE 3 MÉTHODOLOGIE

Ce chapitre présente notre méthodologie de recherche pour répondre aux questions posées en introduction. La première section fournit une brève définition de certains termes utilisés fréquemment dans ce mémoire. La deuxième présente notre réponse au problème de la définition de ce qu'est un domaine de connaissance. La troisième explique notre approche pour évaluer la qualité de DBpedia du point de vue des connaissances conceptuelles liées à un domaine. Enfin, la quatrième fournit notre approche pour améliorer DBpedia au vu des conclusions tirées précédemment. La plupart des informations contenues ici sont présentes dans au moins l'un des deux articles, mais ce chapitre a pour but de donner une vue globale de la méthodologie de recherche tout au long du projet.

3.1 Terminologie

Concept : nous considérons que les entités DBpedia sont soit des *entités nommées* (personnes, entreprises, lieux géographiques…) soit des *concepts*. Ces derniers représentent des idées abstraites, n'ayant pas de réalité physique.

Domaine (de connaissances) : l'ensemble des connaissances liées à une thématique commune. Par exemple, *Mathématiques* ou *Médecine* sont des domaines. Dans notre cas, on s'intéresse uniquement aux *concepts* liés à un domaine, par exemple *Pythagore* ne fait pas partie du domaine *Mathématiques*.

Classe : dans les standards du LOD, une classe représente un ensemble d'éléments ayant des caractéristiques en commun. Informellement, si l'on peut dire « X est un Y », par exemple « Montréal est une ville » ou « Barack Obama est une personne », alors Y est une classe.

Instance : une instance d'une classe est un élément faisant partie de cette classe. Dans l'exemple précédent, Montréal et Barack Obama sont des instances des classes Ville et Personne respectivement.

Ontologie : une ontologie est une structure formelle composée de classes et de propriétés hiérarchisées, indiquant les relations attendues entre les instances de ces classes. Par exemple, on pourrait vouloir indiquer que toute instance de la classe *Personne* a exactement un lieu de naissance, qui doit être une instance de la classe *Lieu*, cette relation étant indiquée par la propriété *lieuDeNaissance*,. On y réfère aussi souvent par le *Schema Level* ou *T-Box* (T pour Terminology), là où les instances représentent l'*Instance Level* ou *A-Box* (A pour Assertion).

Catégorie : dans le premier article, nous utilisons les catégories de Wikipédia. Ces catégories sont des groupes de pages parlant de sujets similaires, organisés en une hiérarchie.

Namespace : dans les deux articles, nous utilisons la notion de namespace. DBpedia est composé de deux parties : une ontologie (T-box) et des instances (A-box). Pour s'y référer plus facilement, nous les appelons respectivement « ontology namespace » et « page namespace ». DBpedia contient également une copie de la structure de catégories de Wikipédia : nous y référons dans notre travail par « category namespace », bien qu'il s'agisse techniquement d'un sous-ensemble du « page namespace ».

3.2 Construction des domaines

Dans l'ensemble du projet, nous nous intéressons aux connaissances liées à un domaine. Concrètement, cela signifie que l'on s'interroge sur la capacité de DBpedia à répondre à des questions du type « Quel est la place de l'algèbre linéaire dans les mathématiques modernes ? » ou « Quelles sont les différentes approches pour le traitement automatique des langues naturelles ? ». La première question que l'on se pose étant donné cette problématique est : comment définir précisément ce qu'est un domaine ? La définition que l'on trouve dans le dictionnaire qui se rapproche le plus de notre sujet est : « Secteur, champ couvert par une science, une technique, etc. »[4]. Ici, nous nous intéressons uniquement à la connaissance conceptuelle, abstraite, de

[4] www.larousse.fr

DBpedia. Par conséquent, nous considérons qu'un domaine est uniquement constitué de concepts, et ignorons donc toutes les entités nommées. Par exemple, nous ne considérons pas que *Microsoft* appartienne au domaine *informatique*. La difficulté est de traduire cela formellement afin de mettre en place des procédés reproductibles de définition d'un domaine.

Notre première approche pour cela, présentée dans le premier article, est de créer une liste de concepts grossière mais de grande taille, en utilisant les catégories Wikipédia. En effet, Wikipédia est organisé autour d'une hiérarchie de catégories, chaque article appartenant à une ou plusieurs catégories. Nous avons sélectionné s*ix* domaines de départ : *Artificial Intelligence, Mathematics, Music theory, Plant taxonomy, Sports science* et *Political science*. Ces domaines ont été choisis avec pour principal but d'être suffisamment variés. Pour chacun de ces domaines, notre point de départ est la catégorie Wikipedia associée. Par la suite, nous explorons toutes les sous-catégories de cette catégorie initiale, et ainsi de suite récursivement jusqu'à un certain seuil. Pour chaque catégorie prise en compte, nous ajoutons toutes les pages y appartenant à notre ensemble de concepts. L'information hiérarchique n'est pas conservée : tous les concepts sont au même niveau, quelle que soit leur catégorie d'origine. Nous obtenons ainsi une liste pour chaque domaine, contenant en moyenne 1900 concepts par domaine.

Cette méthode, bien que relativement efficace étant donné sa simplicité, présente un certain nombre d'inconvénients. Premièrement, on constate rapidement l'apparition de concepts non pertinents, alors que l'on n'a pas encore récupéré tous les concepts pertinents importants. Nous devons donc faire un compromis entre la qualité et la quantité de concepts présents. Étant donné les méthodes d'évaluation que nous appliquons sur ces domaines, il est indispensable de limiter le bruit au maximum, et donc de privilégier la qualité à la quantité. Par conséquent, nos domaines sont incomplets. Le deuxième inconvénient majeur est la présence d'éléments ne représentant pas des concepts. Cela signifie que nos domaines contiennent une quantité non négligeable d'entités nommées. Enfin, le nombre de concepts obtenus varie fortement entre les domaines, allant de 291 pour *Sports science* à 3539 pour *Artificial intelligence*.

Pour ces raisons, dans le second article, qui est une version étendue du premier, nous changeons radicalement d'approche pour définir ces listes de concepts. Il s'agit d'un procédé en deux étapes : création d'une liste initiale, puis extension par annotation sémantique. La première étape de création de la liste de concepts est basée sur un certain type de pages présentes dans Wikipédia, les

pages *Outline of*. Ces pages, exclusives au Wikipédia anglais, présentent un ensemble de pages reliées à un sujet de départ. D'après Wikipédia[5] :

> *Outlines on Wikipedia are stand-alone lists designed to help a reader learn about a subject quickly, by showing what topics it includes, and how those topics are related to each other.*

Par exemple, la page *Outline of mathematics* contient un certain nombre de concepts reliés aux mathématiques comme *Arithmetic* ou *Natural numbers*, organisés par sujet. Ceci correspond tout à fait à notre objectif, ce qui nous permet d'obtenir rapidement une liste de concepts pertinente. Pour cela, nous récupérons, sur la page *Outline of [domain]*, tous les liens hypertexte. Un filtrage est effectué ici afin de ne garder que les éléments qui nous intéressent : nous ignorons par exemple toutes les pages de type *History of…* Cette première étape permet d'obtenir une moyenne de 160 concepts par domaine. Au cours de cette étape, nous avons augmenté le nombre de domaines en y ajoutant 11 nouveaux, choisis aléatoirement : *Business, Construction, Geography, Health sciences, Industry, Literature, Psychology, Religion, Astronomy, Biology* et *Human anatomy*. Le domaine *Plant taxonomy* est également remplacé par *Botany*, car il n'existe pas de page *Outline of plant taxonomy* sur Wikipédia.

Cette approche, bien que beaucoup plus fiable d'un point de vue bruit, présente l'inconvénient de ne fournir qu'un faible nombre de concepts par domaine. Pour cette raison, nous avons ajouté une deuxième étape au procédé. Elle consiste à analyser l'*abstract* (le premier paragraphe de la page Wikipédia) de chaque concept afin d'en retirer les concepts pertinents, en suivant l'hypothèse que ces nouveaux concepts seront également liés au concept de départ. Par exemple, la page *Outline of mathematics* contient le concept *Arithmetic*, dont l'abstract contient « *[…] It consists of the study of numbers […]* ». De cet abstract, le concept *Number* est identifié. Nous l'ajoutons donc au domaine *Mathematics*. Concrètement, ce processus utilise un *annotateur sémantique*, c'est-à-dire un outil identifiant, dans un texte donné en entrée, divers éléments importants. Il peut s'agir de générer des métadonnées, d'extraire les concepts importants (ce qui nous intéresse ici) ou de repérer des mots-clés. En moyenne, nous obtenons 1,5 nouveaux concepts pour chaque concept initial. Il serait théoriquement possible de répéter ce processus, mais nous avons constaté qu'en pratique, une certaine quantité de bruit apparait, c'est-à-dire qu'un concept n'ayant rien à voir avec

[5] https://en.wikipedia.org/wiki/Wikipedia:Outlines

le domaine y est parfois ajouté. Ce bruit est plus faible que dans la première approche mais est tout de même présent. Ceci pose problème, car plus l'on s'approche d'une liste « idéale » contenant tous les concepts qui sont censés s'y trouver, plus il devient difficile d'obtenir les derniers concepts manquants. À l'opposé, le bruit continue à se multiplier exponentiellement, car chaque étape d'expansion sur un concept invalide ajoute de nouveaux concepts invalides au domaine. Pour cette raison, nous avons choisi de n'appliquer qu'une seule itération d'expansion du domaine par annotation sémantique.

3.3 Évaluation de la qualité de DBpedia

L'étape suivant la création du domaine consiste à récupérer toute l'information contenue dans DBpedia sur ces domaines, sous forme de triplets RDF <*sujet, relation, objet*>. Dans le premier article, pour des soucis de simplification et d'espace limité, nous nous sommes focalisés sur les liens sortants, i.e. où le concept de départ est **sujet** du triplet RDF. En effet, ces liens sortants représentent la **description** de ce concept, c'est-à-dire l'information disponible pour le décrire. Dans le troisième article, nous considérons également les liens entrants, qui représentent l'**usage** d'un concept, c'est-à-dire les endroits où il est utilisé pour décrire une autre entité). Ceci nous permet d'avoir une vue d'ensemble de la situation.

Une fois ces triplets récupérés, notre objectif est d'attester si un triplet donné fournit de l'information pertinente, liée au domaine, sur le concept de départ. Par exemple, le triplet <*dbr:Bone rdf:type dbo:AnatomicalStructure*> apporte une information sur ce qu'est un os. À l'opposé, <*dbr:Bone foaf:isPrimaryTopicOf wikipedia-en:Bone*> informe simplement que la page DBpedia *Bone* est extraite de la page Wikipédia *Bone*, n'apportant aucune information conceptuelle. Pour conduire cette évaluation, nous avons travaillé sur les **prédicats** (c'est-à-dire la relation entre le sujet et l'objet, ici *rdf:type* et *foaf:isPrimaryTopicOf*). En RDF, un prédicat donné, par exemple *rdf:type*, est théoriquement toujours utilisé de la même manière, ici pour représenter un lien d'appartenance à une catégorie. Par conséquent, nous avons classé les prédicats en deux groupes : ceux qui permettent d'obtenir de l'information conceptuelle, et les autres. Dans le premier article, nous avons identifié six catégories possibles, trois pour chaque groupe (*DL, Domain* et *CS* étant utiles, *Provenance, Reference* et *Annotation* ne l'étant pas). Parmi ces six groupes, seul le groupe *Domain* est potentiellement bruité, car les cinq autres ont été construits à partir de vocabulaires fiables et établis comme *dcterms, skos* ou *rdf* (voir Section 4.3.3 pour plus de détails).

Dans le second article, nous avons délaissé *Provenance, Reference* et *Annotation* n'apportant pas d'information, et avons constaté que, parmi les trois autres, seuls *DL* et *Domain* étaient utilisés en pratique. Nous nous sommes donc focalisés sur ces deux groupes afin de simplifier le propos.

Une fois ces prédicats catégorisés, l'étape suivante est d'évaluer comment ces groupes de prédicats sont utilisés dans notre ensemble de triplets. Pour cela, nous avons défini trois métriques, permettant d'analyser divers aspects de la présence des groupes. Nous regardons également dans les deux articles la *fréquence* d'apparition des prédicats, c'est-à-dire leur nombre d'apparitions au total. Nous analysons aussi plus en détail l'utilisation des deux catégories de prédicats intéressantes, et enfin, nous calculons la quantité de triplets connectant deux concepts du domaine, et quels prédicats sont utilisés dans de tels liens. Les résultats de ces évaluations se trouvent aux sections 4.4 et 6.4.

3.4 Amélioration de DBpedia par suggestion de nouvelles relations et classes

Dans le second article, nous étendons les conclusions tirées dans le premier sur un plus grand ensemble de test, et suggérons également plusieurs pistes pour améliorer DBpedia. Nous proposons des améliorations de deux types : ajout de nouvelles classes à l'ontologie, et ajout de nouvelles relations dans DBpedia. Les deux sont basées sur l'extraction de relation (ORE). Nous proposons également une méthode alternative basée uniquement sur le contenu de DBpedia, et plus précisément sur le prédicat *dbo:type* pour l'ajout de nouvelles classes.

3.4.1 Amélioration par extraction de relations

L'ORE permet d'extraire d'un texte de départ des relations entre entités, et ce sans aucune information autre que le texte lui-même. Ceci permet d'utiliser ce système sur n'importe quel type de texte, à la seule condition qu'il soit correctement écrit (nous avons rencontré plusieurs erreurs dues à un espace manquant après un point quelque part dans un texte). Les relations sont composées, à l'image des triplets RDF, d'un sujet, d'un prédicat et d'un objet. Cette étape constitue une boite noire pour notre projet : nous fournissons l'*abstract* d'un concept à ReVerb et récupérons un ensemble de relations sur lesquelles nous effectuons notre traitement. Le travail se fait sur l'abstract uniquement car les pages Wikipedia sont inégales dans leur format : certaines sont très

courtes et ne contiennent que l'information essentielle, d'autres sont beaucoup plus longues et fournissent une quantité importante d'information pas nécessairement pertinente. L'abstract, c'est-à-dire les quelques premières phrases de la page, est par convention une courte définition du sujet de la page, et est donc un environnement propice à l'extraction de relations et concepts reliés pertinents.

L'étape suivante est d'associer le sujet et l'objet en langue naturelle à une URI de DBpedia. Pour cela, nous effectuons une normalisation ayant pour but de ramener l'expression obtenue à la forme standard utilisée par les URI de DBpedia. Ceci nous permet de ne considérer que les correspondances exactes (*« exact match »*) pour cette association. Par exemple, le sujet d'une relation extraite « The Milky Way » est normalisé en « Milky_Way », ce qui permet de l'injecter directement dans l'URI pour récupérer l'information DBpedia de la page http://dbpedia.org/page/Milky_Way. Nous n'avons rencontré aucun cas où cette normalisation ne permettait pas d'obtenir une URI valide ou vide.

Par la suite, nous ne conservons que les relations reliant deux concepts du même domaine. Ceci permet d'obtenir 641 relations, utilisant 382 prédicats uniques.

Une fois cet alignement du sujet et de l'objet effectués, l'étape suivante, plus délicate, consiste à identifier le sens du prédicat. Nous avons pour cela pris en compte cinq catégories de relations possibles (excluant la sixième, qui contient les erreurs provenant de l'extracteur ou du texte d'origine) :

- Relations d'équivalence, indiquant que le sujet et l'objet représentent le même concept, par exemple « The term double star is often used synonymously with binary star ».
- Relations d'exclusion mutuelle, indiquant que le sujet et l'objet représentent deux concepts differents, par exemple « Climate is different from weather ».
- Relations d'hypernymie, indiquant que le sujet fait partie d'un groupe représenté par l'objet, par exemple « A fairy tale is a type of short story ».
- Relations d'hyponymie, l'inverse de la précédente : c'est l'objet qui fait partie d'un groupe représenté par le sujet.
- Autres relations. Lorsqu'une relation n'appartient à aucune des catégories précédentes, mais est sémantiquement correcte et a du sens, nous considérons qu'il s'agit d'une relation

spécifique au domaine, par exemple, « Helioseismology is the study of the propagation of wave oscillations ».

- Aucun. Il peut arriver, par exemple lorsque le texte est mal écrit (lorsqu'il manque un espace après un point, faisant que l'extracteur considère les deux mots entourant le point comme un seul), que la relation obtenue n'ait aucun sens. Dans ce cas, nous l'ignorons pour la suite du travail.

Nous avons effectué une classification manuelle des 382 prédicats obtenus parmi ces catégories. Cette évaluation a été effectuée par un vote entre quatre évaluateurs. Les instructions fournies pour cette évaluation sont fournies en annexe. Le détail de la répartition des prédicats parmi les catégories se trouve à la section 6.3.4.3.

La dernière étape consiste à exploiter l'utilisation de ces catégories dans notre dataset afin d'obtenir de l'information. Dans un premier temps, nous examinons les relations extraites et observons que 95% de ces relations (610 parmi 641) n'existent pas dans DBpedia, indiquant que l'ORE est une avenue prometteuse pour améliorer DBpedia. La pertinence et l'exactitude de ces relations reste cependant à déterminer.

Dans un deuxième temps, nous regardons plus spécifiquement les cas où les relations nous permettent d'identifier des liens instance / classe. Ceci nous permet de trouver 36 nouvelles relations entre des entités de DBpedia et une classe de son ontologie.

Enfin, nous étendons l'approche proposée précédemment pour identifier de nouvelles classes : nous partons de l'hypothèse que l'objet d'une relation d'hypernymie (ou, de manière équivalente, le sujet d'une relation d'hyponymie) devrait être une classe. Dans l'exemple fourni, le simple fait que *short story* soit l'objet d'une relation *is a type of* nous indique que ce concept représente une classe potentielle. Partant de là, nous évaluons individuellement chacun de ces candidats afin de déterminer s'il s'agit effectivement d'une classe. 65% des 143 candidats ont été acceptés comme étant des classes, 14% ont été refusés et 30% représentent des cas discutables. Cette précision est relativement faible, mais suffisante pour prouver que l'utilisation d'ORE a du potentiel pour identifier de nouvelles classes. Les instructions données aux évaluateurs lors de l'évaluation des candidats sont données en annexe.

