le cnam

École Doctorale Informatique, Télécommunications et Électronique (Paris)

Centre d'Études et de Recherche en Informatique et Communications

THÈSE DE DOCTORAT

présentée par : **Subhi ISSA**

soutenue le : **13 Décembre 2019**

pour obtenir le grade de : **Docteur du Conservatoire National des Arts et Métiers**

Spécialité : **Informatique**

Linked Data Quality: Completeness and Conciseness

THÈSE DIRIGÉE PAR

Mme. SI-SAID CHERFI Samira — *Professeur des Universités, CNAM, Paris*

THÈSE CO-ENCADRÉ PAR

M. HAMDI Fayçal — *Maître de Conférences, CNAM, Paris*

RAPPORTEURS

M. SUCHANEK Fabian — *Professeur des Universités, Télécom ParisTech*
Mme. PERNELLE Nathalie — *Maître de Conférences, HDR, LRI-Université Paris Sud*

PRÉSIDENT

M. BARKAOUI Kamel — *Professeur des Universités, CNAM, Paris*

EXAMINATEURS

Mme. KEDAD Zoubida — *Maître de Conférences,HDR, UFR-Université de Versailles*
M. BELLATRECHE Ladjel — *Professeur des Universités, ISAE-ENSMA*
Mme. ZANNI-MERK Cecilia — *Professeur des Universités, INSA Rouen Normandie*

Abstract

The wide spread of Semantic Web technologies such as the Resource Description Framework (RDF) enables individuals to build their databases on the Web, write vocabularies, and define rules to arrange and explain the relationships between data according to the Linked Data principles. As a consequence, a large amount of structured and interlinked data is being generated daily. A close examination of the quality of this data could be very critical, especially, if important research and professional decisions depend on it. The quality of Linked Data is an important aspect to indicate their fitness for use in applications. Several dimensions to assess the quality of Linked Data are identified such as accuracy, completeness, provenance, and conciseness. This thesis focuses on assessing completeness and conciseness of Linked Data. In particular, we first proposed a completeness calculation approach based on a generated schema. Indeed, as a reference schema is required to assess completeness, we proposed a mining-based approach to derive a suitable schema (i.e., a set of properties) from data. This approach distinguishes between essential properties and marginal ones to generate, for a given dataset, a conceptual schema that meets the user's expectations regarding data completeness constraints. We implemented a prototype called "LOD-CM" to illustrate the process of deriving a conceptual schema of a dataset based on the user's requirements. We further proposed an approach to discover equivalent predicates to assess the conciseness of Linked Data. This approach is based, in addition to a statistical analysis, on a deep semantic analysis of data and on learning algorithms. We argue that studying the meaning of predicates can help to improve the accuracy of results. Finally, a set of experiments was conducted on real-world datasets to evaluate our proposed approaches.

Keywords: Semantic Web, Linked Data, Knowledge Graph, Data Quality, Data Completeness, Data Conciseness, RDF, Schema Mining, Quality Evaluation.

Résumé

La diffusion à large échelle des technologies du Web Sémantique telles que le *Resource Description Framework* (RDF) a permis aux individus de construire leurs bases de données sur le Web, d'écrire des vocabulaires et de définir des règles pour organiser et expliquer les relations entre les données selon les principes des Données Liées. En conséquence, une grande quantité de données structurées et interconnectées est générée quotidiennement. Un examen attentif de la qualité de ces données pourrait s'avérer très critique, surtout si d'importantes recherches et décisions professionnelles en dépendent. La qualité des Données Liées est un aspect important pour indiquer leur aptitude à être utilisées dans des applications. Plusieurs dimensions permettant d'évaluer la qualité des Données Liées ont été identifiées, telles que la précision, la complétude, la provenance et la concision. Dans cette thèse nous nous sommes focalisés sur l'étude de la complétude et de la concision des Données Liées. Dans un premier temps, nous avons proposé une approche de calcul de complétude fondée sur un schéma généré. En effet, comme un schéma de référence est nécessaire pour évaluer la complétude, nous avons proposé une approche fondée sur la fouille des données pour obtenir ce schéma (c.-à-d. un ensemble de propriétés). Cette approche permet de distinguer les propriétés essentielles des propriétés marginales pour générer, pour un jeu de données, un schéma conceptuel qui répond aux attentes de l'utilisateur par rapport aux contraintes de complétude souhaitées. Nous avons implémenté un prototype appelé "LOD-CM" pour illustrer le processus de dérivation d'un schéma conceptuel d'un jeu de données fondé sur les besoins de l'utilisateur. Dans un second temps, nous avons proposé une approche pour découvrir des prédicats synonymes afin d'évaluer la concision des Données Liées. Cette approche s'appuie, en plus d'une analyse statistique, sur une analyse sémantique approfondie des données et sur des algorithmes

d'apprentissage. Ces analyses permettent de mieux capter le sens des prédicats et ainsi améliorer la précision des résultats obtenus. Enfin, un ensemble d'expériences a été mené sur des jeux de données réelles afin d'évaluer nos différentes approches.

Mots clés : Web Sémantique, Données Liées, Graphe de Connaissances, Qualité des Données, Complétude des Données, Concision des Données, RDF, Extraction de Schéma, Evaluation de la Qualité.

Acknowledgements

I would be glad to clarify the effort of others which some may at first not notice also it is huge but it represents the hidden part of an iceberg in its size and importance.

First of all, I would like to thank my supervisor Prof. Samira Si-said Cherfi, for giving me the opportunity to follow my doctoral, and spending her time for helping me in various aspects during my research. I owe my most sincere gratitude to my co-supervisor Dr. Fayçal Hamdi for his scientific guidance, advice, encouraging and understanding that enabled me to complete my thesis.

I am grateful to Dr. Amrapali Zaveri for helping and guiding me during a period of my thesis, and to Mr. Onaopepo Adekunle for participating in a part of this research. Our remote meetings during several months shaped my research skills. It was a pleasure to work with professional researchers like them.

My sincere thanks go to my PhD reviewers Prof. Fabian Suchanek and Dr. Nathalie Pernelle who accepted to review and evaluate this dissertation. Their valuable feedbacks and remarks helped in improving the context. Many thanks to Prof. Kamel Barkaoui, Dr. Zoubida Kedad, Prof. Ladjel Bellatreche and Prof. Cecilia Zanni-Merk for being a part of my examining committee.

I thoroughly appreciate the good environment and support from my colleagues in ISID team and the other teams in Conservatoire National des Arts et Métiers, especially my close friends, whose friendship I am truly grateful for and especially for brainstorming sessions: Pierre-Henri, Quentin, Odette, Noura and Francesco. A big thank to my family in France for being always there, for hours spend in discussing research and personal issues: Yasser, Anas, Obaida, Yasmine, Noor, Raphaël, Bachar, Solimane and Radwan.

Last but not least, I am grateful to my parents Ahmad and Ahed for their unlimited love and advises to be who I am today, to my brother Zaid who had always a way to encourage me, and to my sisters Iman, Asma and Inas who stood with me and gave me strength all the time. To them, I would say thanks for the support and encouragement that let me feel they were physically close too. Special thanks to the love of my life Riham for sharing me all stressful and cheerful moments and being the source of tolerance and energy.

As I have shared them all with labour and stress, I find that it is fair to share them with success and cheerful, that they were a reason for.

Contents

1 Résumé en Français **17**

1.1 Contexte et objectifs 19

1.2 Contributions 23

1.3 L'organisation du manuscrit 24

2 Introduction **27**

2.1 Context and objectives 29

2.2 Contributions 32

2.3 Thesis outlines 33

3 Background and State-of-the-art **35**

3.1 Knowledge bases and Linked Data 36

3.1.1 Resource description framework 37

3.1.2 The SPARQL query language 39

3.2 Linked Data quality 40

3.3 State-of-the-art 41

3.3.1 Linked Data completeness: a systematic literature review 42

3.3.1.1 Systematic literature review methodology 43

3.3.1.2 Linked Data completeness analysis 47

3.3.1.3 Discussion 64

3.3.2 Linked Data conciseness 66
3.4 Summary 70

4 Assessing Completeness of RDF Datasets 73
4.1 Motivating scenario 73
4.2 Completeness computation: A mining-based approach 75
4.2.1 Properties mining 76
4.2.2 Completeness calculation 78
4.3 Prototype: LOD-CM 79
4.3.1 Overview 79
4.3.2 Conceptual schema derivation 80
4.3.3 Scope and completeness specification 82
4.3.4 UI description 83
4.3.4.1 A first iteration 83
4.3.4.2 A second iteration 85
4.3.5 Use cases 86
4.4 Summary 88

5 Assessing the Conciseness of Linked Datasets 91
5.1 Motivating scenario 91
5.2 Discovering synonym predicates 94
5.2.1 Phase 1: statistical analysis 94
5.2.2 Phase 2: semantic analysis 94
5.2.3 Phase 3: NLP-based analysis 102
5.3 Summary 103

6 Experimental Evaluation 105

6.1 Completeness dimension 105
6.1.1 Experimental setup 106
6.1.2 Completeness evolution over several versions of DBpedia 107
6.1.3 Generate schema from data values under completeness constrains 109
6.2 Conciseness dimension 113
6.2.1 Experimental setup 114
6.2.2 First experiment 116
6.2.3 Second experiment 118
6.3 Summary 120

7 Conclusion and Perspectives **121**
7.1 Thesis summary 121
7.2 Future directions 123

List of Publications **125**

Bibliography **127**

Index **141**

List of Tables

3.1 Number of the articles retrieved in each search engine. 48

3.2 List of the 52 core articles related to Linked Data completeness. 49

3.3 List of the 52 core articles classified according to the seven types. 50

3.4 List of completeness metrics. 67

4.1 A sample of triples from DBpedia. 77

4.2 Transactions created from triples. 77

4.3 DBpedia number of properties by classes and thresholds. 88

5.1 Six configurations of context and target [Abedjan and Naumann 2011]. . . 92

5.2 Facts in SPO structure from DBpedia. 93

5.3 Range content filtering. 93

5.4 Schema analysis. 93

6.1 Number of resources/class. 106

6.2 Statistics on the size of each category in both DBpedia and YAGO. 107

6.3 The completeness values and the number of properties for DBpedia categories at different minimum supports ξ. 110

6.4 The completeness values and the number of properties for YAGO categories at different minimum supports ξ. 110

6.5 Features predicates of DBpedia dataset (v10-2016). 115

List of Figures

1.1 Nuage des Données Liées 19

2.1 The Linked Open Data cloud 30

3.1 A simple RDF graph. 37

3.2 RDF graph representing different kinds of components. 38

3.3 Linked Data quality dimensions and the relationships between them. . . . 42

3.4 Overview of the systematic literature review methodology. 44

3.5 Classification of the 52 core articles by type of completeness. 47

3.6 Number of core articles by year. 48

3.7 Classification of articles by conferences and journals. 52

3.8 Summary of tools based on type of completeness. 61

3.9 An alignment between two ontologies [Shvaiko and Euzenat 2011]. 68

4.1 The *LOD-CM* workflow. 81

4.2 LOD-CM main interface. 83

4.3 List of proposed groups of maximal frequent itemsets. 84

4.4 The *Film* conceptual schema as a class diagram. 85

4.5 The *Artist* diagram class. 86

6.1 Completeness of equivalent resources from DBpedia v3.6, v2015-04 and v2016-10. 108

6.2 Completeness of DBpedia v3.6, v2015-04 and v2016-10. 109

6.3 Completeness values of DBpedia categories at different minimum supports ξ.111

6.4 The number of properties $|P|$ and the number of frequent patterns $|\mathcal{MFP}|$ of DBpedia categories at different minimum supports ξ. 112

6.5 Completeness values of YAGO categories at different minimum supports ξ. 112

6.6 The number of properties $|P|$ and the number of frequent patterns $|\mathcal{MFP}|$ of YAGO categories at different minimum supports ξ. 113

6.7 F1-measure values at each phase based on support threshold. 117

6.8 Number of candidate pair at each phase. 118

6.9 Recall value based on support threshold values. 119

Chapter 1

Résumé en Français

L'histoire du Web a connu trois phases essentielles depuis son invention au début des années 1990. Dans sa toute première version, le Web était accessible à la fois en lecture et en écriture par tout un chacun. Toutefois, très rapidement, la possibilité de modifier les pages Web a été supprimée, le Web devenant ainsi statique, puis cette possibilité de modifier les pages Web a été en redécouverte avec l'apparition des blogs et wikis puis des réseaux sociaux. C'est ce qui est appelé parfois le Web 2.0 ou le Web social. Tim Berners-Lee, inventeur du Web, a ensuite voulu permettre une utilisation plus facile, par des machines, de ces données publiées sur le Web en proposant le Web sémantique [Berners-Lee, Hendler, Lassila, et al. 2001]. Ainsi ces données sont plus facilement échangeables et traitées de manière automatique.

Le Web sémantique est piloté par une organisation internationale qui s'engage à améliorer le Web, qui s'appelle Consortium Web (W3C)[1]. Le W3C a créé des standards pour atteindre le but du Web sémantique. Comme le Web contient des ressources telles que des documents textuels ou des images représentant des lieux, des personnes et des événements, ces standards, appelés "technologies du Web sémantique", sont essentiels pour faciliter le traitement, l'accès, la validation et la gestion automatique des ressources Web. De plus, l'objectif du Web sémantique n'est pas seulement d'atteindre des données, mais aussi d'établir des liens entre elles. Cet ensemble de données interdépendantes sur le Web s'appelle Données Liées (*Linked Data*) [Berners-Lee 2006b].

[1] `https://www.w3.org/standards/semanticweb/`

Le développement des technologies du Web sémantique telles que *Resource Description Framework* (RDF) a conduit à un volume important de données publiées sur le Web sous forme de données ouvertes liées (LOD) [2] [Berners-Lee 2006a]. Cependant, l'évolution rapide des bases de connaissances (KBs) exposées en tant que Données Liées, comme dans le nuage LOD[3], ne respectant pas en général toutes les recommandations [Debattista, Lange, Auer, and Cortis 2017], conduit à une variété de problèmes de qualité à différents niveaux, comme au niveau du schéma ou de l'instance. Une étude empirique menée par Debattista et al. [Debattista, Lange, Auer, and Cortis 2017] montre que les jeux de données publiés dans le nuage LOD ont une qualité globale raisonnable, mais qu'il subsiste des problèmes importants concernant certaines dimensions de la qualité, comme la provenance et la complétude des données. Par conséquent, en mettant l'accent sur une dimension en particulier telle que la complétude, nous pouvons explorer certains aspects de la détection et de l'atténuation de la question de la qualité des données d'une manière plus approfondie, par exemple, déterminer si la complétude est mieux abordée dans les processus de collecte ou d'intégration des données.

Les données publiées dans le nuage LOD sont publiées selon les quatre principes essentiels suivants, connus sous le nom de "principes de Données Liées" [Bizer, Heath, and Berners-Lee 2011] : (i) identifier chaque élément à l'aide d'identificateurs de ressources uniformes (URIs), (ii) utiliser HTTP pour représenter les URIs, et donc avoir la possibilité d'atteindre les éléments identifiés, (iii) fournir des informations utiles en utilisant les standards (RDF, SPARQL) lors de la recherche d'une URI, (iv) établir des liens vers d'autres URIs pour permettre de découvrir plus de données. Plusieurs jeux de données publiés dans le Web selon ces quatre principes tels que DBpedia [Auer, Bizer, Kobilarov, Lehmann, Cyganiak, and Ives 2007], YAGO [Suchanek, Kasneci, and Weikum 2007b] et Wikidata [Vrandečić and Krötzsch 2014] ont des caractéristiques de qualité différentes. En effet, ces principes offrent une infrastructure pour publier et interconnecter des données dans le Web, sans garantir une quelconque qualité sur leur utilisation [Bechhofer, Buchan, De Roure, Missier, Ainsworth, Bhagat, Couch, Cruickshank, Delderfield, Dunlop, et al.