3.4.2 Amélioration par patron utilisant *dbo:type*

Cette méthode est beaucoup plus simple que la précédente. Nous avons constaté que le prédicat *dbo:type* permet d'obtenir des informations pertinentes, car il relie en général une instance à une entité DBpedia qui n'est pas rigoureusement présentée comme une classe (i.e. absente de l'ontologie), mais qui représente tout de même une « classe informelle ». De là, au vu de ce que nous effectuons dans la partie précédente, il est tout naturel de proposer ces « classes » comme d'autres candidats potentiels.

Nous avons donc, parmi les concepts de notre dataset, identifié tous ceux qui sont l'objet d'au moins un triplet utilisant *dbo:type*, soit 539 candidats. Cette fois cependant, nous pouvons d'ores et déjà éliminer certains candidats : en effet, parmi les 539, les 196 apparaissant dans le plus de triplets *dbo:type* représentent 95% du nombre total de triplets *dbo:type* identifiés. Ceci signifie que les 343 autres candidats apparaissent dans très peu de triplets. Étant donné la quantité d'erreurs présentes dans DBpedia de manière générale, ne considérer que les concepts étant l'objet d'un grand nombre de tels triplets nous a paru un moyen simple et efficace de réduire les erreurs, car cela réduit la probabilité que tous les triplets soient erronés. Par la suite, nous ne considérons donc que ces 196 candidats. Comme précédemment, nous évaluons manuellement ces 196 candidats afin de déterminer s'il s'agit effectivement de classes. 112 candidats (57%) sont acceptés, 66 (33%) sont refusés et 18 (9%) sont des cas discutables. La précision est légèrement inférieure par rapport à la méthode présentée dans la section 3.4.1 (57% contre 65%). Les instructions fournies aux évaluateurs lors de l'évaluation de ces candidats sont fournies en annexe.

CHAPITRE 4 ARTICLE 1: ASSESSING THE QUALITY OF DOMAIN CONCEPTS DESCRIPTION IN DBPEDIA

Ludovic Font, Amal Zouaq, Michel Gagnon

Proceedings of 11th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS), 2015

With the increasing volume of datasets on the Linked Open Data (LOD) cloud, it becomes necessary to assess Linked Data quality. This is especially important for DBpedia, which has become a prominent resource on the LOD. In this paper, our aim is to evaluate the quality of the description of domain concepts in DBpedia. Using a data-driven approach on a sample of domain concepts from Wikipedia, we show that a) the resources in our sample are described mainly by facts in DBpedia and seldom refer to the DBpedia ontology; b) DBpedia models very poorly these sample domain concepts at the instance level and schema level; c) very few predicates can be used for inference purposes; and d) very few domain predicates (object properties) are used in the description of domain concepts. This highlights the importance of restructuring the DBpedia knowledge base and including domain knowledge at the schema and instance levels.

4.1 Introduction

Linked Data, the latest evolution for publishing and connecting data over the Web, is a significant movement for the realization of a Web that can "satisfy the requests of people and automated agents to access and process the Web content intelligently" [35]. This evolution is concretized by the development of large knowledge bases such as DBpedia [36], Yago [37] and WikiData [38]. These knowledge bases describe concepts and entities and create links to other available datasets, thus contributing to the emergence of a knowledge graph. In particular, DBpedia defines globally unique identifiers that represent Wikipedia pages/entities. These identifiers can be de-referenced over the Web into RDF[6] descriptions of the pages [39].

[6] Resource Description Framework, a model to store data in the form of triples *Subject; Predicate; Object*.

DBpedia [36] contains RDF triples extracted from Wikipedia infoboxes and categories. These triples constitute a summary of the main facts about the considered Wikipedia page. This dependency on infoboxes poses at least two limitations: i) by definition, an infobox contains much less information than the page itself; and ii) infoboxes are missing in more than 50% of Wikipedia pages [39]. In both cases, we expect that many relevant predicates and attributes will be missing. However, the extent of this limitation is unknown. This issue is far from trivial if we consider a) the definition of DBpedia as a cross-domain knowledge base; b) the objective of representing the whole Wikipedia content as linked data in DBpedia and c) the original aim of the Linked Data cloud: the ability to relate things and infer new knowledge based on interconnected datasets.

Given the central position of DBpedia over the Linked Open Data Cloud (LOD), the richness and nature of these triples become essential for query answering and for the integration of reasoning capabilities when answering SPARQL queries. For example, recommender systems based on DBpedia would benefit from rich semantic links between domain related concepts [40]. This paper targets the evaluation of the RDF description of domain concepts in DBpedia. A concept description is the set of RDF triples having this concept as subject. Domain concepts correspond to a Wikipedia / DBpedia URI that refers to some domain class or topic. By topic we mean "subjects" in a domain D (For instance "Mathematics") that are not usually considered as classes or named entities.

Ideally, at the schema level, this set should constitute the intensional definition of the concept C. By intensional definition, we mean the set of terminological axioms describing C following the definition of intension provided in [41] as a collection of attributes (predicates) describing C. Here a concept can be a class (e.g. db:Mammal) or a topic within a particular domain D. For instance, if the DBpedia concept db:Mathematics contains the triple db:Mathematics rdfs:subClassOf db:Science, then this triple is part of the description of the concept db:Mathematics. Other examples are the concepts db:artificial_intelligence and db:knowledge_representation in the domain of computer science. One would need, if possible, a machine interpretable representation of such concepts in DBpedia that includes semantic links (e.g. rdfs:subclassOf, domain object properties) to other relevant concepts from the same domain. However, as one can notice with a simple check of the DBpedia URI for the examples provided above, concepts' representations seem to be seldom connected to other domain concepts. Thus, query engines are not able to infer that Artificial

Intelligence involves tasks such as planning and knowledge representations, or that Artificial Intelligence uses techniques such as natural language processing and machine learning for instance.

Given that an important number of topics and concepts is not represented in the DBpedia ontology, but exists in the DBpedia knowledge base at the instance or entity level (e.g. Artificial intelligence), another way to assess the richness of the description of these domain topics is to explore their semantic links (predicates) to other resources in DBpedia. This paper presents a study of concepts descriptions over a sample of domains with the objective to assess how domain knowledge is represented in DBpedia and the type of predicates used to describe concepts.

The following section summarizes some related work. Section 3 presents our research methodology, including the research questions, the description of datasets and the evaluation metrics. Section 4 presents our results, and finally section 5 discusses these results and future work.

4.2 Related Work

The evaluation of the quality of DBpedia, and of the Linked Open Data cloud in general, is becoming a timely research topic.

One issue is to determine what exactly the "quality" of a dataset is. Some frameworks such as Sieve [10] or Luzzu [9] have been developed that allow for a flexible definition of quality. For instance, Luzzu [9] is a domain specific language that allows non-programming experts to define quality metrics for the assessment of Linked Open datasets. Similarly, various metrics have been proposed including completeness, relevancy, or consistency as summarized in [5]. These metrics necessitate a gold standard while our approach simply quantitatively analyzes the results to draw some conclusions. A crowd-sourcing approach for the evaluation of DBpedia has also been undertaken [13], which highlights some of DBpedia problems such as broken links, irrelevant or wrong information, or accuracy.

Overall, the main contribution of our approach is the focus on the description of concepts related to a particular domain and the analysis of the predicates used to describe these concepts based on various categories, at the schema and the instance level.

4.3 Research methodology

4.3.1 Research questions

In this work, we attempt to answer the following research questions:

RQ1: How are domain concepts **described** in DBpedia?

Here we investigate how domain concepts are described in the DBpedia knowledge base at the schema level (DBpedia ontology) and at the instance level (DBpedia facts).

RQ2: How are predicates **distributed** in the description of domain concepts in DBpedia?

This research question investigates the distribution and relative frequency of predicates for the description of domain concepts in DBpedia.

4.3.2 Datasets

4.3.2.1 Domain concepts identification in Wikipedia

This section explains our methodology to identify domain-related concepts using Wikipedia categories. In this study, we chose Wikipedia categories as a starting point because they can be used to represent both topics (e.g. Natural language processing) and classes (e.g. Artificial intelligence applications under the category artificial intelligence[7]). Wikipedia categories group together pages around similar topics and may contain sub-categories. This paper defines a domain D as the set of categories and pages that occur under a root domain category C chosen by a domain expert.

We selected several root (domain) categories in Wikipedia: Artificial Intelligence, Mathematics, Music Theory, Plant Taxonomy, Politic Science and Sports Science. Based on these initial categories, we recursively explored all sub-categories and pages as candidate domain concepts, with a depth limit of 3. This depth was chosen empirically as we noticed that deeper levels might

[7] https://en.wikipedia.org/wiki/Category:Artificial_intelligence

include more noise (e.g. "Pythagoreanism" appears in domain "Music Theory", or "Fermentation" in "Sports science" at depth 4).

Table 4.1: Number of Domain concepts

Domain	Domain concepts
Artificial intelligence	3539
Mathematics	3356
Music Theory	2012
Plant Taxonomy	890
Sports Science	291
Political Science	1045
Total	11133

We explored only the Wikipedia categories that correspond to a page with the exact same name, i.e. having a "Main article for this category" in Wikipedia. This filtering was made as we noticed that categories without a main page are usually lists of named entities. Table 1 summarizes the number of concepts extracted in each domain.

From now on, when we use "Music", "Plants", "Sports" and "Politics" instead of the full names, we refer to the domains shown in this table.

4.3.2.2 Data extraction from DBpedia

Let D be a domain and $DC(D)$ the set of concepts extracted in this domain using the procedure described in the previous section. For each concept $cpt \in DC(D)$, we queried its *description* from DBpedia, that is, all the available triples involving *cpt* in their subject:

$$DESC(cpt) = \{\langle cpt, p, o\rangle \mid \langle cpt, p, o\rangle \in DBpedia\}$$

In total, we extracted 267,175 triples from the domain concepts URIs. For each $cpt \in DC(D)$, we queried DBpedia resources in three namespaces: a) the **page namespace** which represents assertions and is found under the URI http://dbpedia.org/resource/*cpt*; b) the **Category namespace**, which represents categories whose URIs are of the form: http://dbpedia.org/resource/category:cpt and c) the **Ontology** namespace under the URI http://dbpedia.org/ontology/*cpt*. The use of the **page** namespace aims at identifying DBpedia concepts' description at the ***instance (assertional) level*** (A-Box), while the ontology namespace describes the ***schema/ terminological level*** (T-Box). Categories can be considered as representing an "informal schema".

Table 4.2 indicates the repartition of DCs among the namespaces. As we can notice, all DCs appear in the "page" namespace, whereas only ~6% appear in the "category" namespace. The DBpedia ontology includes only 0.2% of our DCs. In fact, DBpedia 3.9 is composed of 529 classes, so we cannot reasonably expect to have more than a handful of DCs present in this ontology. However, this simple fact already points to some limitation in the DBpedia knowledge base: domain concepts are very rarely connected to the ontology, thus denoting a poor conceptual schema. The concepts that appear in the ontology are listed in Table 4.3.

Table 4.2: Distribution of DCs in each DBpedia namespace

Data set	Nb. DC Total	Nb. DC Page	Nb. DCs Category	Nb. DC Ontology
A.I.	3539	3539	202	1
Mathematics	3356	3356	157	4
Music Theory	2012	2012	112	5
Plant Taxonomy	890	890	63	9
Sports Science	291	291	25	1
Political Science	1045	1045	143	2
All	11133	11133	702 (6.3%)	22 (0.2%)

Table 4.3: DCs that are present in the DBpedia ontology

Domain	Concept
Artificial Intelligence	Chess Player
Mathematics	Number
	Code
	Distance
	Disease*
Music Theory	Music genre
	Record label
	Note
	Arena
	Song
Plant Taxonomy	Plant
	Moss
	Species
	Cycad
	Flowering plant
	Fungus
	Taxon
	Fern
	Ginkgo
Sports Science	Muscle
Political Science	Law Firm
	Lawyer

* The presence of this erroneous and domain-unrelated concept shows that, even at depth 3 in our exploration of Wikipedia, some noise can occur.

4.3.3 Predicates Groups

Given that DBpedia concepts are described by sets of triples, one way to analyze their description is to categorize the predicates involved in these triples.

Based on a manual inspection of the data sets presented in Table 1, we defined six groups of predicates. First, we exploited several specifications, namely rdf[8], rdfs[9], owl[10], skos[11], dct[12] and prov[13] to identify the list of potential predicates and categorize them according to their respective specification. Then we manually inspected the remaining predicates (i.e. those used exclusively in DBpedia, with the prefix dbo[14]) by looking at several examples of triples. We ignored the dbp[15] predicates since, according to the DBpedia dataset specifications[16], these predicates are noisy.

We identified the following predicate groups:

- ***Provenance predicates*** denote the origin of the concept in Wikipedia;
- ***Reference predicates*** provide a link to an external resource;
- ***Annotation predicates*** provide useful information for a human, such as *rdf:label*.

---

[8] http://www.w3.org/RDF/

[9] http://www.w3.org/TR/rdf-schema/

[10] http://www.w3.org/TR/owl-ref/

[11] http://www.w3.org/2004/02/skos/

[12] Dublin Core Metadata Initiative : http://dublincore.org/

[13] http://www.w3.org/TR/prov-o/

[14] http://dbpedia.org/ontology/

15 http://dbpedia.org/property/

[16] http://wiki.dbpedia.org/data-set-2014, section 4.3.1

- ***Description Logic (DL) predicates*** (such as *rdfs:subClassOf*) are useful for inference and contain most, but not all, of the predicates of the RDF, RDFS and OWL vocabularies (e.g. *rdf:label* is in the annotation group and not in the DL group).
- ***Concept Scheme (CS) predicates*** refer to SKOS and http://www.dublincore.org/documents/dcmi-terms
- ***Domain predicates*** belong to the domain of interest. For instance, the predicate *dbo:symbol* belongs to the domain *Mathematics*.

4.3.4 Evaluation Metrics

We define three metrics based on our needs for this evaluation: *Macro* and *Micro Concept Coverage*, and *Triple Coverage*. These metrics use the **frequency** of a predicate *p*, which is simply the number of times it appears in the description of a concept *cpt*:

$$\mathrm{f}(\mathrm{p},\mathrm{cpt}) = |\{\langle cpt, p, o\rangle \mid \langle cpt, p, o\rangle \in \mathrm{DESC}(\mathrm{cpt})\}|$$

4.3.4.1 Macro Concept Coverage

The macro concept coverage aims at identifying the groups that are rarely used in our data. For a domain D and a group of predicates G, this metric gives the ratio of concepts in D whose description contains at least one occurrence of a predicate from group G. Thus, the group does not "cover" a domain if none of its predicates are used in the descriptions of the concepts in this domain. It is calculated as follows:

$$MCCov(G, D) = \frac{|\{cpt \in DC(D) \mid \exists p \in G, f(p, cpt) > 0\}|}{|DC(D)|}$$

For example, the Reference group has a MCCov of 0.65 in the *Artificial Intelligence* domain. This means that 65% of the concepts in this domain are described by at least one Reference predicate. Equivalently, we could say that Reference predicates are not used at all in the description of 35% of the DCs.

4.3.4.2 Micro Concept Coverage

The Micro Concept Coverage refines the view presented by the preceding measure MCCov, and aims at analyzing the behavior of all predicates of the group. We calculate, for **each predicate**, the

proportion of DCs in which the predicate appears, and then average this value on the cardinality of the group.

$$mCCov(G,D) = \frac{1}{|G|}\sum_{p\in G}\frac{|\{cpt \in DC(D) \mid f(p,cpt) > 0\}|}{|DC(D)|}$$

For example, the DL group has a mCCov of 0.36 in the *Artificial Intelligence* domain. This means that on the average, each DL predicate is used in the description of 36% of the DCs.

With these two metrics, we have a fairly good idea of the overall use of a group: if it has a high MCCov but a low mCCov, we deduce that it contains a few (possibly only one) frequent predicates, and that other predicates are seldom used. On the contrary, if both are high, we deduce that a high number of the group predicates are used in most DCs.

4.3.4.3 Triple Coverage

The *triple coverage* for some group G is used to estimate the importance of G in the description of domain concepts. For each DC, we compute the proportion of triples using predicates of G among all triples of its description. We then average this value on all DCs.

This metric uses the total number of triples in the description of a concept *cpt*:

$$T(cpt) = \sum_{p} f(p,cpt)$$

The Triple Coverage is calculated as follows:

$$TCov(G,D) = \frac{1}{|DC(D)|}\sum_{cpt\in DC(D)}\frac{\sum_{p\in G} f(p,cpt)}{T(cpt)}$$

For example, the DL group has a TCov of 0.42 for the domain *Artificial Intelligence*. This means that on the average, 42% of the triples found in the description of a concept from this domain use a DL predicate.

4.4 Results

4.4.1 Most used Predicates

Table 4 describes the most used predicates for each predicate group. We count the total number of occurrences of each predicate over all concepts and all domains:

$$f(p) = \sum_{D} \sum_{cpt \in DC(D)} f(p, cpt)$$

Table 4.4: Most used predicates per group

Group	Predicate	Frequency	% of group
DL	owl:sameAs	66831	55.0%
	rdf:type	54388	44.7%
	owl:differentFrom	308	0.25%
	dbo:type	70	0.05%
Domain	dbo:class	582	10.8%
	dbo:kingdom	558	10.4%
	dbo:order	491	9.1%
	dbo:division	434	8.1%
	Others	3324	61.7%
CS	dct:subject	36813	99.9%
	skos:broader	32	0.01%
Provenance	foaf:isPrimaryTopicOf	11494	49.7%
	ns:wasDerivedFrom	11432	49.5%
	dbo:wikiPageRedirects	85	0.4%
	others	90	0.4%
Reference	dbo:wikiPageExternalLink	33288	97.0%
	rdfs:seeAlso	1038	3.0%
Annotation	nsX:hasRank	11417	59.4%
	foaf:depiction	3650	19.0%
	dbo:thumbnail	3650	19.0%
	foaf:homepage	419	2.2%
	Others	94	0.4%

As one can notice in Table 4.4, all the groups, except the Domain group, follow the same pattern: they contain 1 to 3 frequent predicates, and very few others. Domain predicates, however, have a fairly low number of occurrences overall with a maximum of 582 occurrences for the predicate *dbo:class*, which constitutes approximately 11% of the triples in the domain group, while the DL group is composed of 55% and 45% of *owl:sameAs* and *rdf:type* predicates respectively. This

difference in predicate frequencies can be partly explained by the fact that domain predicates are, as their name suggest, mostly used in a specific domain.

We also note that the four most frequent predicates (which altogether represent 38,4% of the total) are all in the same domain (Plant). Additionally, our hypothesis is that domain predicates are seldom reused even *in their original domain*. We will verify this hypothesis in the following section.

4.4.2 Macro and Micro Concept Coverage

Fig. 4.1 displays the Macro Concept Coverage (MCCov) results. We notice that four groups have a MCCov of almost 1 in every domain: DL, Provenance links, CS and Annotation, which means that all concepts contain at least one predicate of these groups in their description. CS predicates are used to link concepts to categories: having a MCCov of 1 for this group shows that all DCs are related to at least one category. For Domain predicates, the MCCov is lower than 0.10 in all domains except *Plant taxonomy*:this means that, in these domains, **90% of the DCs do not contain any domain predicate at all**. We analyze the use of the DL group in detail in section IV.4). As for the Reference links, we cannot conclude much except that around 50% of the concepts are linked to a website (other than Wikipedia or a LOD website).

Fig. 4.2 displays the Micro Concept Coverage (mCCov) results. As we can notice, DL, Provenance, Reference and Annotation have a mCCov between 0.2 and 0.5 in all domains. This much lower mCCov, compared to MCCov, is explained by two facts: (1) many predicates that are seldom used decrease the average number of predicate occurrences, and (2) few predicates are used (almost) everywhere (such as *owl:sameAs*).

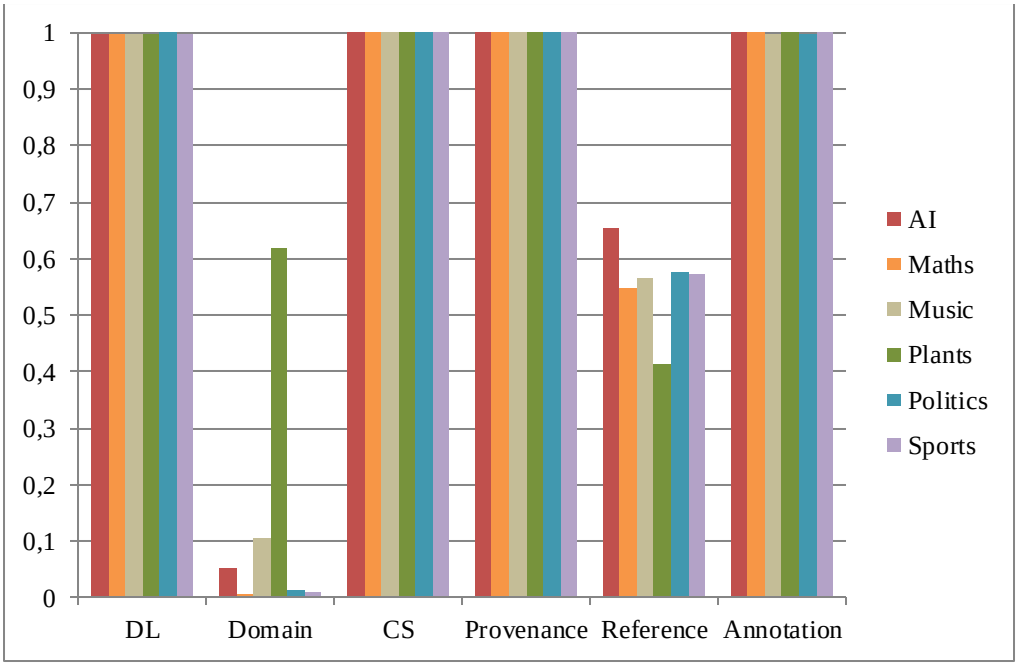

Figure 4.1: MCCov for every group and domain

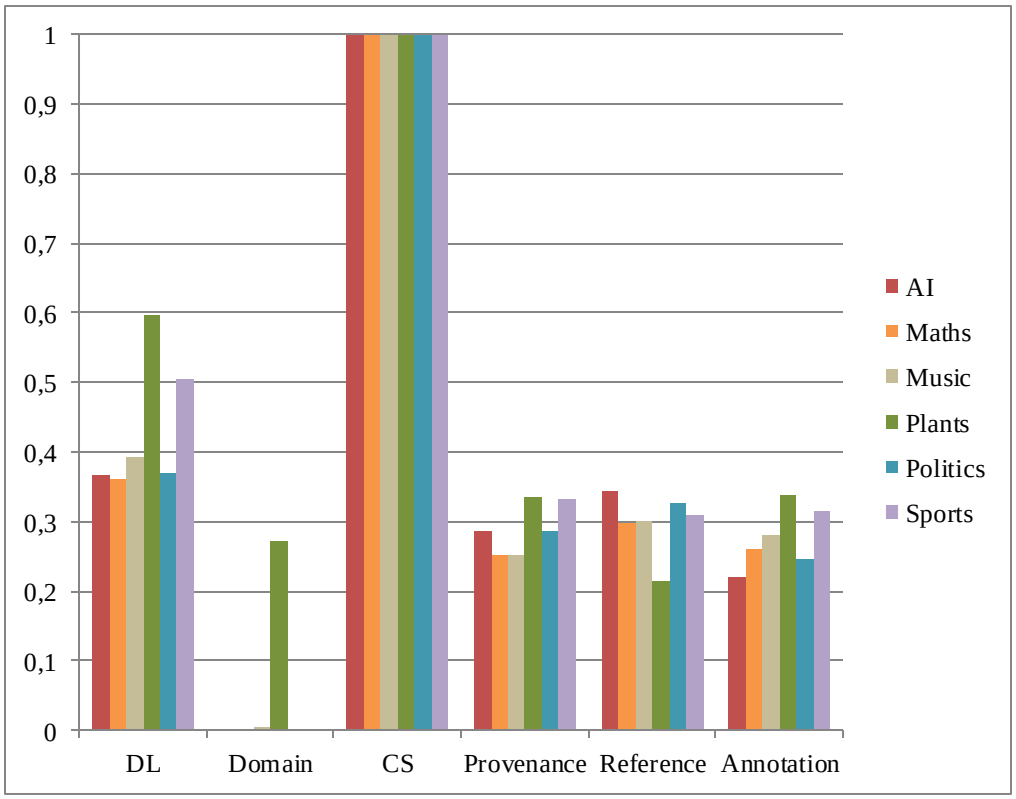

Figure 4.2: mCCov for every group and domain

The CS group obtains an mCCov of 1 for all domains, because the only predicate actually used in this group is *dct:subject*, which indicates the category of the concept.

As for the Domain group, the results are extremely low, ranging from 0.0007 to 0.002, with the exception of the domain *Plant Taxonomy*, whose mCCov is 0.28. Except for the predicates of *Plant taxonomy* that appear in several DCs, most domain predicates are **used very few times in our sample**. For instance, the mCCov in the Artificial Intelligence domain is 0.002, which means that, on average, predicates are used in the description of 0.2% of the 3539 concepts (7.1 concepts). As for Plant taxonomy, several predicates such as *dbo:kingdom, dbo:class* and *dbo:order* are reused across multiple triples (they appear in 44, 40 and 36 concepts, respectively). These results tend to indicate a poor conceptualization (schema) for all domains, with the exception of *Plant Taxonomy*, which is a little bit better.

4.4.3 Repartition of the Predicates among DCs : Triple Coverage

Fig. 4.3 presents the Triple Coverage (TCov) results. In all domains, a high number of predicates found in the concept descriptions pertain to the DL group (40 to 50%).

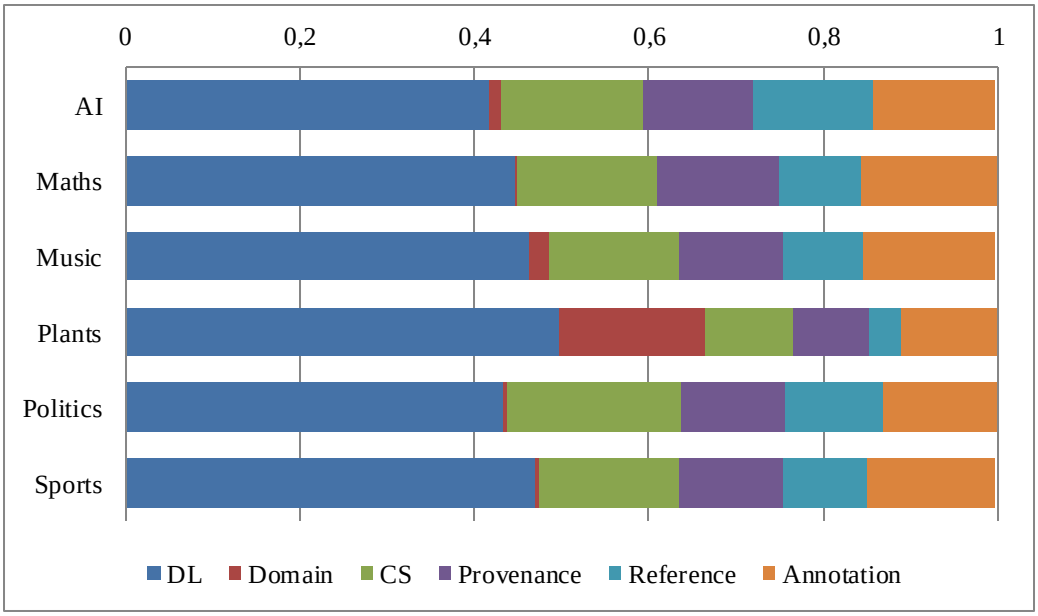

Figure 4.3: TCov for every group and domain

Most of the remaining predicates are distributed among Provenance links, CS, Annotations and Reference links (10 to 20% each). Finally, very few predicates pertain to the Domain group (less than 2%). Once again, Plant Taxonomy is the exception, with 17% of Domain predicates. We noticed previously that Domain predicates occur in only 1% to 10% of DCs (Fig.1, section IV.2), except in the Plant taxonomy domain. This is further supported by the TCov value for the domain

group, which shows that domain predicates always have very few occurrences, unlike DL or CS predicates.

4.4.4 A closer look at the DL group

After looking at these statistics, one essential question remains: Are concept descriptions suitable for automated reasoning? In theory, the groups that can be used for inference purposes are DL and Domain. We already showed that Domain predicates are scarcely used. This section takes a closer look at the DL group.

As we noticed in section IV.1), only two predicates in the DL group have a number of occurrences that is not negligible: *owl:sameAs* and *rdf:type*. We further investigated the triples using *owl:sameAs* and noticed that these triples constitute either a) a linguistic link, i.e. a link to the same page in another language (e.g. "Musical Genre" with "Genre Musical" in fr.dbpedia.org); or b) an external link, i.e. a link to an external URI such as Yago, Wikidata or Freebase (e.g. "Guitar Solo" owl:sameAs http://yago-knowledge.org/resource/Guitar_solo

Table 4.5 shows the proportion of owl:sameAs and rdf:type predicates in the DL group. As the behavior of the DL group does not change much between domains, we report here only numbers calculated for all our data without any domain distinction.

In this table, we notice that 55% of the DL group consists only of owl:sameAs triples, with only 19% referring to Wikidata URIs, 9% to Freebase URIs and 4% to YAGO URIs. rdf:type, which is the other most frequent predicate in our data, constitutes 44.7% of the triples in the DL group.

Table 4.5: Proportion of *owl:sameAs* and *rdf:type* among DL predicates

owl:sameAs					rdf:type
Linguistic	External				
	Wikidata	Yago	Freebase	Other	
0.23	0.19	0.04	0.09	0.002	0.45

We also found that, overall, only 48% of concepts are described by an *rdf:type*. We investigated the nature of these types in our data set. Fig. 4 presents, per domain, the percentage of concepts that are linked, using *rdf:type*, to some other concepts in LOD datasets, such as the DBpedia ontology, Yago or Wikidata. The "Total" column provides the number of concepts that have at least one non-trivial type (i.e. not *owl:Thing* for instance).

As before, the *Plant Taxonomy* domain is quite different from the other ones: a majority (77%) of concepts has at least one type, including 64% in the DBpedia ontology, and 35%, 52% and 63% in Umbel, Yago and Wikidata respectively. However, 23% of the concepts do not possess any (non-trivial) type. In the other domains, around 40% of concepts have a type, among which the majority refers to Yago. Additionally, less than 20% of concepts possess a type defined in the DBpedia ontology in the best case (Sports science). The worst domains in this regard are Mathematics and Political science, with only 1% and 2% respectively.

Overall, the high number of un-typed concepts (23% in Plant Taxonomy, around 60% in the other domains) in the data set, and their poor domain description, point to low inference capabilities in our DBpedia sample.

We also investigated the concepts that appear as classes in the DBpedia ontology (22 concepts in our data). Let us call them the *ontological concepts*. Here only the DL and Domain groups are of interest. We noticed that all concepts are described by *rdf:type* to indicate that they are instances of *owl:Class*, and *rdf:subClassOf* to create the class hierarchy.

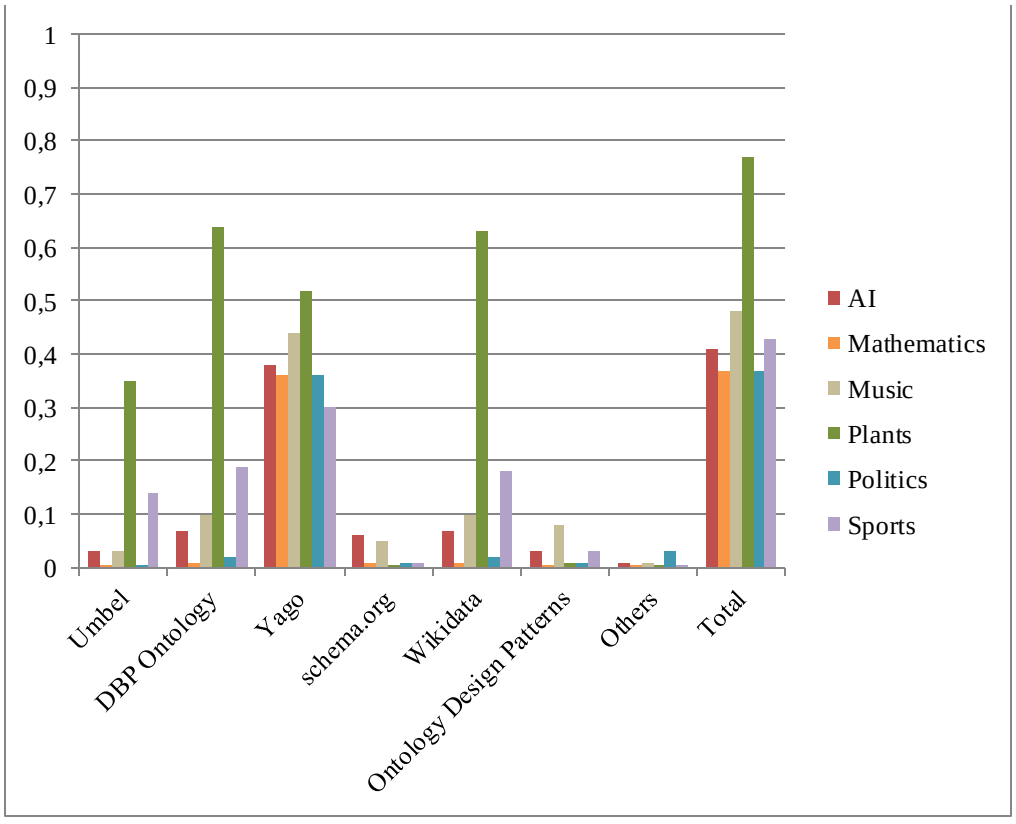

Figure 4.4: Proportion of concepts that have a type in various LOD datasets

To further extend our analysis on the concept types, we investigated the predicates for which the domain or range is specified (with *rdfs:domain* or *rdfs:range)* and whose value is one of our ontological concepts. In these cases, the type of the concept involved in the domain or range of these predicates is implicit and can be determined through inference. For instance, we have the triples ⟨*dbo:musicSubgenre rdfs:domain db:Music_genre*⟩ and ⟨*dbo:musicSubGenre rdfs:range db:Music_genre*⟩. If we find a triple ⟨*db:Industrial_dance dbo:musicSubGenre db:Electro-industrial*⟩, we can infer that both "Industrial dance" and "Electro-industrial" are "Music genres".

Table 4.6 displays these predicates, which involve only 7 of our ontological concepts. Overall, we could infer indirect types for only 0.68% of concepts (76 among 11,113), using 0.17% of triples (466 among 267,175). Furthermore, these triples only use one of the following predicates: *dbo:derivate*, *dbo:musicFusionGenre*, *dbo:musicSubgenre* and *dbo:stylisticOrigin*. Therefore, the only inferred type in our sample would be "Music Genre".

Table 4.6: Usage of the concepts found in DBpedia ontology in the domain or range of some predicates

Domain	Concept	Predicates using the concept	
		As domain	As range
A.I.	Chess player	dbo:elo	
		dbo:eloRecord	
Music	Arena		dbo:homeArena
	Music genre	dbo:derivative	
		dbo:musicFusionGenre	
		dbo:musicSubgenre	
		dbo:stylisticOrigin	
	Song	dbo:trackNumber	
Plants	Taxon		dbo:subfamily
			dbo:superfamily
			dbo:taxon
Politics	Law firm	dbo:numberOfLawyers	
		dbo:numberOfOffices	
Sports	Muscle	dbo:origo	

4.4.5 Interlinking between DCs

As we explained in the introduction, a high linkage in a given domain can provide richer querying capabilities. This section presents the percentage of interlinks **among concepts in the same domain**, i.e. triples where the subject and the object are DCs.