[2] http://linkeddata.org/
[3] http://lod-cloud.net/

2013]. Le nuage LOD contient 1,239 jeux de données avec 16,147 liens entre eux[4] et des milliards de faits qui couvrent plusieurs domaines tels que la géographie, les données gouvernementales, les réseaux sociaux, etc. La figure 1.1 montre la complexité et la variété des données ouvertes liées de nos jours. Ce volume de données soulève la question de leur qualité. L'inconvénient, c'est que plus nous avons de données, plus nous sommes confrontés à des problèmes de qualité.

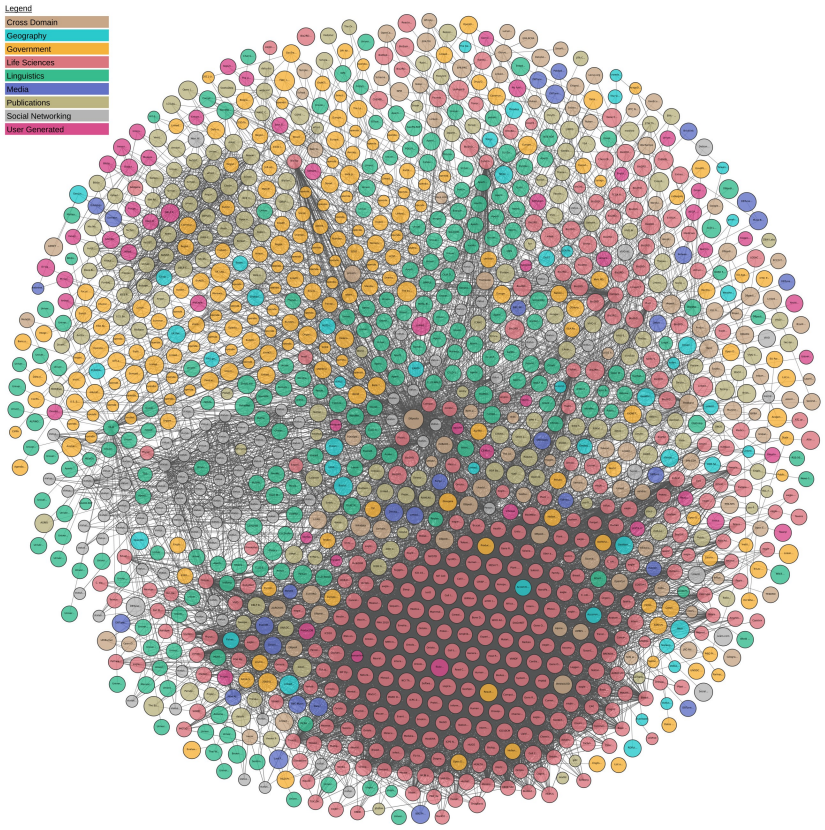

Figure 1.1: Nuage des Données Liées.[5]

1.1 Contexte et objectifs

Avant même l'apparition des Données Liées, la question de la qualité des données a représenté un défi pour les systèmes d'information traditionnels et a pointé la nécessité de mener des études rigoureuses pour assurer une qualité adéquate des données stockées

[4]Extrait le 22 juin 2019 de `https://lod-cloud.net/`
[5]https://lod-cloud.net/clouds/lod-cloud-sm.jpg

dans les bases de données relationnelles. En proposant différentes approches, ces études ont eu un impact positif sur les processus organisationnels utilisés pour les bases de données relationnelles [Scannapieco and Batini 2004; Motro and Rakov 1998]. Ainsi, la réutilisation de ces approches dans le contexte du Web sémantique permet de tirer parti de l'expérience acquise avec les systèmes d'information traditionnels. Étant donné que la qualité élevée des données garantit leur aptitude à l'emploi [Juran, Gryna, and Bingham 1974] dans un large éventail d'applications, il est très important de disposer des bons paramètres pour évaluer et améliorer la qualité des Données Liées. Plusieurs cadres et approches ont été proposés pour évaluer les différentes dimensions de la qualité des Données Liées.

Le terme "qualité", communément décrit comme *aptitude à l'usage* [Juran, Gryna, and Bingham 1974], inclut plusieurs dimensions telles que l'exactitude, la cohérence et la complétude. Néanmoins, ces dimensions sont parfois subjectives, comme, par exemple, le fait que les données sont suffisantes pour un usage, mais par un autre.

Plusieurs études ont proposé différentes définitions et classification des dimensions de qualité [Wand and Wang 1996; Wang and Strong 1996; Naumann 2003]. L'une des plus utilisées est celle proposée par [Wand and Wang 1996], qui classe les dimensions de qualité de données en quatre groupes fondamentaux. Cette classification dépend de la façon dont les données peuvent être mesurées :

- Dimensions intrinsèques : qui peuvent être mesurées par des métriques qui évaluent la validité et la cohérence des données, comme *l'exactitude*.
- Dimensions contextuelles : où la qualité des données doit être considérée dans le contexte d'une tâche en particulier, comme *la véracité*.
- Dimensions représentationnelles : qui comprennent des mesures liées à la conception des données ; en d'autres termes, la qualité de la représentation des données, comme *l'interopérabilité*.
- Dimensions d'accessibilités : qui sont liées à la facilité d'accès et à la sécurité des données, comme *la performance*.

En 2016, Zaveri et al., ont défini 18 dimensions de qualité [Zaveri, Rula, Maurino, Pietrobon, Lehmann, and Auer 2016] qui peuvent être utilisées pour évaluer la qualité des Données Liées. Dans cette thèse, nous nous intéressons à deux dimensions importantes qui sont **la complétude** et **la concision**. Ce choix est motivé par le fait que ces deux dimensions affectent d'autres dimensions de la qualité des données, telles que l'exactitude, et ont un impact direct sur les requêtes utilisateurs.

Complétude :

La complétude des données est reconnue comme une des dimensions de qualité les plus importantes[Margaritopoulos, Margaritopoulos, Mavridis, and Manitsaris 2012a]. Selon [Mendes, Mühleisen, and Bizer 2012b], la complétude des données est le fait de savoir à quel point un jeu de données contient toutes les données nécessaires pour une tâche en particulier. Par conséquent, il est presque impossible de définir une approche universelle pour mesurer la complétude des données, car un jeu de données peut convenir à un usage, mais pas à un autre. Une façon traditionnelle de mesurer la complétude dans ce cas est le taux de valeurs manquantes. Cela suppose que les données reposent sur un schéma convenu et bien conçu dans lequel les attributs du schéma sont également pertinents. Le problème est que l'évaluation de la complétude en tant que taux de valeurs manquantes ne tient pas compte du fait que l'absence de valeurs pour un attribut marginal devrait avoir moins d'incidences sur la complétude qu'un autre attribut considéré comme essentiel. Dans le contexte du Web sémantique, le défi est de trouver, pour un jeu de données, un schéma approprié qui peut être considéré comme un schéma de référence. Pour résoudre ce problème, il est nécessaire d'explorer les instances pour avoir une idée de la façon dont elles sont réellement décrites et de l'importance des attributs et propriétés utilisés dans leurs descriptions.

Example 1.1.1. *Tout le monde a un nom, un lieu de naissance et une date de naissance. Ce fait devrait être reflété dans les données, où chaque instance de personne devrait avoir toutes ces propriétés (c.-à-d. nom, lieu de naissance et date de naissance). Cependant, si on considère la propriété "lieu de décès", cette denière peut être absente dans la description de plusieurs personnes (qui ne sont pas encore décédés).*

Ainsi, si une propriété *p* est suffisamment utilisée parmi un ensemble d'instances d'une classe *T*, nous considérons que cette propriété *p* est une sorte de propriété importante ou qu'elle existe effectivement même si elle n'est pas présente dans toutes les instances de cette classe *T*. Par conséquent, si *p* est manquante pour une instance de *T*, nous pouvons raisonnablement supposé que ceci diminue la complétude du jeu de données.