Table 7 shows the number of interlinks in each domain. Here, we included not only the concepts discussed previously (concepts related to a Wikipedia page and concepts from the ontology), but also the concepts that correspond to Wikipedia categories. The percentage corresponds to the ratio of existing interlinks over the maximal possible number of interlinks (which corresponds to the case where every concept is linked to all other ones). This later value is computed as follows:

$\frac{n(n-1)}{2}$, where n is the number of DCs.

Table 4.7: Number of interlinked DCs

Domain	Interlinks	DCs	Interlinks per DC	% of all possible links
A.I.	669	3539	0.19	0.01%
Mathematics	1078	3356	0.32	0.02%
Music theory	565	2012	0.28	0.03%
Plant taxonomy	5609	890	6.30	1.42%
Political science	679	1045	0.65	0.12%
Sports science	44	291	0.15	0.10%

As before, the values are very different from one domain to another: Plant taxonomy has the highest number of interlinks, on average 6.3 per DC, further indicating that it is an example of a better-organized domain, suitable for reasoning. In the other domains, DCs are poorly interlinked: on average, there are 0.15 (min) to 0.65 (max) interlinks per DC, depending on the domain. This is even more visible when we consider the maximum number of interlinks: the ratio of existing links ranges from ~0.01% to ~0.1%.

We investigated the type of predicates used for these links. This information appears in Fig. 4.5. Without surprise, we notice that Domain predicates are mainly used for interlinking in the Plant taxonomy domain. We also notice that Reference links and Provenance links are seldom used: less than 7% for Reference links (entirely with *rdfs:seeAlso* predicates) and less than 0.6% for Provenance links (entirely erroneous *dbo:wikiPageRedirects*). In some domains, DL predicates are used frequently (67% in Sports science). However, given the small number of interlinks in general (as indicated in the last column), these numbers are not very conclusive.

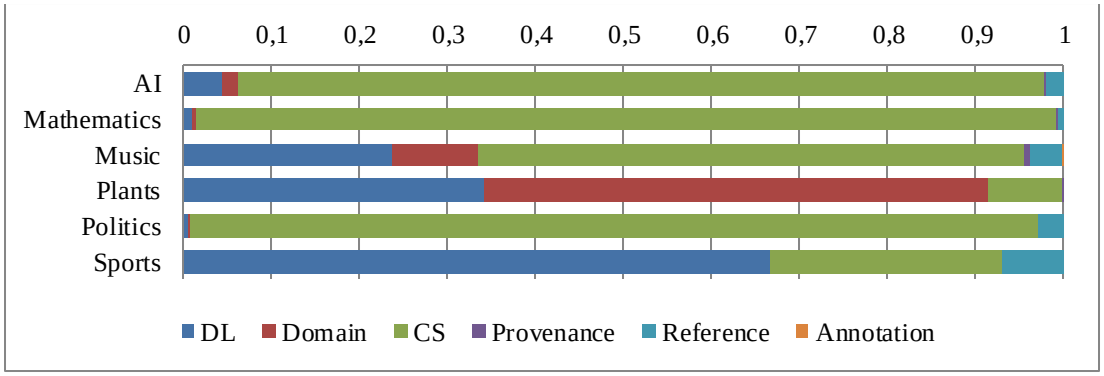

Figure 4.5: Predicate groups used to interlink domain concepts

4.5 Discussion and Conclusion

Making the Web a global knowledge space requires quality linked data. In particular, DBpedia, which represents one of the hubs in the linked data cloud, should include *rich* concepts descriptions, both at the schema level and at the instance level. In particular, the nature of the predicates used in the description of domain concepts is important for semantic search. Some groups of predicates, such as the DL or Domain groups are of particular importance for *querying* and *reasoning*.

In this paper, we addressed two research questions:

RQ1:How are domain concepts described in DBpedia?

Our answer to the first question is that domain concepts are poorly described and poorly interconnected. They are even more poorly related through DL predicates. Except in one domain (Plants), at least 54% of concepts are un-typed when considering both direct *rdf:type* links and indirect types. In addition, a high proportion (ranging from 81% to 99%) of concepts are not linked to the DBpedia ontology, and very few domains contain domain predicates that occur frequently among the triples of the domain.

RQ2: How are predicates distributed in the description of domain concepts in DBpedia?

Regarding the second question, we can see that, globally, DCs descriptions are mainly constituted by generic triples (such as linguistic variations, or links to Wikipedia categories) and very few triples specific to the domain. As we noticed, most of these generic triples cannot be used for inference purposes.

At this time when serious efforts are envisaged to restructure the DBpedia ontology by the research community, this paper sheds the light on a potential weakness in the representation of domain concepts in DBpedia that would need to be resolved, especially if we consider DBpedia as an ontology across domains. To the best of our knowledge, this study is the first to focus on the evaluation of domain concepts representations in DBpedia.

Our results show that we need to increase the linkage between concepts (and instances in general), and the DBpedia ontology. Some methods such as [12] seem to provide interesting perspectives in this respect.

There are some limitations to our study. Our experiment is run on a limited sample of six domains. The same methodology would have to be repeated on several other domains, or on the whole Wikipedia, to confirm the results of this study. This will be the focus of our future work along with the automatic extraction of domain predicates from Wikipedia.

4.6 Acknowledgments

This research was supported by the Royal Military College of Canada Sabbatical Fund and the Natural Sciences and Engineering Research Council of Canada (NSERC) Discovery Grant Program.

4.7 References

[5] A. Zaveri, A. Rula, A. Maurino, R. Pietrobon, J. Lehmann, et S. Auer, « Quality assessment for linked data: A survey », *Semantic Web*, vol. 7, nº 1, p. 63-93, 2015.

[9] J. Debattista, C. Lange, et S. Auer, « Luzzu Quality Metric Language -- A DSL for Linked Data Quality Assessment », *ArXiv150407758 Cs*, avr. 2015.

[10] P. N. Mendes, H. Mühleisen, et C. Bizer, « Sieve: Linked Data Quality Assessment and Fusion », dans *Proceedings of the 2012 Joint EDBT/ICDT Workshops*, New York, NY, USA, 2012, p. 116-123.

[13] A. Zaveri *et al.*, « User-driven Quality Evaluation of DBpedia », dans *Proceedings of the 9th International Conference on Semantic Systems*, New York, NY, USA, 2013, p. 97-104.

[35] T. Berners-lee, D. Connolly, L. Kagal, Y. Scharf, et J. Hendler, « N3Logic: A Logical Framework for the World Wide Web », *Theory Pr. Log Program*, vol. 8, nº 3, p. 249-269, mai 2008.

[36] J. Lehmann *et al.*, « DBpedia–a large-scale, multilingual knowledge base extracted from Wikipedia », *Semantic Web*, vol. 6, nº 2, p. 167-195, 2015.

[37] F. M. Suchanek, G. Kasneci, et G. Weikum, « Yago: A Core of Semantic Knowledge », dans *Proceedings of the 16th International Conference on World Wide Web*, New York, NY, USA, 2007, p. 697-706.

[38] D. Vrandečić et M. Krötzsch, « Wikidata: A Free Collaborative Knowledgebase », *Commun ACM*, vol. 57, nº 10, p. 78-85, sept. 2014.

[39] A. P. Aprosio, C. Giuliano, et A. Lavelli, « Extending the Coverage of DBpedia Properties Using Distant Supervision over Wikipedia », dans *Proceedings of the 2013th International Conference on NLP & DBpedia - Volume 1064*, Aachen, Germany, Germany, 2013, p. 20-31.

[40] G. Piao et J. G. Breslin, « Measuring Semantic Distance for Linked Open Data-enabled Recommender Systems », dans *Proceedings of the 31st Annual ACM Symposium on Applied Computing*, New York, NY, USA, 2016, p. 315-320.

[41] « Ontology Learning from Text », dans *Ontology Learning and Population from Text*, Springer US, 2006, p. 19-34.

CHAPITRE 5 ARTICLE 2: ASSESSING AND IMPROVING DOMAIN KNOWLEDGE REPRESENTATION IN DBPEDIA

Ludovic Font, Amal Zouaq, Michel Gagnon

Submitted to the Open Journal of Semantic Web (OJSW), 2016

With the development of knowledge graphs and the billions of triples generated on the Linked Data cloud, it is paramount to ensure the quality of data. In this work, we focus on one of the central hubs of the Linked Data cloud, DBpedia. In particular, we assess the quality of DBpedia for domain knowledge representation. Our results show that DBpedia has still much room for improvement in this regard, especially for the description of concepts and their linkage with the DBpedia ontology. Based on this analysis, we leverage open relation extraction and the information already available on DBpedia to partly correct the issue, by providing novel relations extracted from Wikipedia abstracts and discovering entity types using the *dbo:type* predicate. Our results show that open relation extraction can indeed help enrich domain knowledge representation in DBpedia.

5.1 Introduction

Linked Data, the latest paradigm for publishing and connecting data over the Web, is a significant step towards the realization of a Web that can "satisfy the requests of people and automated agents to access and process the Web content intelligently" [35]. This evolution is concretized by the development of large knowledge bases such as DBpedia [42], Yago [37] and WikiData [38]. These knowledge bases describe concepts and entities and create links to other available datasets, thus contributing to the emergence of a *knowledge graph*. In particular, DBpedia defines globally unique identifiers that represent Wikipedia pages/entities. These identifiers can be de-referenced over the Web into RDF descriptions of the entities [39]. These RDF descriptions are composed of triples of the form ⟨*s,p,o*⟩, where *p* represents a relation between entities *s* and *o* (for example, we find the triple ⟨*dbr:Sun rdf:type yago:Star109444100*⟩ in DBpedia's description of the entity *dbr:Sun*). In our previous work [1], we have established that DBpedia lacks terminological knowledge (T-box), especially for domain knowledge, despite its rich A-box (instance level). We highlighted some quality issues in the *description of domain concepts*, on a small subset of DBpedia, and we demonstrated a lack of linkage between the DBpedia ontology and the knowledge base. We showed that this lack of linkage is especially true for resources that describe a *domain*

concept, such as *planet*, *village* and *integer* (respectively in the domains of astronomy, geography and mathematics). Without a correct and reliable schema, instances are of limited interest, especially when dealing with big data: it becomes difficult or impossible to detect incoherencies, to reason, or to answer complex queries that go beyond stated triples. In the case of DBpedia, the T-box (schema level) is represented by an ontology that is manually created by the community. This manual work ensures its quality. However, a good linkage between the T-box and the A-box is also paramount to ensure DBpedia quality and its knowledge inference capabilities.

In this paper, we first extend the quality assessment conducted in our previous work [1] by 1) studying 11 new domains, including 8 chosen randomly; 2) using semantic annotation to further extend these domains; 3) evaluating the *usage* along with the *description* of domain concepts and their linkage to an ontological schema. In this new quality assessment, we confirm the lack of important triples in the description of domain concepts and the poor linkage among domain concepts in general, and with the ontology in particular, even in domains that are "well represented" in the ontology. Second, we propose a solution to help alleviate these issues using semantic annotation [43] and open relation extraction (ORE) [26]. In this work, we use ReVerb [30], one of the available ORE tools, to extract relations from Wikipedia abstracts. Each relation is a triple, much like an RDF triple, except that its elements are not URIs, but instead words or groups of words extracted from text. We associate both the subject and object to DBpedia URIs using a semantic annotator, and classify relations into groups, each corresponding to several possible RDF predicates.

Overall, we attempt to answer the following research questions:

Q1: How are domain concepts described in the DBpedia knowledge base, i.e. what are the links relating a concept to other DBpedia concepts (*describing* the concept), both at the schema level (DBpedia ontology) and at the instance level (DBpedia facts)?

Q2: How are domain concepts used in the DBpedia knowledge base, i.e. what are the links relating DBpedia concepts to domain concepts (*using* these domain concepts), both at the schema level (DBpedia ontology) and at the instance level (DBpedia facts)?

Q3: What type of predicates appear in the description and usage of domain concepts, and which of them can be used for inferring domain knowledge?

Q4: Can we enhance DBpedia, by extracting novel relations between domain concepts, and by identifying potential new classes, using open relation extraction on Wikipedia abstracts?

The following section describes the state of the art in quality assessment and concept and relation identification. Section 3 presents an overview of our methodology. We describe the dataset used in our experiments in Section 3 and present our results in Section 4. Section 5 presents our work using open relation extraction, and finally, Sections 6 and 7 discuss our findings before a conclusion.

5.2 Related Work

General Linked Open Data quality. In the first part of this paper, we provide an analysis of the quality of DBpedia for the description of domain knowledge. Several research works have been performed to assess the quality of linked open datasets in general. The usual consensus about the quality of a dataset is its "fitness for use" [4]. In our case, it means "fitness for finding and using knowledge related to a domain". More specifically, when it comes to linked open data, several quality factors have been established: Bizer points out that quality must be assessed according to the task we want to accomplish, provides 17 quality dimensions and related metrics organized in 4 categories [7]. Later, Zaveri et al. provide an updated and extensive list of available metrics [5]. Among this list, our work can be related to aspects of the metric "detection of good quality interlinks". However, the fitness of DBpedia for domain knowledge inference is a very specific problem, and does not fall into any of the established categories, hence our need to introduce novel metrics in this paper. Other quality factors have been defined ([8], [9]), and some frameworks exist to assess the quality of a given dataset. For instance, Luzzu provides a framework customizable by domain experts [9] and Sieve [10] provides ways to express the meaning of "quality" for a given dataset and a specific task. To the best of our knowledge, there is not any other work which focuses on the quality of domain knowledge representation in DBpedia.

DBpedia quality. The DBpedia knowledge base is a huge dataset containing information on many domains ([12], [13]). However, the current method to automatically extract DBpedia data from Wikipedia is based mostly on infoboxes [12]. Even though this method has obvious advantages in terms of automatization and ensure wide coverage, it also poses some issues. According to a user-driven quality evaluation done by Zaveri et al. [13], DBpedia has indeed quality problems (around 12% of the evaluated triples have issues), that can be summarized as follows: incorrect/missing

values, incorrect data types and incorrect links. Kontostas et al. [14] provide several automatic quality tests on LOD datasets based on patterns modeling various error cases, and detect 63 million errors among 817 million triples. Mendes et al. [10] also point out issues in completeness, conciseness and consistency in DBpedia. In our previous work [1], we showed that domain concepts are often poorly described in DBpedia. We also pointed out at the low number of concepts with a (rdf) type, which is a crippling problem for the knowledge inference capabilities of DBpedia. All these issues can take origin in the extraction framework of DBpedia, the mappings wiki (which is used to create automatically the DBpedia triples), or Wikipedia itself. Some efforts have been made to locate and fix errors in DBpedia, and the Linked Data in general, using crowdsourcing approaches [15]. A crowdsourcing approach could be applied to domain knowledge quality assessment in DBpedia, but, given the size of DBpedia, our goal is to explore automatic methods for such a task.

Semantic annotation: semantic annotation consists in tagging important words or groups of words in a text (entity mentions) in order to generate metadata. This process covers several aspects of text comprehension, such as named entity recognition [16], concept identification [17], sentiment analysis [18], or relation extraction ([18], [25], [44]). The efficiency of these tools depends on many factors, such as the task, the type of text, and the number of texts available in the corpus [20]. In this paper, given a source concept's abstract, we exploit the concept identification capabilities of semantic annotators to measure, for a given domain concept, the coverage of the Wikipedia abstract by its DBpedia RDF description and to identify concepts that should appear in relation to this domain concept.

Open relation extraction: Introduced by Banko et al. [26], open information extraction (OIE) is a paradigm to extract a large set of relational tuples without requiring any human input. We have witnessed in the past decade the development of several open relation extractors ([30]–[32]), and some concrete uses are emerging, such as reading news feed to quickly detect economic events [45]. The open relation extraction has recently witnessed improvements based on the usage of external sources from the Web [28] and joint inference [29]. In our work, we use the ORE capabilities to bridge the gap between the textual knowledge of Wikipedia and the formal RDF relations in DBpedia. For this task, we used the ReVerb system [30].

5.3 Research Methodology

5.3.1 Definitions

In this section, we define the terminology used in this paper.

Domain: A domain is, informally, "*A specified sphere of activity or knowledge*" [17]. In our approach, a domain D is a set of concepts that are all related to a particular subject or field: for instance, the domain "*Mathematics*" contains concepts such as "*Geometry*" and "*Algebra*".

Domain concept: In the Linked Data standards, knowledge is stored in the form of RDF triples *⟨Subject, Relation, Object⟩*. The subject and the object are URIs that can represent either concepts (such as *mathematics*) or named entities (such as *Canada*). *Domain concepts* can represent a class of domain objects, like *Integer* or *Planet*, that are usually defined by restrictions on properties in a formal ontology. They can also represent instances, such as *Saturn*, which is a specific entity in the domain of astronomy, or what is usually called a *topic* or *subject*, such as *Algebra*, in the domain of mathematics.

Concept description: A concept description contains all the triples that comprise the concept in the subject position.

Concept usage: A concept usage contains all the triples that use the concept in the object position.

In DBpedia, each concept can be represented in one or both of the following two namespaces:

a) The **resource namespace** represents assertions, and corresponds to the **instance (assertional) level** (A-Box). Investigating this namespace allows us to identify whether concepts are typed, i.e. whether they are related to some ontology, and whether these concepts are related to other concepts through domain-related object properties. Having a concept in this namespace (with an URI of the form http://dbpedia.org/resource/*⟨concept_name⟩*, also abbreviated as *dbr:⟨concept_name⟩*) means that the concept also has a corresponding Wikipedia page (whose location is *http://wikipedia.org/page/⟨concept_name⟩*).