Concision :

Les Données Liées sont généralement collectées à partir de sources hétérogènes à l'aide d'une variété d'outils à des fins différentes [Mika 2005; Lei, Sabou, Lopez, Zhu, Uren, and Motta 2006]. En conséquence, une sorte de redondance de données peut se produire lorsque les mêmes données sont collectées. Pour être concis, les jeux de données doivent éviter les répétitions entre des éléments ayant la même signification avec des identificateurs ou des noms différents. Cette dimension de qualité est très importante, car, d'une part, le fait d'avoir des données inutiles peut avoir une influence négative sur l'exactitude des informations et, d'autre part, une extension de requête est nécessaire pour extraire toutes les informations dont un utilisateur aurait besoin.

Dans les Données Liées, la concision peut être calculée au niveau du schéma ou bien au niveau des instances [Mendes, Mühleisen, and Bizer 2012b]. Au niveau du schéma, un jeu de données est concis s'il n'existe pas de classes ou de prédicats équivalents avec des noms différents. Cependant, au niveau des instances, un jeu de données est considéré comme concis s'il n'existe pas des instances équivalentes avec des noms différents.

Example 1.1.2. *La requête suivante retourne la liste des personnes nées en France :*

Listing 1.1 – Exemple d'une requête

```
SELECT * WHERE {
        ?someone type Person .
        ?someone name ?name .
        ?someone birthPlace France .
        }
```

Supposons les faits suivants : <Antoine_Griezmann birthPlace France> et <Emma_Watson bornIn France>. Cette requête ne retourne pas les instances qui ont comme type Person, mais qui utilisent

à la place de birthPlace, le prédicat synonyme bornIn. Pour résoudre ce problème, la requête doit être étendue à l'aide de l'opérateur UNION pour récupérer toutes les instances qui utilisent bornIn ou un autre synonyme.

Ce processus peut devenir trop compliqué si nous avons (i) beaucoup de synonymes, et (ii) aucune idée des éléments de l'ontologie.

Dans cette thèse, nous abordons l'évaluation de la concision par la découverte de prédicats équivalents dans un jeu de données. Pour atteindre cet objectif, nous proposons une approche en trois phases qui s'appuie, en plus d'une analyse statistique et du traitement automatique du langage naturel, sur une analyse sémantique par l'étude de la signification de chaque prédicat pour détecter d'éventuels conflits logiques.

1.2 Contributions

Les contributions principales de cette thèse sont les suivantes :

- Nous avons effectué une revue systématique de la littérature pour identifier tous les articles liés à la complétude des Données Liées. 52 articles qui portent sur sept types de complétude liée à différents problèmes ont été analysés. Pour chaque type nous avons fourni une définition et les approches et mesures utilisées ainsi que les outils implémentés pour évaluer la complétude.

- Nous avons proposé une approche pour l'évaluation de la complétude des Données Liées au niveau schéma. Comme cette dimension de qualité repose souvent sur des schémas de données de référence qui ne sont pas toujours disponibles ni réalistes d'un point de vue pratique, nous avons proposé d'utiliser des approches de fouille de données pour déduire un schéma à partir des données. L'approche proposée est un processus en deux étapes : premièrement, extraire à partir du jeu de données un schéma qui reflète la représentation réelle des données qui, dans un deuxième temps, est utilisée pour l'évaluation de la complétude.

- Nous avons développé un prototype, appelé *LOD-CM*[6], pour démontrer comment

[6] `http://cedric.cnam.fr/lod-cm/`

notre approche révèle des schémas conceptuels de référence basés sur les exigences et contraintes des utilisateurs. L'objectif de ce prototype est d'aider l'utilisater à comprendre le schéma et à découvrir les propriétés associées. Le schéma généré inclut la catégorie choisie et les attributs et les relations entre données marquées par des valeurs de complétude. Nous avons présenté deux cas d'utilisation, liés à la découverte de schémas de référence en fonction des besoins de l'utilisateur, qui montre l'utilité du prototype.

- Nous avons proposé une approche pour évaluer la concision d'un jeu de données en découvrant des prédicats synonymes. Notre approche inclut trois phases : (1) l'analyse statistique pour obtenir un ensemble initial de prédicats synonymes, (2) des analyses sémantiques pour détecter d'éventuelles incohérences et (3) des analyses basées sur des techniques de Traitement Automatique du Langage Naturel (TALN) et des algorithmes d'apprentissage. L'objectif des deux dernières phases et d'améliorer la précision de l'identification des synonymes en éliminant les faux postives.

- Nous avons réalisé un ensemble d'expériences sur différents jeux de données pour évaluer les approches que nous avons proposés. Pour la dimension de complétude, nous avons analysé l'évolution de la complétude sur des versions espacées dans le temps de DBpedia. Ensuite, nous avons évalué l'efficacité de la mesure de complétude par rapport au nombre d'instances et les seuils spécifiés par l'utilisateur sur DBpedia et YAGO. Enfin, pour la dimension de concision, nous avons présenté deux expériences réalisées sur les deux mêmes jeux de données pour identifier les prédicats synonymes.

1.3 L'organisation du manuscrit

Ce manuscrit est composé de sept chapitres dont un chapitre de résumé en français, et un chapitre introduction.

Les cinq autres chapitres sont organisés comme suit :

- **Le Chapitre 3** présente quelques concepts de base qui sont nécessaires pour introduire

les contributions de ce travail. Ensuite, il passe en revue les approches existantes qui traitent le problème d'évaluation de la complétude et de la concision des Données Liées.

Ce chapitre contient deux parties principales. La première partie (les sections 3.1 and 3.2) donne un aperçu général de certaines des technologies du Web sémantique utilisées dans cette thèse, ainsi qu'un aperçu sur la qualité des Données Liées. La deuxième partie (section 3.3), présente notre stratégie de recherche qui consiste à effectuer une revue systématique de la littérature afin de rassembler tous les travaux pertinents sur la complétude des Données Liées. Ensuite, elle fournit une description des travaux connexes dans le domaine de la concision des Données Liées. Enfin, le chapitre conclut par un résumé sur le positionnement de notre travail par rapport à l'état de l'art.

- **Le Chapitre 4** présente notre première approche qui porte sur la découverte d'un schéma afin de calculer la complétude d'une base de connaissances publiée dans le Web de données.

  Ce chapitre contient trois parties principales. La première partie (section 4.1) présente un scénario de motivation. Ensuite la section 4.2, explique notre approche qui comprend deux étapes : la fouille de données et le calcul de complétude des propriétés. Enfin, la section 4.3 présente notre prototype *LOD-CM* pour révéler automatiquement, en considérant des contraintes spécifiées par l'utilisateur, des schémas conceptuels de référence pour un jeu de données RDF et les représenter sous forme de diagrammes UML.

- **Le Chapitre 5** présente notre deuxième approche qui porte sur l'évaluation de la concision des jeux de Données Liées.

  Ce chapitre est structuré comme suit. Tout d'abord la section 5.1 décrit le problème des prédicats synonymes dans le contexte des jeux de Données Lieés. Ensuite, la section 5.2 présente notre approche pour découvrir des prédicats synonymes qui est composée de trois phases : (i) une analyse statistique pour obtenir un ensemble initial de prédicats synonymes, (ii) une analyse sémantique qui tire profit des axiomes

du langage OWL2 (spécifiquement, les types de relation entre concepts telles que *functional, transitive, cardinality, etc.*) et (iii) une analyse basée sur des techniques TALN qui calcule, à l'aide d'algorithmes d'apprentissage, des similarités entre les vecteurs contextuels représentant les prédicats candidats.

- **Le Chapitre 6** présente les différentes expérimentations que nous avons effectuées pour évaluer nos approches.

  Ce chapitre comporte deux parties principales. Premièrement, pour l'évaluation de la complétude la section 6.1 présente une première expérimentation effectuée sur trois versions différentes de la base de connaissances DBpedia (DBpedia v3.6, DBpedia v2015-04 et DBpedia v2016-10). Une seconde expérimentation effectuée sur DBpedia et YAGO est présentée qui montre comment notre approche se comporte avec différents paramètres. Deuxièmement, pour l'évaluation de la concision, la section 6.2 présente des expériences sur DBpedia et YAGO, qui montrent dans quelle mesure notre approche améliore la précision de la découverte de prédicats synonymes.