---

[17] According to the Oxford dictionary : http://www.oxforddictionaries.com/definition/english/domain

b) The **ontology namespace** represents all concepts that have an URI of the form *http://dbpedia.org/ontology/⟨concept_name⟩*, also abbreviated as *dbo:⟨concept_name⟩*. This namespace describes the **schema / terminological level** (T-Box). Unlike the resource namespace, concepts in the DBpedia ontology are not specifically associated with a Wikipedia page and are supposed to represent classes.

5.3.2 Approach overview

In this section, we give an overview of our methodology, which consists of four steps. The first three steps concern the extraction of domain concepts from DBpedia, which will be analyzed to determine how well they represent the domain. In the fourth step, we evaluate the potential of open relation extraction to enrich the representation of domain concepts in DBpedia. These steps are the following ones:

Initial dataset extraction: In this work, we use Wikipedia Outline pages[18] to identify domain concepts. Such pages provide numerous concepts related to the domain of interest. For instance, the page "Outline of mathematics" contains links with mathematical concepts organized by subject (the subject "Space" contains the concepts "Geometry" and "Topology" for instance)

Domain expansion: Because of the low number of concepts obtained in the initial dataset extraction step, we expand this set using the Wikipedia abstracts of these domain concepts. Our hypothesis is that the most important concepts present in the abstract are **also part of the domain and should be represented (as objects) in the description of their source concept**. Thus, each source domain concept should be directly related to the concepts identified in its abstract (thereafter called related concepts).

Data extraction from DBpedia: The next step is to retrieve all the triples in the description or usage of domain concepts, i.e. all triples containing one of the previously identified concepts as subject or object. Unlike our previous study, where we focused exclusively on the description of concepts, we also examine whether the usage of a concept follows the same trend as its description.

---

[18]) According to the definition given by Wikipedia: "Outlines on Wikipedia are stand-alone lists designed to help a reader learn about a subject quickly, by showing what topics it includes, and how those topics are related to each other"

Open relation extraction: In the last step, we exploit the information contained in the abstracts of domain concepts to identify predicates between the source domain concept and its related concepts and then compare this information with the description of the concept in DBpedia. The extracted relations are either used to confirm the existing links in DBpedia or to learn new predicates.

5.3.3 Dataset

In this section, we explain in details the first three steps, which result in a dataset of domain concepts and predicates that are analyzed in Section 5.4.

5.3.3.1 Domain concepts identification in Wikipedia

Our list of domains (see Table 5.1) contains nine domains selected manually, with the objective to select fields as diverse as possible, and eight domains chosen randomly among all the "outline of" pages of Wikipedia.

Table 5.1. Domains List

Selection	Domains
Manual	Artificial intelligence ; Mathematics ; Botany; Astronomy ; Biology ; Human anatomy ; Music theory ; Political science ; Sports science
Random	Business ; Construction ; Geography ; Health sciences ; Industry ; Literature ; Psychology ; Religion

To identify domain concepts, we extracted all relevant hyperlinks from their associated outline pages. We performed some filtering to remove the obviously 'non-conceptual' pages (e.g. pages describing named entities) using ad hoc rules. Some sections and hyperlinks were systematically removed such as "*List of…*" (This kind of hyperlink is always used to list entities, and not concepts, e.g. "List of publications", "List of researchers"…), "*Table of…*", "*History*" sections, "*External links*" sections, Links describing a country or nationality (e.g. "Greek mathematicians") or named entities (persons, organizations, books…).

Table 5.2 shows the number of concepts obtained at the end of this step. On the average, we extracted about 160 concepts per domain (with a median of 97) with the richest domains being *Geography, Astronomy* and *Human anatomy*.

Table 5.2. Number of concepts per domain based on the "outline of" pages

Domain	Number of concepts	Domain	Number of concepts
A.I.	120	Health sciences	105
Mathematics	60	Industry	100
Botany	85	Literature	139
Music theory	63	Psychology	91
Political science	59	Religion	92
Sports science	99	Astronomy	315
Business	92	Biology	97
Construction	66	Human anatomy	870
Geography	251	**Total**	**2704**

5.3.3.2 Domain concept extraction using semantic annotation

As we can observe in Table 5.2, the number of concepts extracted from the Outline pages is quite low. For this reason, we expanded the initial set of domain concepts using a semantic annotator. A semantic annotator is a tool that takes raw text as input and identifies segments in the text that represent keywords, concepts or named entities. For each concept in the initial set, we processed its abstract with the Yahoo Content Analysis[19] semantic annotator to obtain the "important concepts". For instance, let us consider the abstract of the concept "Handwriting recognition", where the concepts detected by the semantic annotator are indicated in boldface:

Handwriting recognition *(or HWR) is the ability of a computer to receive and interpret intelligible handwritten input from sources such as paper documents, photographs, touch-screens and other devices. The image of the written text may be sensed "off line" from a piece of paper by optical scanning (****optical character recognition****) or* ***intelligent word recognition.***

We hypothesize that those concepts are part of the **same domain** as the initial concept. We included those novel concepts in their respective domain. Table 5.3 provides the number of concepts in each domain after the expansion step, in the resource and ontology namespaces.

In total, we obtained 6834 domain concepts with a page in the *resource* namespace. We can notice that very few of these concepts are represented as classes in the DBpedia ontology.

[19] https://developer.yahoo,com/contentanalysis

Table 5.3. Number of concepts per domain after expansion

Domain	Number of concepts	
	Resource	Ontology
A.I.	352	0
Mathematics	154	0
Botany	153	1
Music theory	188	3
Political science	110	3
Sports science	245	6
Business	264	5
Construction	151	5
Geography	585	37
Health sciences	244	3
Industry	261	1
Literature	342	5
Psychology	251	2
Religion	206	1
Astronomy	880	8
Biology	350	11
Human anatomy	2098	9
All	**6834**	**100**

5.3.3.3 Data extraction from DBpedia

We ran a series of SPARQL queries to extract DBpedia triples that refer to our domain concepts along with a predicate of interest. Predicates of interest include:

Description Logic (DL) predicates, which are useful for inference, such as *rdfs:subClassOf* or *rdf:type*, and contain most of the predicates of the RDF, RDFS and OWL vocabularies. We also included the predicate *dbo:type* in this group as we observed that its usage is similar to *rdf:type*.

Domain predicates, which belong to the domain of interest and generally correspond to object properties. For instance, the predicate *dbo:symbol* belongs to the domain *Mathematics*. The most used predicates of this group in our dataset are *dbo:genre*, *dbo:country* and *dbo:class*.

Typically, we expect *DL* predicates to provide structural and domain-independent links (*Planet* ***rdfs:subClassOf*** *Astronomical_object*), whereas *domain* predicates provide domain links (*Planet* ***dbo:orbits*** *Star*).

More specifically, let D be a domain and DC(D) the set of concepts in this domain and P the set of predicates of interest, i.e. $P = DL \cup Domain$. For each concept $c \in DC(D)$, we queried its description and its usage from DBpedia, that is, all the available triples involving c in their subject or object, respectively:

$$DESC(c) = \{\langle c, p, o\rangle \mid \langle c, p, o\rangle \in DBpedia, p \in P\}$$

$$USE(c) = \{\langle s, p, c\rangle \mid \langle s, p, c\rangle \in DBpedia, p \in P\}$$

We refer to the first set (*DESC)* as the *description mode*, and the second (*USE*) as the *usage mode.* In total, we extracted 1,259,689 triples, distributed between namespaces and modes (description, usage) as shown in Table 5.4.

Table 5.4. Distribution of the extracted triples among namespaces and modes

Namespace	Nb. Triples	
	Description	Usage
Resource	146,016	650,773
Ontology	329	462,571
Total	146,345	1,113,344

Here, we can already notice that triples are not equally distributed: the *usage* mode contains approximately 7.5 times more triples that the *description* mode. This difference is even more noticeable in the *ontology* namespace, with more than 1400 times more triples in the *usage* than in the *description*. This is consistent with the fact that the ontology is supposed to be widely **used** in the DBpedia knowledge base, but only **described** with few other elements of the ontology. An example of such a descriptive triple is [*dbo:Galaxy* ; *rdfs:subClassOf* ; *dbo:CelestialBody*].

5.4 Analysis of Domain Concepts in DBpedia

In this section, we assess the quality of the representation of domain concepts (description and usage) in DBpedia. In our analysis, we consider that the most important characteristics of a domain for knowledge inference purposes are the following ones: domain concepts should be described by

triples that relate them to other domain concepts in DBpedia, and these related concepts should represent classes from the ontology and concepts (instances) of the same domain. Subsections 4.1 to 4.3 present the three metrics used to analyze the *DL* and *domain* predicates (and hence triples) in the dataset. Subsection 4.4 presents a finer analysis of the *DL* group.

5.4.1 Predicates' global frequency

In this first step, our goal is to obtain global results to determine how the triples are distributed among namespaces, modes, and domains.

Given a predicate p and a concept c, we define the **frequency** $f(p,c)$ as the total number of triples involving p and c, either in the description or in the usage of c, i.e. <*s*, *p*, *c*> and <*c*, *p*, *o*>. By extension, we also define $f(G,c)$, the frequency of a group (DL, Domain, as defined in Section 3.4.3) *G* for a concept *c*, as the sum of the frequencies of all predicates of *G* for *c*, and the **global frequency of *G*** by the sum of $f(G,c)$ on all the concepts of our dataset. For instance, the *DL* group has a global frequency of 136,605 in the description mode. This means that 136,605 triples that describe a concept in our dataset use a *DL* predicate.

Figure 5.1 shows the distribution and global frequency of predicates' groups for both modes in the resource namespace. The ontology namespace statistics are not shown, since all the predicates in the description or usage of a concept in this namespace belong to the *DL* group.

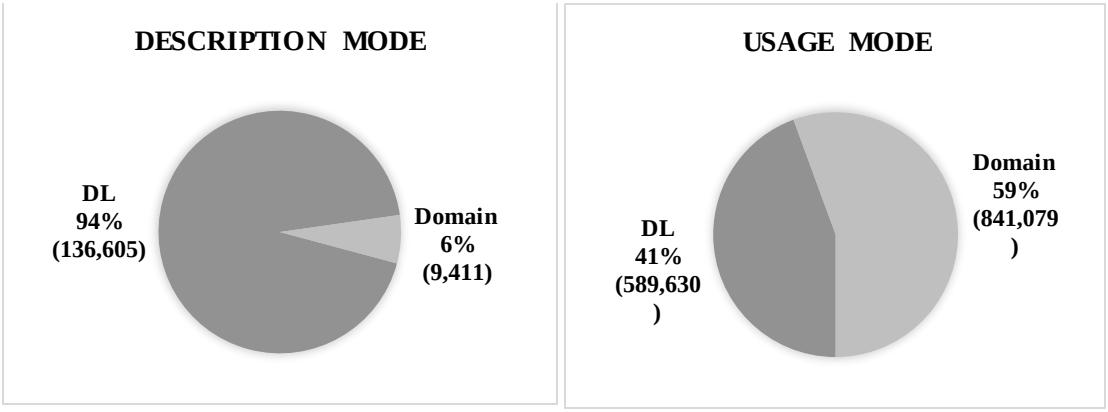

Figure 5.1. Distribution and global frequency of predicates in the resource namespace

There is an important difference in the predicates distribution in each mode. In the *description* mode, *domain* predicates are very few compared to *DL*, whereas in *usage* mode, they are almost equally balanced but far more numerous. We can conclude that *domain* concepts are widely **used**

in DBpedia in relation with domain predicates, but that they themselves seldom exploit this group in their **description**.

5.4.2 Analysis of the distribution of predicates in each group

As a first step, it is interesting to know how many concepts use at least one predicate of each group (this corresponds to the *Macro Concept Coverage Ratio* we used in our previous work [1]). We observed that, in the best case (in the *Music* domain), only 20% of the concepts are described by a predicate of the *domain* group, and in the worst case (in the *Construction* domain), the ratio does not exceed 1%. We calculated these numbers by considering that a concept is "covered" by a group if the concept is described by at least a predicate of the group, and by calculating the proportion of "covered" concepts for each group. These figures only apply to the *resource* namespace and not to the *ontology* namespace, since the DBpedia ontology does not contain any *domain* predicate. We also noticed that almost all concepts, regardless of the domain, are described by at least one *DL* predicate, with 4% of concepts not described by the *DL* group in the worst case.

To refine these observations, we introduce the measure of *concept coverage* (*Micro Concept Coverage Ratio* in [1]), which aims at analyzing the behavior of all predicates of the group, in a given domain. We calculate, for each predicate **that represents at least 10% of the occurrences of the group**, the proportion of domain concepts in whose description or usage the predicate appears, and then average this value on the cardinality of the group.

$$G_{10\%} = \left\{ p \in G \mid \sum_{c \in DC(D)} f(p,c) > 0.1 * \left(\sum_{c \in DC(D)} \sum_{p \in G} f(p,c) \right) \right\}$$

A predicate that belongs to $G_{10\%}$ is called a **main predicate** of the group G. We introduce this selection because each group contains a small subset of widely used predicates (typically, *DL* predicates: *rdf:type*, *owl:sameAs* and *dbo:type*) and one or more other predicates that seldom appear, meaning that, when calculating the average, an erroneous predicate that appears only a couple of times would reduce drastically the result. By taking into account only the main predicates of a group G, the concept coverage for a domain D is defined in the following way:

$$CCov(G,D) = \frac{1}{|G_{10\%}|} \sum_{p \in G_{10\%}} \frac{|\{c \in DC(D) \mid f(p,c) > 0\}|}{|DC(D)|}$$

This means that, for example, if a group G has a concept coverage of 0.15 for a given domain, on average, a main predicate of the group is used in the description of 15% of the DCs.

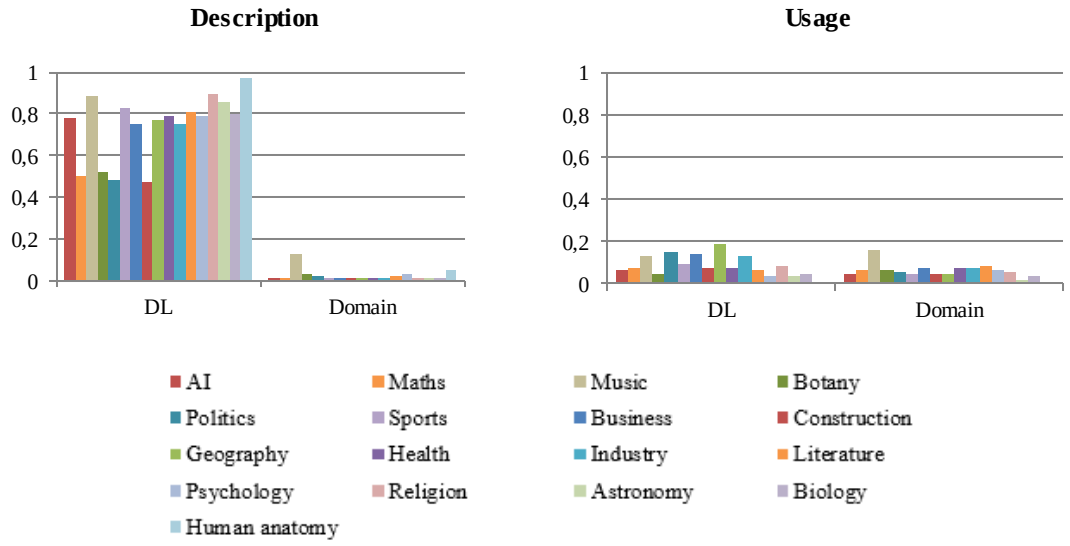

Figure 5.2. Concept coverage for the DL and Domain groups per domain, in the resource namespace

Figure 5.2 shows the concept coverage values for all domains. These results confirm the observations made at the beginning of the section: *DL* predicates are widely used in the description of concepts, regardless of the domain, whereas *domain* predicates appear very rarely. The novel information, however, is that each individual *domain* predicate appears in a very low number of concepts: on average, each *domain* predicate appears in less than 6% (except in the domain Music) of the concepts' **description** and less than 15% in their **usage**.

5.4.3 A closer look at the DL group

As we mentioned previously, the *DL* group contains 3 main predicates that represent the majority of all the triples: *rdf:type*, *owl:sameAs* and *dbo:type*. In this section, we look more finely at the usage of this group.

Concerning the DBpedia resources' description (**Q1**), the predominant predicates are *owl:sameAs* and *rdf:type*, used respectively to indicate an URI that describes the same entity or concept, and to provide a type relation with the ontology, such as *dbr:Barack_Obama rdf:type dbo:Person*. These two predicates represent respectively 54.6% and 45.1% of the *DL* group in this namespace

(resource) and mode (description). Only the *rdf:type* predicate is of interest here, as *owl:sameAs*'s only potential usage for knowledge inference is to indicate an equivalent resource in another LOD set, and we focus only on DBpedia in this paper. To assess the capabilities of DBpedia for knowledge inference by using *rdf:type*, we want to know the proportion of DBpedia resources that are typed, and the origin of the type, as the object of the *rdf:type* triple can be either in the DBpedia ontology, or in another dataset. Figure 5.3 provides the distribution of concepts that have a type in various LOD datasets.