- **Le Chapitre 7** résume nos contributions et donne quelques perspectives pour des orientations futures.

  Ce chapitre rappelle nos objectifs et résume nos différentes propositions et les résultats des évaluations expérimentales. Il présente également quelques directions futures: (1) l'extestion de la revue systématique de la littérature par l'ajout de nouveau mots clés tels que "Graphe de connaissances", (2) évaluer les approches sur d'autres jeux de données tels que Wikidata ou IMDb, (3) améliorer le prototype LOD-CM, par la prise en compte plusieurs jeux de données permettant ainsi à l'utilisateur de choisir le jeu qui répond le plus à ses besoins, (4) la prise en compte, dans l'approche d'évaluation de la concision, des prédicats qui sont très peu utilisés dans un jeu de données et qui fausse les résultats d'identification des prédicats synonymes.

Chapter 2

Introduction

The history of the World Wide Web has gone through three essential phases since its invention in the early 1990s. Basically, it all began with Web 1.0 including read-only static pages where it is not possible for users to interact with these pages. Later, Web 2.0 appeared as the second generation of the Web. Unlike Web 1.0, the second phase facilitated interaction between web pages and users. It enabled the user not only to access to information but also to create it, such as *blogs* and *Wikipedia*. Web 3.0, which is the future, is going to become more *intelligent* by adding *meaning* to data published on the Web. This extension, that uses Semantic Web technologies [Berners-Lee, Hendler, Lassila, et al. 2001], aims to allow machines to automatically create, exchange, and link data based on the ability to understand the meaning of the Web content. Therefore, machines can interpret information and take decisions. The Semantic Web is driven by an international organization committed to improve the Web, that is called World Wide Web Consortium (W3C)[7]. The W3C created standards to achieve the aim of Semantic Web. As Semantic Web contains resources representing objects such as documents or images as well as objects such as places, people and events, these standards that are called Semantic Web technologies are essential to treat, access, validate and manage a huge amount of data. Moreover, the objective of Semantic Web is not merely reaching data but also establishing links between them. This set of interrelated data on the Web is called Linked Data [Berners-Lee 2006b].

The development of Semantic Web technologies such as the Resource Description

[7] https://www.w3.org/standards/semanticweb/

Framework (RDF)[8] has led to an admirable volume of data published on the Web as Linked Open Data (LOD)[9] [Berners-Lee 2006a]. The collection and publication of such vast amounts of data into Knowledge Bases (KBs) is certainly a progression in the right direction towards the *Web of Data*. However, the evolution of KBs exposed as Linked Data such as in the LOD Cloud[10] is generally unrestrained [Debattista, Lange, Auer, and Cortis 2017], which leads to a variety of quality issues at various levels such as at the schema or the instance level. An empirical study carried out by Debattista et al. [Debattista, Lange, Auer, and Cortis 2017] shows that datasets published in the LOD cloud have a reasonable overall quality, but significant issues remain concerning some quality dimensions, such as data provenance and completeness. Therefore, by focusing on an individual dimension such as completeness, we can explore certain aspects of detecting and mitigating the completeness data quality issue in a more thorough fashion, for example, exploring whether completeness is better tackled in the data collection or integration processes.

Data published in the LOD cloud are represented regarding the following four essentials known as "Linked Data Principles" [Bizer, Heath, and Berners-Lee 2011]: (i) each item should be identified using Uniform Resource Identifiers (URIs), (ii) using HTTP to represent URIs, therefore it can be possible to find the identified items, (iii) providing useful information using the standards (RDF, SPARQL) during looking up a URI, (iv) linking to other URIs should be included to enable discovering more data. Several datasets published according to these Linked Data principles such as DBpedia [Auer, Bizer, Kobilarov, Lehmann, Cyganiak, and Ives 2007], YAGO [Suchanek, Kasneci, and Weikum 2007b] and Wikidata [Vrandečić and Krötzsch 2014] are with various quality features. Indeed, these principles offer an infrastructure for publishing and navigating through data without ensuring the quality of its usage [Bechhofer, Buchan, De Roure, Missier, Ainsworth, Bhagat, Couch, Cruickshank, Delderfield, Dunlop, et al. 2013]. Linked Open Data (LOD) cloud contains 1,239 datasets with 16,147 links between them[11] and billions of facts that cover several domains such as geography, government, social networking, etc. Figure 2.1 shows the complexity and variety of Linked Open Data nowadays. This

[8] `https://www.w3.org/RDF/`
[9] `http://linkeddata.org/`
[10] `http://lod-cloud.net/`
[11] Retrieved June 22, 2019 from `https://lod-cloud.net/`

volume of data raises the question about its quality. The downside is that the more data we have, the more quality issues we face.

2.1 Context and objectives

Data quality issue is also a challenge for traditional information systems, hence, rigorous researches on ensuring adequate quality of data in relational databases have been carried out, even before the onset of Linked Data. By proposing different approaches, these researches have led to a positive impact on the organizational processes for relational databases [Scannapieco and Batini 2004; Motro and Rakov 1998]. Thus, the applicability of these approaches in the context of Web of Data provides an avenue to leverage experience gained from traditional information systems. Since high quality of data ensures its fitness for use [Juran, Gryna, and Bingham 1974] in a wide range of applications, having the right metrics to assess and improve the quality of Linked Data is of great importance. Several frameworks and approaches have been proposed to evaluate varying dimensions of Linked Data quality.

The term "quality" is commonly described as *fitness for use* [Juran, Gryna, and Bingham 1974] which includes several dimensions such as accuracy, timeliness, consistency, correctness, and completeness. Nevertheless, these dimensions are subjective such as completeness which implies that data is sufficient for the consumer's needs. In 2016, Zaveri defined 18 different quality dimensions [Zaveri, Rula, Maurino, Pietrobon, Lehmann, and Auer 2016] that can be applied to assess the quality of Linked Data. In this thesis, we are interested in two dimensions of Linked Data quality that are completeness and conciseness. This choice is motivated by the fact that these two dimensions affect other data quality dimensions, such as accuracy, and impact directly users queries.

Completeness:

Data completeness is recognized as an important quality dimension [Margaritopoulos, Margaritopoulos, Mavridis, and Manitsaris 2012a]. Moreover, providing completeness

[12]https://lod-cloud.net/clouds/lod-cloud-sm.jpg

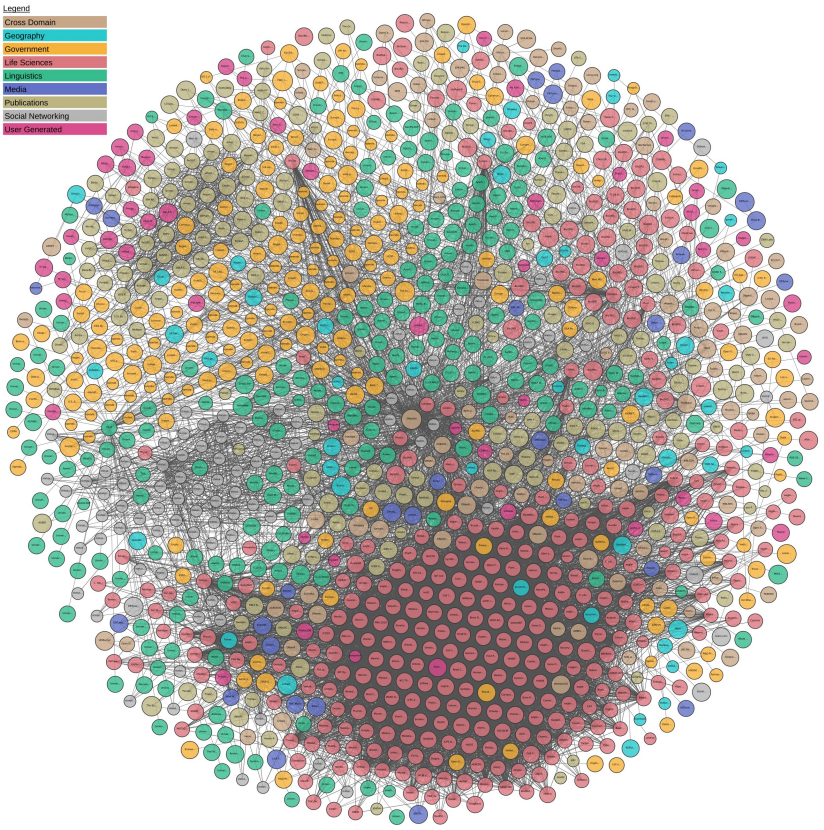

Figure 2.1: The Linked Open Data Cloud.[12]