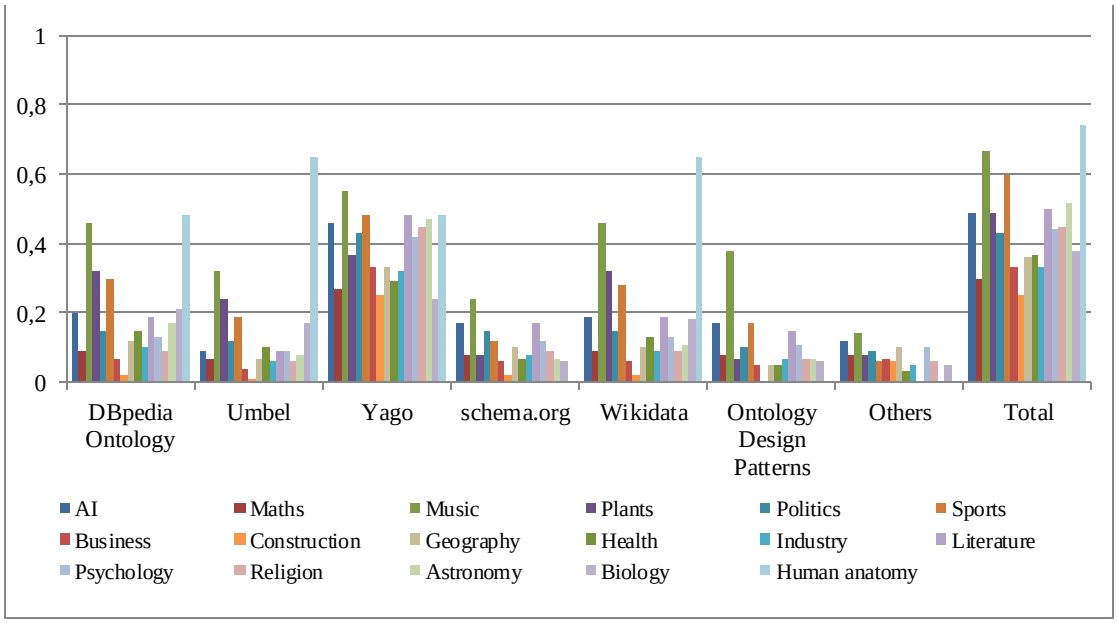

Figure 5.3. Typing of concepts in various Linked Open Datasets

In section 4.2, we mentioned that almost every concept uses a *DL* predicate. However, as we can notice here, many concepts are still un-typed: depending on the domain, only 2 to 48% have a type in the DBpedia ontology, and only 25 to 74% have a type overall. On the average 81% of the concepts do not have a type in the DBpedia ontology and 55% do not have a type at all.

Concerning the resources' usage (**Q2**), the dominant predicate is *dbo:type*, representing 97.4% of the *DL* group in this namespace and mode (30,934 occurrences among 31,765). This predicate appears in the usage of 539 concepts, an average of 57.4 occurrences per concept. The way this predicate is used in DBpedia suggests that its semantics is very similar to *rdf:type*, since its object is almost always something that could be considered as a class (such as *dbr:Village*, *dbr:Town*, *dbr:Lake*). Therefore, it could be used to answer our **Q4** by identifying potential classes. We consider this in Section 5.

Concerning the *ontology* (**Q1-3**), the *DL* group is mostly used to create the ontological structure using *rdfs:subClassOf, owl:equivalentClass* and *owl:disjointWith* among classes (both in their description and usage), and *rdf:type* between resources and classes. Each class in our dataset has on average 4769 instances, represented by *rdf:type* links.

5.4.4 Concepts linking among domains

In the previous sections, we studied the linkage between domain concepts and other DBpedia resources. In this section, we focus on the links **between concepts in the same domain**.

There are three possible types of links: resource to resource (6101 links), resource to ontology (2056 links) and ontology to ontology (13 links). Table 5.5 provides the average number of links per concept (total number of links/total number of concepts) in each namespace and domain. In the first two columns we give the average number of outbound links per domain concept in the *resource* namespace (a link may be to another resource or a concept in the DBpedia ontology). The last two columns concern domain concepts that are in the *ontology* namespace. Note that the third column considers the inbound links, since it is the kind of link we expect to find between a resource and a class in the ontology.

Table 5.5. Ratio links / number of concepts

Domain	Resource namespace		Ontology namespace	
	Links to a resource	Links to ontology	Links from a resource	Links to ontology
Artificial intelligence	0.20	0.00	0.00	0.00
Mathematics	0.19	0.00	0.00	0.00
Music theory	2.85	0.01	0.33	0.00
Botany	0.14	0.00	0.00	0.00
Political science	0.35	0.02	0.67	0.00
Sports science	0.29	0.21	8.50	0.00
Business	0.17	0.03	1.80	0.20
Construction	0.07	0.01	0.40	0.00
Geography	0.22	0.05	0.81	0.14
Health sciences	0.33	0.11	8.67	0.00
Industry	0.18	0.00	0.00	0.00
Literature	0.25	0.38	25.80	0.20
Psychology	0.51	0.00	0.00	0.00

Religion	0.70	0.00	0.00	0.00
Astronomy	0.46	0.14	15.50	0.13
Biology	0.29	0.18	5.82	0.36
Human anatomy	1.98	0.77	179.78	0.11
Average	**0.54**	**0.13 (*)**	**16.54 (*)**	**0.08 (*)**

(*) When the *ontology* namespace is concerned, the average is calculated only on the 15 (out of 17) domains that have at least one concept in the ontology.

As we can see, apart few exceptions, the number of links is quite low. In a well-described domain, we would expect at the very least one link to another concept of the same domain, which is the case here only for Music theory (on average 2.85 links with another resource) and Human anatomy (on average 1.98 links with another resource).

Concerning the resource-to-ontology links, the situation is even worse: among more than one million triples, there are only 13 links to the 100 DBpedia classes found in our dataset (an average 0.13 links per class).

Each domain concept that is present in the ontology is linked to an average of 16.5 resources (instances) of the same domain. Since each class has 4769 instances on average, this number is very low. It means that only 0.3% each class instances are in the same domain (16.5 out of 4769).

5.4.5 Summary of the results

In this section, we highlighted three weaknesses in the conceptual and domain-related knowledge inference capabilities of DBpedia:

1- Poor description of DBpedia resources in general, with almost no presence of domain-related predicates to describe concepts (Sections 4.1 and 4.2);

2- Poor linkage between the DBpedia T-box and A-box, with very few (2 to 48%) concepts that are actually typed in the DBpedia ontology (Section 4.3);

3- Very few links between concepts of a same domain (Section 4.4).

5.5 DBpedia Enrichment

In this section, we propose two methods to correct these limitations. This first method relies on open relation extraction (Section 5.1) for the extraction of predicates from the abstracts associated

to our domain concepts in DBpedia. We extract both domain-related predicates (Section 5.2) and *rdf:type* predicates (Section 5.3). The second method consists in analyzing the *dbo:type* predicate used in DBpedia and the hyponymy relations extracted by our first method, to identify potential classes among our domain concepts (Section 5.4).

5.5.1 Open relation extraction

Open relation extraction consists in extracting segments that express a relation from texts, without any predefined and limited set of relations. In our experiment, we ran the open relation extractor ReVerb on the Wikipedia abstracts of every concept in our dataset. [9]. Given the following input text (the first sentence of the abstract of the concept *Handwriting recognition*):

"Handwriting recognition (or HWR) is the ability of a computer to receive and interpret intelligible handwritten input from sources such as paper documents, photographs, touch-screens and other devices."

ReVerb extracts two relations:

⟨Handwriting recognition; is the ability of; a computer⟩

⟨the ability of a computer; interpret; intelligible handwritten input⟩

We lemmatized the subject, the object and the relation based on Stanford CoreNLP [20] and removed determiners from the subjects and objects, to obtain a format similar to DBpedia URIs (no plural, no article, etc.). The lemmatized forms of the two previous relations are the following ones:

⟨handwriting_recognition; be_the_ability_of; computer⟩

⟨ability_of_a_computer; interpret; intelligible_handwritten_input⟩

Based on the set of relations extracted from all the abstracts, we only keep the relations for which **both the subject and the object are among the previously identified domain concepts**. In the example, we keep only the first one, since *Computer* is a recognized concept whereas *intelligible_handwritten_input* is not.

[20] http://stanfordnlp.github.io/CoreNLP/

There were 382 unique relations extracted by ReVerb, but most of them (329) appear only once, mostly because they are very specific ("is any set of", "are very tightly bound by") or sometimes because they are erroneous, with the inclusion of punctuation for instance (". There are various types of"). Table 5.6 gives the most frequent relations (at least 5 occurrences) and, for each one, the number of occurrences.

Table 5.6. Most frequent relations extracted by ReVerb in our dataset

Predicate	Freq.
is	91
is a branch of	43
is the branch of	21
is a type of	11
includes	9
is a form of	8
is a subfield of	5
is an artery of	5
is a genre of	5

We manually classified the relations into the following categories and their associated predicates:

- Equivalence relations: *owl:sameAs / owl:equivalentClass*
- Mutual exclusion relations: *owl:differentFrom / owl:disjointWith*
- Hypernymy/hyponymy relations: *rdf:type / rdfs:subClassOf / dbo:type*
- Domain relations: used as default if none of the preceding categories is selected.
- None: when the extracted relation is erroneous or nonsensical.

To perform the classification into categories, we asked four computer science Master's students at École Polytechnique de Montréal to assess the relations extracted by ReVerb. Each one assigned a category to every relation. The final category of each relation was selected by performing a majority vote. In case of equality, we asked a fifth evaluator to choose.

Table 5.7 indicates the categories in which we classified the most frequent relations. We can observe that the majority of occurrences are domain relations, followed by hypernymy relations.

Table 5.7. Distribution of the most frequent relations

Relation category	Relations	Total number of occurrences
Equivalence	is the equivalent of ; is sometimes referred to as ; is often used synonymously with ; is also known as ; is often called ; is known as	10
Mutual exclusion	is neither ; is distinguished from ; is not to be confused with ; is different from ; is not synonymous with	9
Hypernymy	is ; is a type of ; are examples of ; is a certain kind of ; is a particular pattern of ; is sometimes classified as ; is the type of	254
Hyponymy	includes ; consists of ; can include activities such as	11
Domain	is an artery of ; has ; is the scientific study of ; is the study of ; is an approach to ; arises from	347
None	is substantially altered.It is difficult to find absolutely ; are dwarf ; .There are various types of	10

As mentioned previously, this distribution is the result of a vote among four evaluators. The Fleiss' kappa on this evaluation is 0.59, with a 95% confidence interval of [0.57, 0.61], representing a moderate / strong agreement. In case of equality, we asked a fifth evaluator to decide.

5.5.2 Extraction of domain-related predicates

In this section, we are interested in determining if the extracted relations are already represented in DBpedia in the resource namespace (links with ontology namespace are discussed in next section). This enables us to evaluate how open relation extraction may contribute to enrich DBpedia with new triples.

To accomplish this task, the first step is to look for triples in DBpedia relating domain concepts pairs extracted by ReVerb. For instance, based on the relation ⟨*robotics; focuses on; robots*⟩ extracted from the abstract of "robotics", we note that the pair (*Robotics, Robot*) is linked through the triple ⟨*dbr:robotics, rdfs:seeAlso, dbr:robot*⟩ in DBpedia.

In the second step, we manually assessed if the extracted relations between concepts' pairs provide, at least partially, some novel information compared to the triples already in DBpedia. In the previous example, it is the case, as *rdfs:seeAlso* only indicates that the two concepts are somehow related, whereas the relation "*focuses on*" points out that *robots* is a central concept in *robotics*, providing some novel information. This comparison was done by retrieving all triples linking the subject and object of an extracted relation, and comparing them. For most relations, no such triple could be found in DBpedia, reducing drastically the difficulty of the task.

Table 5.8. Number of novel relations

Domain	Extracted relations	Novel relations	Ratio
A.I.	35	34	0.97
Astronomy	144	143	0.99
Biology	3	3	1.00
Botany	24	22	0.92
Business	21	21	1.00
Construction	10	10	1.00
Geography	52	50	0.96
Health sciences	42	41	0.98
Human anatomy	126	110	0.87
Industry	9	8	0.89
Literature	34	34	1.00
Mathematics	32	32	1.00
Music theory	27	26	0.96
Political science	10	10	1.00
Psychology	39	37	0.95
Religion	18	18	1.00
Sports science	5	5	1.00
All domains	**631**	**604**	**0.96**

Table 5.9. Number of novel relations

Category	Extracted relations	Novel relations	Ratio
Hypernymy	254	254	1.00
Hyponymy	11	11	1.00
Mutual exclusion	9	9	1.00
Equivalence	10	10	1.00
Domain	347	320	0.92
None	10	N/A	N/A

Table 5.8 provides, for each domain, the number of novel relations, and their proportion among all extracted relations.

We also provide in Table 5.9 the number of novel relations per category, for all domains together.

We can note that most of the extracted relations **are not represented** in DBpedia (all the ratios are close to 1, meaning that almost all extracted relations are novel). In 8 domains out of 17 (that is, where the ratio is equal to 1), DBpedia does not contain any triple between the concept pairs extracted by ReVerb. Out of the 631 extracted relations (excluding the 10 "invalid" relations, i.e. the group "none" in table 8), only 27 are represented in DBpedia (4%), and all of them are of the *Domain* category. This shows that most relations are indeed novel in DBpedia, and that open relation extractors are a suitable technology to generate new domain knowledge.

5.5.3 Extraction of rdf:type links

In this section, our objective is to assess if the Open Relation Extraction paradigm can be used efficiently to relate DBpedia resources with the DBpedia ontology. For each relation, we queried DBpedia to find if the subject or the object has a corresponding concept in the ontology. For instance, given the relation ⟨*flat_bone, is, bone*⟩, we find that the class *dbo:Bone* exists in the DBpedia ontology and we know that *flat_bone* already exists in the DBpedia resources *dbr:Flat_bone*. Thus *dbo:Bone* should be related to the entity *dbr:Flat_bone* through an *rdf:type* link (since "is" designates a hypernymy relation). If this link is not present in DBpedia, our approach highlights that it should be.

Table 5.10 provides the number of concept pairs present in the ontology where the relation represents hypernymy or hyponymy. In the 36 cases where a correspondence is found (out of 631), only the subject or the object is mapped to the ontology and never both.

Table 5.10. Number of hyponymy relations for which the subject is in the ontology, and hypernymy relations for which the object is in the ontology

Domain	Nb. relations extracted	Links with the ontology	
		Subject	Object
Artificial_intelligence	35	0	0
Astronomy	144	3	3
Biology	3	0	0
Botany	24	0	3
Business	21	1	0
Construction	10	0	0
Geography	52	2	3
Health_sciences	42	0	12
Human_anatomy	126	0	2
Industry	9	0	0
Literature	34	3	3
Mathematics	32	0	0
Music_theory	27	0	0
Political_science	10	0	0
Psychology	39	0	0
Religion	18	0	0
Sports_science	5	1	0
Total	631	10	26

An important point is that **all these relations are novel**. We highlighted before the lack of linkage between the A-box and the T-box in DBpedia, and especially the poor typing of domain concepts. We prove here that ORE tools are relevant to partly correct this issue. An example of such an extracted relation is <*Milky Way, is, galaxy*>, allowing us to infer that <*dbr:Milky_Way, rdf:type, dbo:Galaxy*>, which is not present in DBpedia. We manually assessed the extracted relations and concluded that, for 14 out of 36 cases, there is indeed an instance/class relationship between the concepts that is not represented by a *rdf:type* in DBpedia.

5.5.4 Domain class identification

In this section, we present an approach to identify domain concepts that represent classes, but that do not appear in the DBpedia ontology. To accomplish this, we propose two methods. The first one is based solely on the information present in DBpedia, more specifically on the predicate *dbo:type*. The second uses the hypernymy relations extracted by ReVerb.

5.5.4.1 Identification by *dbo:type*

In this approach, we hypothesize, based on our observation of its usage, that the *dbo:type* predicate has a similar role to *rdf:type*, i.e. to indicate an instance/class relationship between two DBpedia entities. Therefore, the object of such a predicate is potentially a class. For example, if we have the triple *dbr:Seattle dbo:type dbr:City*, we consider that *dbr:City* is a potential class, even though it is not present in the ontology.

In our dataset, we identified 539 potential classes (that are the object of at least one *dbo:type* triple), with an average of 54.24 instances per potential class. However, 196 among the 539 potential classes have the biggest number of instances (at least 5 instances) and represent more than 95% of the occurrences. Because we conducted a manual evaluation of whether these candidates are indeed classes, we focus on these 196 potential classes.

We relied on a vote between four evaluators, who assessed the validity of each of those 196 candidates. The Fleiss' kappa for this evaluation is 0.43, with a 95% confidence interval of [0.40, 0.46], representing a moderate agreement.

Table 11 provides the results of this vote. A candidate can be accepted (it is a class that should be in the ontology), refused (it is not a class) or questionable (for instance, *Research* can be considered as a class, but the *dbo:type* triples present in DBpedia are nonsensical, such as *<dbr:University_of_Oregon dbo:type dbr:Research>*).

Table 5.11. Results of the evaluation for the dbo:type-based method

Result	Accepted	Refused	Questionable
Number of concepts	112	66	18
Percentage	57%	33%	9%

As we can see, this method yields moderately good results, with a precision of 57% (68% when we also consider the questionable classes).

5.5.4.2 Identification by hypernymy relations

In this method, we exploit the relations extracted in Section 5.1. In our classification of the extracted relations, we determined that some of them represented hypernymy links. Because of the nature of such links, the object is a potential candidate class. We extracted 254 hypernymy relations. Some have the same object, leading to a total of 143 candidates.

Following the same approach, we evaluated each candidate to assess if it should be a class by performing a vote between four evaluators. The Fleiss' kappa for this evaluation is 0.59, with a 95% confidence interval of [0.55, 0.63], representing a strong agreement.

Table 5.12 provides the results of this evaluation. Like before, a candidate can be accepted, refused or questionable.

Table 5.12. Results of the evaluation for the ORE-based method

Result	Accepted	Refused	Questionable
Number of concepts	93	20	30
Percentage	65%	14%	30%

This second method yields better results than the first one, with a precision of 65%. Besides, there is a low number of firm refusals (14%), with 30% of questionable cases. These cases represent candidates that could arguably be classes depending on the context, and therefore the precision in practice could be as high as 86%.

Overall, the first method, based on *dbo:type*, provides 112 concepts that should be classes, out of 196 candidates. The second method, based on ORE, provides 610 novel relations and identifies 93 concepts that should be classes.

5.6 Discussion

In this section, we refer to the elements highlighted previously in order to answer our research questions, presented in Section 1.