information about a data source helps increasing the confidence of such a source. According to [Mendes, Mühleisen, and Bizer 2012b], data completeness is about to know to which degree a dataset contains all of the necessary objects for a given task. As a consequence, a dataset may fit for a usage but not for another, and it is almost impossible to have an absolute standard to measure completeness. A traditional way to measure completeness in this case is the rate of missing values. This subsumes that data relies on an agreed and well-designed schema in which schema attributes are equally relevant. Indeed, assessing completeness as the rate of missing values does not take into account the fact that missing a marginal attribute should be less important than missing an essential one. In the context of Semantic Web, the challenge is to find a suitable schema to consider it as a reference schema. To overcome this issue, there is a need to explore instances to get an idea about how they are actually described and which properties, with the importance of each one, are used.

Example 2.1.1. *All people have a name, a birth place and a date of birth. Thus, this should be reflected in the data, each person instance should have all these properties. But not all people have a death place (at least not yet), consequently this property may not be found in all person instances.*

Hence, if a property p is sufficiently used among a set of instances of the same class T, we postulate that this property p is a kind of mandatory or it exists actually even if it is not present in the dataset for all instances of this class T. Therefore, if p is missing for an instance of T, we may reasonably argue that this decreases the completeness of the dataset.

Based on data mining techniques, the approach that we proposed in this thesis deals with this issue by extracting, from a set of instances sharing the same type, the set of the most representative properties and calculating completeness in respect to this set

Conciseness:

Semantic data is usually collected from heterogeneous sources through a variety of tools for different purposes [Mika 2005; Lei, Sabou, Lopez, Zhu, Uren, and Motta 2006]. As a consequence, we get wasteful data redundancy that occurs when a given piece of data should not be repeated. To enhance the conciseness, dataset should avoid repetition through elements having the same meaning with different identifiers or names. This dimension is highly important because, on the one hand, having useless data may influence the accuracy of information negatively, and, on the other hand, a query expansion is required to extract all needed information. In Linked Data, conciseness is categorized into two levels [Mendes, Mühleisen, and Bizer 2012b]. At schema level, a dataset is concise if there are no equivalent classes or predicates with different names. However, at data level, a dataset is considered as concise if there are no equivalent instances with different names.

Example 2.1.2. *The following query returns a list of people born in France:*

Listing 2.1 – Example of a query

```
SELECT * WHERE {
        ?someone type Person .
        ?someone name ?name .
        ?someone birthPlace France .
        }
```

Suppose that we have the following facts: <Antoine_Griezmann birthPlace France> and <Emma_Watson bornIn France>. This query fails to return instances that have as type Person but use synonym predicates of birthPlace such as bornIn. To overcome this problem, the query needs to be expanded using UNION operator to match additional instances. This process may become too complicated in case of we have (i) many synonyms, and (ii) no idea about the vocabularies of the ontology.

In this thesis, we address the assessing of conciseness through the discovering of equivalent predicates of a given dataset. In order to achieve this objective, we propose an approach that consists of three sequential phases. It is based in addition to a statistical and NLP-based analysis, on a semantic analysis through studying the meaning of each predicate to detect logical conflicts.

2.2 Contributions

The contributions of this thesis are as follows:

1. We conduct a Systematic Literature Review to identify all articles related to Linked Data completeness. Then, the proposed solutions are summarized and classified according to the addressed problem, proposed approaches and metrics, and developed tools to assess the issue of completeness.

2. We propose an approach for Linked Data completeness assessment. As this quality dimension often relies on gold standards and/or a reference data schema that are neither always available nor realistic from a practical point of view, we propose to use mining approaches to infer a suitable schema from data values. The proposed approach is a two-step process: first, mining from a dataset a schema that reflects the actual representation of data, which, in the second step, is used for completeness evaluation.

3. We develop a prototype, called LOD-CM[13], to demonstrate how our approach reveals reference conceptual schemas based on user's requirements and constraints. The

[13] `http://cedric.cnam.fr/lod-cm/`

generated schema includes the chosen category, the attributes and the relationships tagged by completeness values.

4. We provide an approach for assessing the conciseness of a dataset by discovering equivalent predicates. Our method consists basically of three sequential phases to detect equivalent predicates. Beside the statistical analysis to obtain an initial set of synonyms predicates, semantic and NLP-based analyses are performed in order to improve the precision of the identification of synonyms.

5. A set of experiments is fulfilled with one or more datasets to assess our proposed approaches. For the completeness dimension, we analyze the evolution of completeness quality over timely spaced versions of DBpedia. Then, we evaluate the completeness measure robustness regrading the number of instances and user-specified thresholds on real-world datasets, DBpedia and YAGO. For the conciseness dimension, we present two experiments performed on both datasets to identify synonym predicates.

2.3 Thesis outlines

The thesis is structured as follows:

Chapter 3 introduces some basic notions and discusses related works. The first part of this chapter gives a general background about some of Semantic Web technologies used in this thesis, and an overview of Linked Data quality. the second part shows our research strategy to conduct a systematic literal review to gather all relevant works on Linked Data completeness. In addition to that, it provides related works in the area of Linked Data conciseness. Finally, it shows what our work brings as contributions to the state-of-the-art.

Chapter 4 describes our first contribution to discover schema in order to calculate the completeness of a dataset. First, we start with a motivation scenario. Then, we explain our mining-based approach that consists of two steps: properties mining and completeness calculation. Furthermore, we present our prototype *LOD-CM* to reveal automatically, using a user-specified threshold, conceptual schemas from RDF data sources and represent it as an UML diagram.

Chapter 5 presents our proposed method to evaluate the conciseness of an RDF dataset. First, we discuss briefly the problem of equivalent predicates for linked datasets. After that, we describe our approach to discover equivalent predicates which consists of three phases: (i) statistical analysis, (ii) semantic analysis and (iii) NLP-Based analysis.

Chapter 6 presents the experimental evaluations that we performed to assess our approaches. First, for the assessing of the completeness, we perform an experimental evaluation on three different versions of the DBpedia knowledge base (DBpedia v3.6, DBpedia v2015-04 and DBpedia v2016-10). Then, we analyze how our approach behaves with several varied parameters on DBpedia and YAGO. Secondly, for the assessing of the conciseness, we provide experiments on DBpedia and YAGO, that show how our approach improve the precision of discovering equivalent predicates.

Chapter 7 summarizes our contributions and gives some perspective points for future directions.

Chapter 3

Background and State-of-the-art

In this chapter, we present some basic concepts that are required to introduce the contributions of this work. Then, we survey existing approaches that address the assessment of Linked Data completeness and conciseness. The chapter is structured as follows: in Section 3.1 we give a brief description of Knowledge Bases that uses Resource Description Framework (RDF) to organize data, and we explain how to query these data using SPARQL. In Section 3.2 we illustrate, in a nutshell, different dimensions of Linked Data quality. In Section 3.3 we discuss two points: First, we present our methodology for conducting a systematic literature review (SLR) on Linked Data completeness. In this part, we gather existing approaches from the literature and analyze them qualitatively and quantitatively. We have distinguished between schema, property, population, interlinking, currency, metadata and labelling completeness as detailed in Section 3.3.1. Second, we discuss related works about the assessment of Linked Data conciseness. Conciseness is classified into schema and instance level [Mendes, Mühleisen, and Bizer 2012b] to measure redundant attributes and objects, respectively. The reason why we choose to perform an SLR only on the completeness dimension is motivated by the fact that the number of approaches that study the conciseness of RDF datasets are very limited (cf. 3.3.2). Finally, we conclude this chapter by summarizing the related works and explaining our contributions regarding these works illustrated in Section 3.4.

3.1 Knowledge bases and Linked Data

A Knowledge Base (KB) is a database used for knowledge exchanging and management, where the contained information is stored, organized and shared. In the Semantic Web community, this information (i.e., facts) on the Web should be published and queried in a semantically structured way [Färber, Bartscherer, Menne, and Rettinger 2018].

Linked Data (LD) is a familiar pattern of the Semantic Web. Tim Berners-Lee proposed a five-star scheme to publish good Linked Data KBs [Berners-Lee 2006a]. Each star represents a level of scheme development, and includes the features of the previous star. The "five-star scheme" has been suggested for measuring how this usability is achieved, it is described as follows:

- $*$ data is published on the Web using non-specific formats, e.g., PDF under free license.
- $**$ data is published as structured data formats, e.g., Excel.
- $***$ data is published as non-proprietary formats, e.g., CSV.
- $****$ data is published using universal formats, e.g., RDF.
- $*****$ data is linked to other datasets to improve the context.

It is important to have the features of the fourth or fifth star in order to have an easy detectable and exchangeable data. This requires publishing data in RDF, providing access points in SPARQL and interlinking data. The structured datasets in KB are generated in any format that is fit to Linked Data principles [Berners-Lee 2006a] to unify data from various sources. This thesis focuses on RDF format as it is widespread used to publish Linked Data, and the standard query language for RDF is SPARQL [Harris, Seaborne, and Prud'hommeaux 2013]. In the next subsections, we describe RDF that is used to represent KBs and SPARQL used to query RDF datasets through the matching SPARQL endpoint.

3.1.1 Resource description framework

The Resource Description Framework (RDF) [RDF 2014] provides a general syntax to represent information available on the Web from diverse data sources. It was published by the RDF Working Group as a W3C recommendation [Lassila and Swick 1999]. Every piece of information expressed in RDF is structured as a <*subject*,*predicate*,*object*> form called *RDF triple*, *triple* or *statement*. A statement relates one resource (subject) to a value (object) via a predicate such as <*Adam*,*hasSister*,*Suzanna*>. A collection of triples forms an RDF graph where each triple is a directed node-edge-node portion of the graph. In this graph, the subject and object of a triple are mapped to two nodes and the predicate is mapped to a labeled arc (i.e., an edge points from one node to another node) linking the subject and the object nodes of that graph as shown in Figure 3.1.

Figure 3.1: A simple RDF graph.

Adam is the subject that is the source node, *Suzanna* is the object that is the target node, and *hasSister* is the relationship between the two nodes called a predicate. There are three different kinds of RDF components:

- IRI: Internationalized Resource Identifier is a compact string for identifying uniquely real-world entities and their relationships. IRI can point not only to anything on the Web but also to objects that are not part of the Web such as abstract concepts (e.g., family relationships or logical operators).

- Literal: A sequence of characters such as string, numbers, and dates. Literal are used to identify values such as an ISBN[14] or the birth date of a person.

- Blank node (*bnode*): A node in RDF graph with neither IRI nor literal that can appear on the subject and object position of a triple. It is appropriate to refer the nodes that are locally used.

[14] An ISBN is an International Standard Book Number.

Only the target node, which is the object in RDF triple, can have literal value, IRI or bnode. However, predicate cannot be a bnode.

An example for an RDF graph with IRI, literal and blank node is given in Figure 3.2.

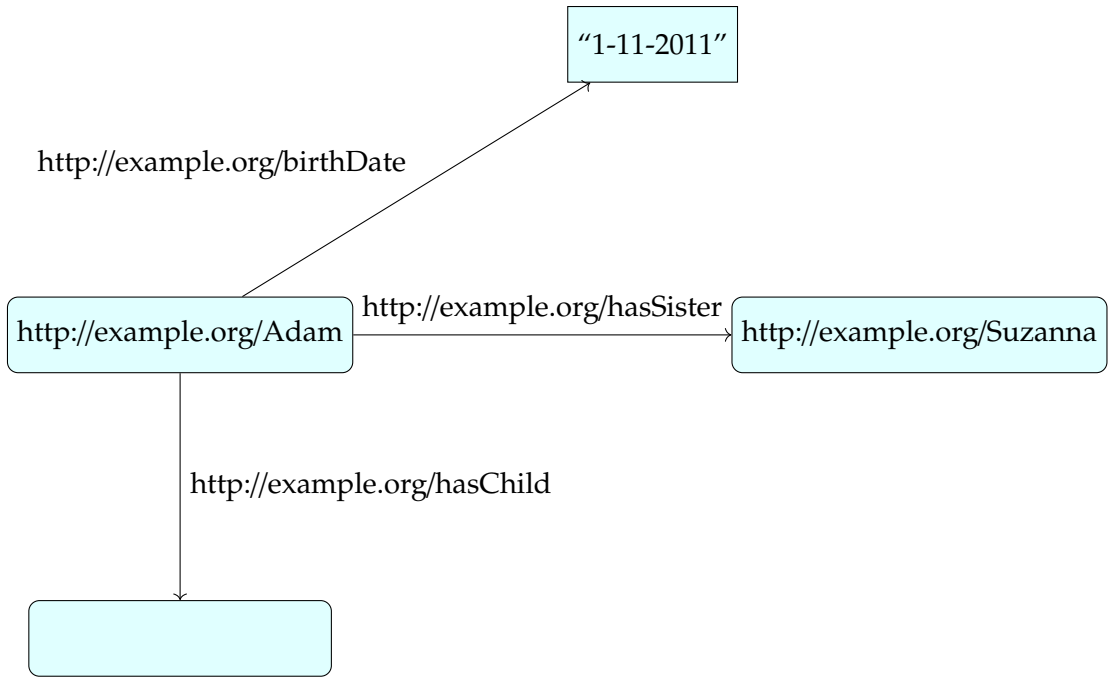

Figure 3.2: RDF graph representing different kinds of components.

Formally, an RDF graph is defined as follows:

Definition 3.1.1. *(RDF Graph) Given RDF graph $\mathcal{G}$ is a finite set of RDF triples. Let $\mathcal{I}$ be a set of IRIs, $\mathcal{B}$ a set of blank nodes, and $\mathcal{L}$ a set of literals. An RDF triple is $< s, p, o > \in (\mathcal{I} \cup \mathcal{B}) \times \mathcal{I} \times (\mathcal{I} \cup \mathcal{B} \cup \mathcal{L})$, where s, p, o represents the subject, predicate and object respectively of the triple.*

RDF defines a *resource* as any real-world entity such as a person, a place or an abstract concept that is uniquely identifiable by IRI. The resource is also called *entity*. A collection of entities which share common characteristics is called *class*. An RDF class describes a group of entities that have the same *rdf:type* such as Scientist, Organisation or Country.

3.1.2 The SPARQL query language

Simple Protocol and RDF Query Language (SPARQL) is a declarative query language to manipulate data represented as RDF triples [Harris, Seaborne, and Prud'hommeaux 2013]. A SPARQL query is transformed into a sub-graph. Solving a SPARQL query consists of finding a sub graph that matches a query graph of a given RDF graph. There are different types of graph patterns i.e., RDF graph with variables. In this thesis, we consider *Basic Graph Pattern* (BGP) queries. Formally, A BGP Query is defined as follows:

Definition 3.1.2. *(BGP Query) A BGP query is a query of the form*
SELECT ?$x1$...?xi *WHERE* $t1 \wedge t2 \wedge .. \wedge tj$
where $t1...tj$ *are triples and* ?$x1$...?xi *are distinguished variables used in* $t1...tj$.

BGP query uses a subset of triple patterns to extract the desired information from the RDF graph. These triple patterns are RDF triples with variable prefixed by "?".

Basically, a structure of SPARQL query consists of five components.

- an optional *prefix definition* to make the query more human-readable.
- a *form specification* that can be one of the following: *Select*, *Construct*, *Ask* and *Describe*. It contains variables i.e., what to search for.
- *dataset specification* to specify the dataset to be queried using *FROM*.
- *WHERE* to specify the graph pattern to be corresponded i.e., the conditions that have to be met.
- an optional *solution modifiers* to perform a specific ordering or sorting of the result set such as *LIMIT*, *FILTER*, *ORDER BY*, and *OFFSET*.