5.6.1 Assessing the quality of domain knowledge in DBpedia (Q1-3)

The first three research questions concern three aspects of the quality of domain knowledge. Q1 and Q2 ask whether domain concepts are well *described* and *used* in DBpedia, respectively, whereas Q3 concerns the predicates that are present in the description and usage of domain concepts.

In Section 4, we confirmed some of the conclusions drawn in our previous work [6]. Even for the domains that are the most represented in the DBpedia ontology (Astronomy, Biology, Geography, Human anatomy), we noticed a serious lack of connection between the ontology and the resources, with only 48% of concepts typed in the DBpedia ontology in the best case, and 2% in the worst. We also noticed that concepts are much more used than they are described: this means that, when exploring DBpedia as a graph, many concepts represent "Domain sinks", i.e. nodes with only inbound *Domain* links (Q1, Q2). We also noticed a disparity in the *domain* (i.e. *dbo*) predicates: some of them are much more used than others, to the point where some predicates only appear once in the entire dataset, such as *dbo:governor* or *dbo:musicBy* (Q3). We have not investigated this further, as this is not the point of this paper, but we suspect that there could be room for improvement here: for instance, the predicate *dbo:musicBy* appears only 1,402 times in all of DBpedia and could be replaced in most cases by the predicate *dbo:musicComposer* (62,034 occurrences).

Concerning the linking among concepts of the same domain in the *resource* namespace, we confirmed the extremely low number of links (less than 1 per concept to another concept in the same domain, for all but two domains). There is also a low number of links towards domain concepts present in the ontology: even though between 25 and 74% of concepts are typed (depending on the domain), only 13% on average are typed **within** the domain. The conclusion that DBpedia lacks domain knowledge is however tempered by the fact that our method to create domains is still incomplete and probably misses many concepts that should be in the domain.

Another important point concerns the ontology. We already knew from our previous work that the DBpedia ontology is poorly linked to the domain concepts. In this study, we noticed a new crucial point: there are several classes in our dataset (33 out of 100) that appear in the ontology and have **no instance at all**, like "psychologist" or "law". Unlike most of our other conclusions, this lack of linkage applies to all DBpedia resources, and not only to our relatively small set of domain

concepts: these 33 classes do not have any instance in **all of DBpedia**. Given the small size of the DBpedia ontology as a whole (685 classes[21]), these classes still represent 5% of the ontology that is completely unlinked to the A-Box.

However, in all cases, our point is that the *domain* group is almost never present to **describe** concepts. This point is even more valid as this group arguably contains more predicates than it should. Many predicates occur very rarely, indicating a lack of reuse across DBpedia.

5.6.2 Predicate and class discovery using relation extraction and *dbo:type* (Q4)

In the second part of this study, we used open relation extraction to identify relations in Wikipedia abstracts that could enrich the DBpedia description of domain concepts. We restricted our work on the abstract because Wikipedia page are highly variable in their size and precision, whereas the abstract follows rules, formal and informal, ensuring that the information obtained is relevant and concise. We also used the particular predicate *dbo:type* and the extracted hypernymy relations to identify potential classes to be added to the DBpedia ontology.

Even if ReVerb did not provide a high number of new relations, we showed that most of the extracted relations were not already present in DBpedia, with only 4% of redundancy. This means that 96% of the extracted relations were entirely novel, or at least provided some novel information compared to the triple(s) already present in DBpedia.

We also pointed throughout this paper that the links between resources and the ontology are rare, and that the DBpedia ontology only contains a few domain concepts. Some of the extracted relations could be used to suggest DBpedia resources that should be ontology classes or to provide a type to a resource in the DBpedia ontology (14 relations). One limit of our approach is that these numbers represent only a small proportion of the extracted relations. In fact, a limitation of our work comes from the approach used to identify domain concepts. This method is by no means exhaustive, so we cannot consider that we were able to identify all the concepts relevant to a particular domain. Because we only consider relations where both the subject and object are part

[21] http://wiki.dbpedia.org/services-resources/ontology

of a domain, an enrichment of the recognized domain concepts could help further expand the set of applicable relations.

When it comes to relations between resources, the small number of identified relations can also be considered as a limit of our approach. We have a total of 631 extracted relations that link two domain concepts, for a total of 6835 concepts in our dataset. This represents approximately one new relation for every 11 concepts, or 0.089 relation per concept. This could be mitigated by exploring other open relation extractors or by parsing all Wikipedia texts mentioning concept pairs rather than only the abstract of each domain concept.

Additionally, we have classified the extracted relations into categories that contain at least two predicates (*Mutual exclusion* for instance), and at most a very high number of predicates (*Domain*). This is sufficient for a first coarse-grained analysis of the results. However, a finer-grained analysis would be to associate the extracted relations to RDF predicates automatically. This is left for future work.

Concerning the potential classes identification, our two methods obtained respectively a precision of 57% for the first one (with 112 new classes), and 65% for the second (with 93 new classes). However, these results do not take into account the granularity of the DBpedia ontology. Several of our identified classes are probably too precise to be included in the DBpedia ontology as such. One potential idea would be to create several fine-grained domain ontologies related to the upper-level DBpedia ontology.

Altogether, we showed the relevance of Open Relation Extraction for the task of improving DBpedia, both at the assertion-level and at the schema-level.

5.7 Conclusion and future Work

In this paper, we confirmed the conclusion drawn in our previous work [6] on a larger set of domains, highlighting the lack of domain knowledge representation in DBpedia, especially at the ontology level. We also enhanced our method to answer the question “What are the concepts that should belong to a given domain?”, notably by exploiting the information contained in the abstracts of a small number of reliable concepts. We extended our analysis of the current state of DBpedia by also considering the linkage with the ontology and the usage of concepts. We concluded that

improvements are still to be made on DBpedia to represent more extensively the knowledge contained in Wikipedia, essentially for the description of concepts and their linkage to the ontology.

We also proposed a method to exploit Wikipedia abstracts to infer relations between domain concepts. This method proved quite effective although limited in terms of the number of discovered relations. In parallel, we exploit these relations to discover new classes. This approach proved more effective than the method based on a direct exploration of DBpedia RDF triples.

The approach we propose here is still in development, but already provides interesting results. Our future work will consist of providing automatic methods to classify the extracted relations to compare more finely the redundancy between the results of open relation extraction and the triples already present in DBpedia.

5.8 Acknowledgments.

This research has been funded by the NSERC discovery grant program.

5.9 References

[1] L. Font, A. Zouaq, et M. Gagnon, « Assessing the Quality of Domain Concepts Descriptions in DBpedia », dans *2015 11th International Conference on Signal-Image Technology Internet-Based Systems (SITIS)*, Bangkok, 2015, p. 254-261.

[4] J. M. Juran et J. A. De Feo, *Juran's Quality Handbook*. .

[5] A. Zaveri, A. Rula, A. Maurino, R. Pietrobon, J. Lehmann, et S. Auer, « Quality assessment for linked data: A survey », *Semantic Web*, vol. 7, nº 1, p. 63-93, 2015.

[7] C. Bizer, *Quality-Driven Information Filtering- In the Context of Web-Based Information Systems*. Saarbrücken, Germany: VDM Verlag, 2007.

[8] A. Hogan, J. Umbrich, A. Harth, R. Cyganiak, A. Polleres, et S. Decker, « An empirical survey of Linked Data conformance », *Web Semant. Sci. Serv. Agents World Wide Web*, vol. 14, p. 14-44, juill. 2012.

[9] J. Debattista, C. Lange, et S. Auer, « Luzzu Quality Metric Language -- A DSL for Linked Data Quality Assessment », *ArXiv150407758 Cs*, avr. 2015.

[10] P. N. Mendes, H. Mühleisen, et C. Bizer, « Sieve: Linked Data Quality Assessment and Fusion », dans *Proceedings of the 2012 Joint EDBT/ICDT Workshops*, New York, NY, USA, 2012, p. 116-123.

[12] C. Bizer *et al.*, « DBpedia - A Crystallization Point for the Web of Data », *Web Semant*, vol. 7, nº 3, p. 154-165, sept. 2009.

[13] A. Zaveri *et al.*, « User-driven Quality Evaluation of DBpedia », dans *Proceedings of the 9th International Conference on Semantic Systems*, New York, NY, USA, 2013, p. 97-104.

[14] D. Kontokostas *et al.*, « Test-driven Evaluation of Linked Data Quality », dans *Proceedings of the 23rd International Conference on World Wide Web*, New York, NY, USA, 2014, p. 747-758.

[15] M. Acosta, A. Zaveri, E. Simperl, D. Kontokostas, S. Auer, et J. Lehmann, « Crowdsourcing Linked Data Quality Assessment », dans *The Semantic Web – ISWC 2013: 12th International Semantic Web Conference, Sydney, NSW, Australia, October 21-25, 2013, Proceedings, Part II*, H. Alani, L. Kagal, A. Fokoue, P. Groth, C. Biemann, J. X. Parreira, L. Aroyo, N. Noy, C. Welty, et K. Janowicz, Éd. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, p. 260-276.

[16] S. Atdağ et V. Labatut, « A Comparison of Named Entity Recognition Tools Applied to Biographical Texts », dans *2nd International Conference on Systems and Computer Science*, Villeneuve d'Ascq, France, 2013, p. 6p.

[17] C. De Maio, G. Fenza, M. Gallo, V. Loia, et S. Senatore, « Formal and relational concept analysis for fuzzy-based automatic semantic annotation », *Appl. Intell.*, vol. 40, nº 1, p. 154-177, 2014.

[18] B. Liu, « Sentiment analysis and opinion mining », *Synth. Lect. Hum. Lang. Technol.*, vol. 5, nº 1, p. 1-167, 2012.

[20] M. Gagnon, A. Zouaq, F. Aranha, F. Ensan, et L. Jean-Louis, « Semantic Annotation on the Linked Data Cloud: A Comprehensive Evaluation », *Journal of Web Semantics, Elsevier*, submitted-2016.

[25] J. Weston, A. Bordes, O. Yakhnenko, et N. Usunier, « Connecting Language and Knowledge Bases with Embedding Models for Relation Extraction », *ArXiv13077973 Cs*, juill. 2013.

[26] O. Etzioni, M. Banko, S. Soderland, et D. S. Weld, « Open Information Extraction from the Web », *Commun ACM*, vol. 51, nº 12, p. 68-74, déc. 2008.

[28] K. Narasimhan, A. Yala, et R. Barzilay, « Improving Information Extraction by Acquiring External Evidence with Reinforcement Learning », *ArXiv160307954 Cs*, mars 2016.

[29] H. Poon et P. Domingos, « Joint inference in information extraction », dans *Association for the Advancement of Artificial Intelligence*, 2007, vol. 7, p. 913-918.

[30] A. Fader, S. Soderland, et O. Etzioni, « Identifying Relations for Open Information Extraction », dans *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, USA, 2011, p. 1535-1545.

[31] F. Wu et D. S. Weld, « Open Information Extraction Using Wikipedia », dans *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, USA, 2010, p. 118-127.

[32] L. Del Corro et R. Gemulla, « ClausIE: Clause-based Open Information Extraction », dans *Proceedings of the 22Nd International Conference on World Wide Web*, New York, NY, USA, 2013, p. 355-366.

[44] Y. Yan, N. Okazaki, Y. Matsuo, Z. Yang, et M. Ishizuka, « Unsupervised Relation Extraction by Mining Wikipedia Texts Using Information from the Web », dans *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2*, USA, 2009, p. 1021-1029.

[45] A. Hogenboom, F. Hogenboom, F. Frasincar, K. Schouten, et O. van der Meer, « Semantics-based information extraction for detecting economic events », *Multimed. Tools Appl.*, vol. 64, nº 1, p. 27-52, 2013.

CHAPITRE 6 DISCUSSION GÉNÉRALE

Ce chapitre a pour but de fournir une discussion objective sur l'ensemble du sujet abordé lors de cette maîtrise. Bien que les deux articles présentés disposent chacun d'une discussion, nous allons plus loin ici et regardons l'ensemble du travail effectué. La première section reprend les questions de recherche posées dans la section 1.2, la deuxième présente les limitations de notre travail, et la troisième propose des avenues possibles pour le travail futur.

6.1 Réponse aux questions de recherche

Q1 : Comment définir exactement ce qu'est un « domaine de connaissance » ?

Pour tenter de répondre à cette question, nous avons utilisé deux approches. Celle du premier article est basée sur les catégories Wikipédia, et celle du second est basée sur les pages *Outline of* de Wikipédia. Ces deux approches présentent l'inconvénient d'être dépendantes de la communauté Wikipédia, et donc sujettes à des imprécisions ou erreurs. Cependant, la deuxième approche repose sur des pages dont le but explicite est de fournir un ensemble de concepts liés au domaine de départ, ce qui correspond tout à fait à notre tâche. L'inconvénient, que l'on retrouve dans une moindre mesure dans la première approche, est l'absence inévitable d'un certain nombre de concepts qui devraient faire partie du domaine, mais que l'on n'obtient pas par notre extraction. C'est là qu'apparait la difficulté majeure de cette question de recherche : jusqu'où doit-on aller dans la précision pour définir un domaine ? Les deux approches que nous proposons ne font essentiellement que proposer une solution pratique au problème. Notre réponse à cette question est donc simplement : un domaine est défini par l'ensemble des concepts que l'on peut récupérer par l'une des approches présentées. Il ne s'agit pas d'une réponse satisfaisante à l'aspect philosophique de la question, mais suffisante pour ce travail : étant donné que notre tâche principale est d'évaluer la qualité de DBpedia,

Q2 : Quels aspects de la « qualité » d'un domaine pouvons-nous évaluer quantitativement, et quel est le résultat de cette évaluation ?

Nous avons évalué plusieurs aspects de la qualité d'un domaine quantitativement : quantité de liens avec d'autres ressources permettant d'obtenir de l'information pertinente sur le concept initial, quantité de liens avec d'autres ressources du même domaine, nombre de liens avec l'ontologie DBpedia, type et quantité de prédicats utilisés dans les triplets impliquant le domaine. Le résultat

révèle certaines limites de DBpedia : les concepts sont souvent mal décrits (peu de liens avec les autres ressources DBpedia, et encore moins avec des ressources liées au domaine), et ne disposent souvent pas de liens avec l'ontologie (81% de concepts non typés dans DBpedia), qui constituent un élément essentiel pour faire des inférences avec l'ontologie.

Nous avons également, dans le deuxième article, évalué informellement l'efficacité de DBpedia pour répondre à un problème concret. Bien que cela n'ait pas fait l'objet d'une analyse poussée car il ne s'agissait pas du sujet de l'article en question, nous avons constaté, premièrement, que la majorité des types extraits, qui devraient donc par définition être des classes, ne sont pas présents dans l'ontologie DBpedia, et, deuxièmement, que les entités DBpedia associées aux types extraits ne disposent en général d'aucun lien direct avec l'ontologie DBpedia. Nous avons tenté d'exploiter d'autres informations disponibles dans DBpedia pour établir ces liens.

Q3 : Par quelles méthodes pouvons-nous améliorer la qualité de l'information conceptuelle de DBpedia ?

Nous proposons plusieurs idées pour améliorer DBpedia. La première consiste à créer de nouveaux liens entre les concepts liés à un domaine, en exploitant l'information contenue dans le texte de Wikipédia. Nous obtenons 631 nouveaux liens, dont 604 sont inédits dans DBpedia.

La seconde est en fait une simple extension de la première, dans le cas spécifique des liens de typage entre les concepts et l'ontologie DBpedia. Nous obtenons 14 nouvelles relations de typage.

Enfin, nous proposons deux approches pour suggérer de nouvelles classes à ajouter à l'ontologie. La première est une simple utilisation logique du prédicat *dbo:type* déjà présent, et étant donc facilement applicable à tout DBpedia mais fournissant une quantité d'information entièrement dépendante du contenu de DBpedia. L'autre approche est plus intéressante, car elle exploite, à l'image des deux autres méthodes présentées précédemment, l'information textuelle de Wikipédia. Dans son état actuel, elle obtient une meilleure précision que la première (81% contre 57%), mais fournit un plus petit nombre de classes (56 contre 112). Cependant, il s'agit d'une avenue beaucoup plus prometteuse pour l'avenir, car il y a énormément d'information à tirer du texte. Dans l'idéal, DBpedia devrait contenir toute l'information présente dans Wikipédia, ce qui inclut le texte des articles, et non juste les infobox. Ce premier jet d'une approche basée sur l'extraction de relations à partir du texte des articles Wikipédia permet de prouver qu'il s'agit effectivement d'une méthode

valide qui, moyennant un certain nombre d'améliorations, sera probablement pertinente à l'avenir pour améliorer DBpedia.

6.2 Limitations

Notre travail présente un certain nombre de limitations. La plupart de ces limitations sont discutées dans les sections respectives des articles (4.5 et 5.6), cette section a donc pour but de fournir une prise de recul sur le travail général effectué.

Tout d'abord, les deux approches que nous avons utilisées pour générer le dataset de test, et donc répondant à la question de « qu'est-ce qu'un domaine ? » sont imparfaites. La première crée une quantité de bruit non négligeable, et la seconde produit beaucoup moins de concepts, faisant qu'il peut manquer des concepts importants dans un domaine. De plus, ces deux méthodes dépendent entièrement du contenu de Wikipédia (arbre des catégories pour la première, pages *Outline of* et premier paragraphe des pages des concepts pour la seconde). Sans remettre en question la fiabilité de l'information offerte, Wikipédia n'est pas prévu pour une telle utilisation. Par exemple, le contenu des pages *Outline of* ne se veut pas exhaustif, d'où notre problème d'absence de certains concepts. Enfin, ces deux méthodes ne répondent pas vraiment à la question théorique, philosophique de ce qu'est un domaine de connaissances. Nous y répondons en définissant un domaine à partir de ce qui est présent sur Wikipédia.

Concernant l'évaluation, notre classification des prédicats en groupes reste subjective. Notre définition du groupe *DL* par exemple est « *DL* predicates are useful for inference ». Nous avons choisi de n'y inclure que les prédicats dont l'utilité pour de l'inférence est avérée (prédicats des vocabulaires *rdf*, *rdfs* et *owl*, à l'exception notable de *dbo:type*), mais cette définition pourrait aisément inclure d'autres prédicats *dbo*, à condition d'analyser plus finement leur utilisation dans DBpedia. De même, le groupe *Domain* contient tous les prédicats *dbo* sauf *dbo:type*. Il s'agit donc d'un éventail de prédicats assez large, qui pourrait être raffiné.

Par ailleurs, nous évaluons dans les deux articles (Chapitres 4 et 5) le nombre de liens reliant deux concepts du domaine. Bien qu'intéressants en théorie, en pratique ces nombres sont partiellement mis en défaut par l'imperfection de nos méthodes pour obtenir les domaines, particulièrement celle basée sur les pages *Outline of*. En effet, si les domaines sont incomplets, il est probable que notre mesure rate une quantité non négligeable de liens.

6.3 Avenues futures

Du point de vue de l'évaluation de DBpedia, il est important de noter qu'il s'agit d'un dataset évoluant continuellement. Entre l'écriture du premier article et de ce mémoire, une nouvelle version (2016-4) a été mise en ligne, corrigeant un certain nombre des problèmes mentionnés, notamment au niveau de la quantité de concepts typés. Ceci démontre qu'il est important d'avoir des outils permettant d'évaluer continuellement la qualité de DBpedia. Il serait très instructif d'effectuer toutes les étapes d'évaluation décrites dans ce travail sur les futures versions de DBpedia afin de constater les changements effectués et de vérifier dans quelle mesure les problèmes sont corrigés au fur et à mesure.

Concernant l'amélioration de DBpedia, les avenues futures sont multiples. En effet, les approches que nous proposons sont un premier jet permettant de fournir une évaluation préliminaire de leur efficacité potentielle. La méthode basée sur *dbo:type* est très spécifique : il serait possible de rechercher d'autres prédicats ayant des caractéristiques intéressantes, permettant d'inférer plus de connaissances. Il serait également intéressant d'évaluer l'efficacité de cette méthode sur l'ensemble de DBpedia, et non pas sur notre dataset de test restreint. La seconde méthode, basée sur l'extraction de relation, offre elle aussi de nombreuses possibilités d'amélioration. La plus simple est de considérer toutes les relations reliant le concept initial à n'importe quelle ressource DBpedia, et non plus uniquement celles reliant à un autre concept du domaine. Ceci introduirait sans doute un certain nombre d'erreurs, mais augmenterait drastiquement la quantité d'information récupérée. Il y a également un travail important à effectuer dans la classification des relations extraites : dans l'idéal, il faudrait être capable d'associer automatiquement la relation en langue naturelle extraite avec un prédicat DBpedia existant, là où notre système actuel dépend d'une évaluation manuelle, et n'associe que certaines relations (135 sur 641) avec des prédicats existants. Enfin, une autre possibilité est de regarder ce que l'on obtient lorsque l'on fait rouler l'extraction de relations sur l'ensemble du texte de la page Wikipédia, et non pas juste sur le premier paragraphe.

CHAPITRE 7 CONCLUSION

Ce travail de recherche a plusieurs apports. Tout d'abord, nous avons mis en place des outils et une méthodologie pour évaluer la qualité de DBpedia. Bien que les conclusions négatives que nous tirons puissent être invalidées, comme on peut l'espérer, par les prochaines versions de DBpedia, la méthodologie reste valide et permettra de suivre l'évolution de sa qualité au fil des versions. En l'état, nous avons pointé et chiffré numériquement certaines faiblesses de DBpedia pour l'information conceptuelle liée à un domaine. Certains points sont relativement aisés à corriger, comme l'absence de types dans une partie non négligeable de nos concepts, d'autres représentent des problèmes plus fondamentaux : il est difficile de structurer rigoureusement de l'information conceptuelle.

Nous avons également fourni un système permettant d'obtenir des informations ciblées (le type d'une entité) depuis une phrase respectant certaines contraintes. Le système dans son ensemble est efficace : il a obtenu les meilleurs résultats de la compétition, pour l'extraction et l'alignement.

Enfin, nous proposons une méthode permettant d'obtenir de nouvelles relations entre concepts, et deux méthodes suggérant de nouvelles classes à ajouter à l'ontologie. Les relations obtenues sont en immense majorité inédites, et, bien que l'on obtienne peu de nouvelles relations par rapport au nombre de concepts de départ, nous avons prouvé la pertinence de cette approche. Les nouvelles classes, quant à elles, sont, d'après l'évaluation manuelle effectuée, majoritairement correctes.

De manière générale, ce travail justifie le besoin de nouveaux outils pour améliorer DBpedia, et fournit des pistes intéressantes pour créer ces outils.

BIBLIOGRAPHIE

[1] L. Font, A. Zouaq, et M. Gagnon, « Assessing the Quality of Domain Concepts Descriptions in DBpedia », dans *2015 11th International Conference on Signal-Image Technology Internet-Based Systems (SITIS)*, Bangkok, 2015, p. 254-261.

[2] L. Font, A. Zouaq, et M. Gagnon, « Assessing and Improving Domain Knowledge Representation in DBpedia », *Submitted to the Open Journal of Semantic Web, 2016*.

[3] J. Z. Pan, « Resource Description Framework », dans *Handbook on Ontologies*, S. Staab et R. Studer, Éd. Springer Berlin Heidelberg, 2009, p. 71-90.

[4] J. M. Juran et J. A. De Feo, *Juran's Quality Handbook*. .

[5] A. Zaveri, A. Rula, A. Maurino, R. Pietrobon, J. Lehmann, et S. Auer, « Quality assessment for linked data: A survey », *Semantic Web*, vol. 7, nº 1, p. 63-93, 2015.

[6] A. Flemming, *Quality Characteristics of Linked Data Publishing Datasources*. 2010.

[7] C. Bizer, *Quality-Driven Information Filtering- In the Context of Web-Based Information Systems*. Saarbrücken, Germany: VDM Verlag, 2007.

[8] A. Hogan, J. Umbrich, A. Harth, R. Cyganiak, A. Polleres, et S. Decker, « An empirical survey of Linked Data conformance », *Web Semant. Sci. Serv. Agents World Wide Web*, vol. 14, p. 14-44, juill. 2012.

[9] J. Debattista, C. Lange, et S. Auer, « Luzzu Quality Metric Language -- A DSL for Linked Data Quality Assessment », *ArXiv150407758 Cs*, avr. 2015.

[10] P. N. Mendes, H. Mühleisen, et C. Bizer, « Sieve: Linked Data Quality Assessment and Fusion », dans *Proceedings of the 2012 Joint EDBT/ICDT Workshops*, New York, NY, USA, 2012, p. 116-123.

[11] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, et Z. Ives, « DBpedia: A Nucleus for a Web of Open Data », dans *The Semantic Web: 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, Busan, Korea, November 11-15, 2007. Proceedings*, K. Aberer, K.-S. Choi, N. Noy, D. Allemang, K.-I. Lee, L. Nixon, J. Golbeck, P. Mika, D. Maynard, R. Mizoguchi, G. Schreiber, et P. Cudré-Mauroux, Éd. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, p. 722-735.

[12] C. Bizer *et al.*, « DBpedia - A Crystallization Point for the Web of Data », *Web Semant*, vol. 7, nº 3, p. 154-165, sept. 2009.

[13] A. Zaveri *et al.*, « User-driven Quality Evaluation of DBpedia », dans *Proceedings of the 9th International Conference on Semantic Systems*, New York, NY, USA, 2013, p. 97-104.

[14] D. Kontokostas *et al.*, « Test-driven Evaluation of Linked Data Quality », dans *Proceedings of the 23rd International Conference on World Wide Web*, New York, NY, USA, 2014, p. 747-758.

[15] M. Acosta, A. Zaveri, E. Simperl, D. Kontokostas, S. Auer, et J. Lehmann, « Crowdsourcing Linked Data Quality Assessment », dans *The Semantic Web – ISWC 2013: 12th International Semantic Web Conference, Sydney, NSW, Australia, October 21-25, 2013, Proceedings, Part II*, H. Alani, L. Kagal, A. Fokoue, P. Groth, C. Biemann, J. X. Parreira, L. Aroyo, N. Noy, C. Welty, et K. Janowicz, Éd. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, p. 260-276.

[16] S. Atdağ et V. Labatut, « A Comparison of Named Entity Recognition Tools Applied to Biographical Texts », dans *2nd International Conference on Systems and Computer Science*, Villeneuve d'Ascq, France, 2013, p. 6p.

[17] C. De Maio, G. Fenza, M. Gallo, V. Loia, et S. Senatore, « Formal and relational concept analysis for fuzzy-based automatic semantic annotation », *Appl. Intell.*, vol. 40, nº 1, p. 154-177, 2014.

[18] B. Liu, « Sentiment analysis and opinion mining », *Synth. Lect. Hum. Lang. Technol.*, vol. 5, nº 1, p. 1-167, 2012.

[19] L. Reeve et H. Han, « Survey of Semantic Annotation Platforms », dans *Proceedings of the 2005 ACM Symposium on Applied Computing*, New York, NY, USA, 2005, p. 1634–1638.

[20] M. Gagnon, A. Zouaq, F. Aranha, F. Ensan, et L. Jean-Louis, « Semantic Annotation on the Linked Data Cloud: A Comprehensive Evaluation », *Journal of Web Semantics, Elsevier*, submitted-2016.

[21] D. Zelenko, C. Aone, et A. Richardella, « Kernel Methods for Relation Extraction », *J. Mach. Learn. Res.*, vol. 3, nº Feb, p. 1083-1106, 2003.

[22] A. Culotta et J. Sorensen, « Dependency Tree Kernels for Relation Extraction », dans *Proceedings of the 42Nd Annual Meeting on Association for Computational Linguistics*, Stroudsburg, PA, USA, 2004.

[23] A. Schutz et P. Buitelaar, « RelExt: A Tool for Relation Extraction from Text in Ontology Extension », dans *The Semantic Web – ISWC 2005*, 2005, p. 593-606.

[24] K. Fundel, R. Küffner, et R. Zimmer, « RelEx—Relation extraction using dependency parse trees », *Bioinformatics*, vol. 23, nº 3, p. 365-371, févr. 2007.

[25] J. Weston, A. Bordes, O. Yakhnenko, et N. Usunier, « Connecting Language and Knowledge Bases with Embedding Models for Relation Extraction », *ArXiv13077973 Cs*, juill. 2013.

[26] O. Etzioni, M. Banko, S. Soderland, et D. S. Weld, « Open Information Extraction from the Web », *Commun ACM*, vol. 51, nº 12, p. 68-74, déc. 2008.

[27] M. Banko et O. Etzioni, « The Tradeoffs Between Open and Traditional Relation Extraction. », 2008.

[28] K. Narasimhan, A. Yala, et R. Barzilay, « Improving Information Extraction by Acquiring External Evidence with Reinforcement Learning », *ArXiv160307954 Cs*, mars 2016.

[29] H. Poon et P. Domingos, « Joint inference in information extraction », dans *Association for the Advancement of Artificial Intelligence*, 2007, vol. 7, p. 913-918.

[30] A. Fader, S. Soderland, et O. Etzioni, « Identifying Relations for Open Information Extraction », dans *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, USA, 2011, p. 1535-1545.

[31] F. Wu et D. S. Weld, « Open Information Extraction Using Wikipedia », dans *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, USA, 2010, p. 118-127.

[32] L. Del Corro et R. Gemulla, « ClausIE: Clause-based Open Information Extraction », dans *Proceedings of the 22Nd International Conference on World Wide Web*, New York, NY, USA, 2013, p. 355-366.

[33] Mausam, M. Schmitz, R. Bart, S. Soderland, et O. Etzioni, « Open Language Learning for Information Extraction », dans *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, Stroudsburg, PA, USA, 2012, p. 523–534.

[34] B. Min, S. Shi, R. Grishman, et C.-Y. Lin, « Ensemble Semantics for Large-scale Unsupervised Relation Extraction », dans *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, Stroudsburg, PA, USA, 2012, p. 1027–1037.

[35] T. Berners-lee, D. Connolly, L. Kagal, Y. Scharf, et J. Hendler, « N3Logic: A Logical Framework for the World Wide Web », *Theory Pr. Log Program*, vol. 8, nº 3, p. 249-269, mai 2008.

[36] J. Lehmann *et al.*, « DBpedia–a large-scale, multilingual knowledge base extracted from Wikipedia », *Semantic Web*, vol. 6, nº 2, p. 167-195, 2015.

[37] F. M. Suchanek, G. Kasneci, et G. Weikum, « Yago: A Core of Semantic Knowledge », dans *Proceedings of the 16th International Conference on World Wide Web*, New York, NY, USA, 2007, p. 697-706.

[38] D. Vrandečić et M. Krötzsch, « Wikidata: A Free Collaborative Knowledgebase », *Commun ACM*, vol. 57, nº 10, p. 78-85, sept. 2014.

[39] A. P. Aprosio, C. Giuliano, et A. Lavelli, « Extending the Coverage of DBpedia Properties Using Distant Supervision over Wikipedia », dans *Proceedings of the 2013th International Conference on NLP & DBpedia - Volume 1064*, Aachen, Germany, Germany, 2013, p. 20-31.

[40] G. Piao et J. G. Breslin, « Measuring Semantic Distance for Linked Open Data-enabled Recommender Systems », dans *Proceedings of the 31st Annual ACM Symposium on Applied Computing*, New York, NY, USA, 2016, p. 315-320.

[41] « Ontology Learning from Text », dans *Ontology Learning and Population from Text*, Springer US, 2006, p. 19-34.

[42] S. Faralli et S. P. Ponzetto, « DWS at the 2016 Open Knowledge Extraction Challenge: A Hearst-Like Pattern-Based Approach to Hypernym Extraction and Class Induction », dans *Semantic Web Challenges*, H. Sack, S. Dietze, A. Tordai, et C. Lange, Éd. Springer International Publishing, 2016, p. 48-60.

[43] A. Kiryakov, B. Popov, I. Terziev, D. Manov, et D. Ognyanoff, « Semantic annotation, indexing, and retrieval », *Web Semant. Sci. Serv. Agents World Wide Web*, vol. 2, nº 1, p. 49-79, déc. 2004.

[44] Y. Yan, N. Okazaki, Y. Matsuo, Z. Yang, et M. Ishizuka, « Unsupervised Relation Extraction by Mining Wikipedia Texts Using Information from the Web », dans *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2*, USA, 2009, p. 1021-1029.

[45] A. Hogenboom, F. Hogenboom, F. Frasincar, K. Schouten, et O. van der Meer, « Semantics-based information extraction for detecting economic events », *Multimed. Tools Appl.*, vol. 64, nº 1, p. 27-52, 2013.

ANNEXE A – INSTRUCTIONS POUR LES ÉVALUATEURS

1- Classification des relations extraites par ReVerb

Dans cette évaluation se trouve des triplets (sujet, predicat, objet) représentant une relation. Il faut classifier la relation parmi les catégories suivantes :

- Equivalence, correspondant à un lien de type owl:sameAs, par exemple "is the equivalent of" ou "is known as"

- Separation, indiquant que les deux concepts sont distincts, par exemple "is distinguished from" ou "is not to be confused with"

- Hypernymy, indiquant un lien instance / classe ou sous-classe / classe, par exemple "is a type of", "is a division of"

- Hyponymy, l'inverse du précédent (le sujet est l'hypernyme de l'objet)

- Other, pour toutes les relations n'entrant dans aucune de ces catégories

- None, pour toutes les relations qui n'en sont pas (erreurs de parsing, e.g. ".There are various types of")

Pour indiquer qu'une relation appartient à une catégorie, simplement mettre un "1" dans la colonne appropriée et laisser les autres vides.

Quelques exemples :

<self-replicating_machine, is a type of, autonomous_robot>. Cette relation indique une hypernymie, car "self-replicating_machine" est une sous-classe de "autonomous_robot".

<bicameralism, is distinguished from, unicameralism>. Cette relation indique que unicameralism est différent de bicameralism, il s'agit donc d'une separation.

<starspot, are the equivalent of, sunspot>. Cette relation indique que le deux concepts sont identiques, il s'agit donc d'une équivalence.

<planet, orbited, sun>. Cette relation n'indique pas de lien sémantique entre les deux concepts, il s'agit donc d'une "Other".

2- Validation des classes potentielles obtenues par ORE

Dans cette évaluation, il faut déterminer, parmi la liste de candidats proposés, lesquels sont des classes.

N.B. : On s'intéresse ici à la définition ontologique d'une classe, i.e. il s'agit de collections, définies par des contraintes pour en faire partie. Par exemple, "City" est une classe car il s'agit d'une collection (contenant Montréal, Paris, New York...) imposant un certain nombre de conditions sur ces membres.

Il y a 3 réponses possibles :

- Accept : le candidat est sans doute possible une classe (ex : Village, Song, Company...)

- Reject : le candidat n'est clairement pas une classe (ex : Catholicism)

- Unknown : en cas d'ambiguité ou si le candidat vous est inconnu, placez-le dans cette catégorie

Pour classer un élément dans une catégorie, ajoutez simplement un "1" dans la colonne appropriée et laissez les autres vides.

Dans le cas de la première évaluation 1-dboTypeCandidates, il y a plusieurs exemples pour chaque candidat (colonne E). Il ne faut voter qu'une fois par candidat (colonne A).

Dans le cas de la seconde évaluation 2-hypernymyCandidates, il y a pour chaque candidat à la fois la "relation inferrée", c'est à dire la relation de typage que l'on a déduite, et la phrase d'origine.

Voici quelques exemples de ce qu'il faut faire :

- Evaluation 2-hypernymyCandidates :

Candidat : "music_genre", exemple : "Reggae is a music genre ..." Il s'agit bien d'une classe, on accepte.

Candidat : "hydrogen", exemple : "Metallic hydrogen is a phase of hydrogen". Bien que la phrase puisse porter à confusion, l'hydrogène n'est pas une classe. On refuse.