Example 3.1.1. *The following query lists the English names of all known rivers (see Listing 3.1).*

Listing 3.1 – Example of a SPARQL query

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX dbo: <http://dbpedia.org/ontology/>
```

```
SELECT ?name
FROM <http://dbpedia.org/>
WHERE {
    ?s rdf:type dbo:River .
    ?s rdfs:label ?name .
    FILTER (lang(?name) = 'en')
    }
```

3.2 Linked Data quality

Assessing data quality is one of the important challenges that data consumers are facing [Wang and Strong 1996]. It is a multifaceted challenge, for instance, the term "quality" is commonly described as *fitness for use* [Juran, Gryna, and Bingham 1974] which encompasses several dimensions such as accuracy, timeliness, consistency, correctness, completeness, etc. These dimensions are subjective as long as data quality may be suitable or not depending on the purpose of the task at hand. For example, completeness implies that the amount of data is sufficient for the consumer's needs. It can be measured as the percentage of data available divided by the data required, where 100% is the best value. The question, however, is whether we can consider 70% to be of high quality. This amount of information could be sufficient, for example, for the description of a film but not enough for a medical use case. Besides, the quality dimensions are not totally separated and several dimensions can interact to assess the quality of a given use case, for instance choosing the preferred dimensions will affect other dimensions. For example, having an accurate data results in high trustworthiness and validity or having timely data may cause low accuracy, incompleteness and/or inconsistency [Zaveri, Rula, Maurino, Pietrobon, Lehmann, and Auer 2016].

Different studies have defined and categorized data quality dimensions [Wand and Wang 1996; Wang and Strong 1996; Naumann 2003]. In this section, we clarify one of the most used data quality structures [Wand and Wang 1996], it classifies the data quality dimensions into four fundamental groups. This classification depends on the way that data can be measurable:

- **Intrinsic dimensions** are those dimensions that can be measured by metrics to assess

the validity and consistent of the data, such as *accuracy*.

- **Contextual dimensions** are those dimensions where data quality must be considered within the context of the given task, such as *timeliness*.
- **Representational dimensions** include metrics related to the design of data; in other words, how well the data is represented, such as *interpretability*.
- **Accessibility dimensions** are about how easily accessible and secure data is, such as *availability* and *security*.

Later, a number of data quality dimensions appeared which are related to Linked Data and categorized in the data quality classifications[Bizer and Cyganiak 2009; Zaveri, Rula, Maurino, Pietrobon, Lehmann, and Auer 2016].

In the context of Linked Data, different comprehensive surveys, which focus on data quality methodologies for structured and Linked Data, already exist in the literature [Batini, Cappiello, Francalanci, and Maurino 2009; Zaveri, Rula, Maurino, Pietrobon, Lehmann, and Auer 2016; Paulheim 2017]. Zaveri et al. [Zaveri, Rula, Maurino, Pietrobon, Lehmann, and Auer 2016] conducted a comprehensive Systematic Literature Review (SLR) and identified 18 different quality dimensions that can be applied to assess the quality of Linked Data. They classified these dimensions according to the four groups identified by [Wand and Wang 1996]. Figure 3.3 illustrates the 18 proposed dimensions by [Zaveri, Rula, Maurino, Pietrobon, Lehmann, and Auer 2016] with the relationships between them.

3.3 State-of-the-art

In this thesis, we focus on two dimensions, completeness and conciseness, for several reasons. In the first place, these dimensions affect other dimensions of data quality such as accuracy, timeliness and consistency as shown in Figure 3.3. Second, both dimensions enhance answer quality of queries. Thus, the proposed methods are useful to provide answers to user's queries in a rightness and precise way. Third, completeness and conciseness dimensions have different attention in Linked Data researches. Whereas completeness is one of the most essential dimension in data quality dimensions [Margaritopoulos,

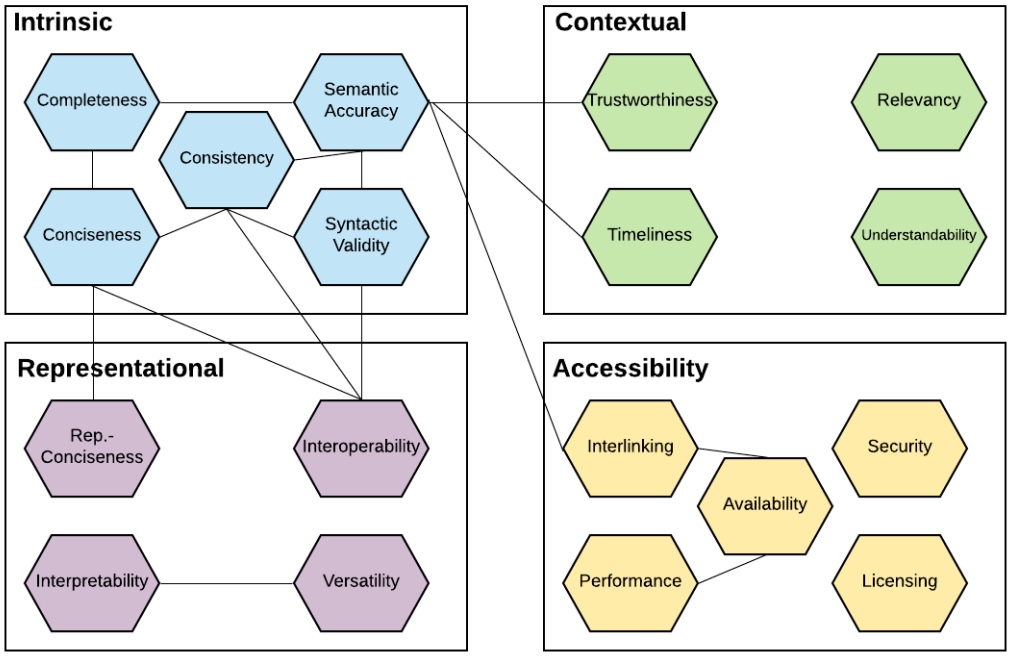

Figure 3.3: Linked Data quality dimensions and the relationships between them.

Margaritopoulos, Mavridis, and Manitsaris 2012a], conciseness does not get fair attention, and this will be illustrated in Section 3.3.2.

In the following subsections, we present two different reviews to survey completeness and conciseness dimensions. We have done a Systematic Literature Review (SLR) for the completeness dimension and a traditional literature review for the conciseness dimension. As we explained in the introduction of this chapter, regarding the limited number of works dealing with Linked Data conciseness, we felt that there is no need to perform an SLR for this dimension.

3.3.1 Linked Data completeness: a systematic literature review

Different surveys which focus on Linked Data quality have been proposed [Batini, Cappiello, Francalanci, and Maurino 2009; Zaveri, Rula, Maurino, Pietrobon, Lehmann, and Auer 2016; Paulheim 2017]. In this work, we present a Systematic Literature Review (SLR) on one of the core dimensions of Linked Data quality i.e., completeness

[Margaritopoulos, Margaritopoulos, Mavridis, and Manitsaris 2012b]. Completeness is a data quality measure that refers to the amount of information present in a particular dataset [Zaveri, Rula, Maurino, Pietrobon, Lehmann, and Auer 2016]. For example, the instance *Albert_Einstein* might suffer from a data completeness problem when his birth place is missing in the dataset. In this part, we have conducted a comprehensive SLR focus on completeness of Linked Data, which includes different types of completeness. Our objective is to qualitatively and quantitatively analyze the existing articles that propose methods to assess or improve several types of completeness dimensions. We also classify the existing measures into various types of completeness.

In terms of Linked Data, existing literature [Zaveri, Rula, Maurino, Pietrobon, Lehmann, and Auer 2016] identified four types of completeness to measure the degree of completeness of data sources. Pipino et al. [Pipino, Lee, and Wang 2002] divided completeness into: (i) schema completeness that is the degree to which classes and properties are presented in a schema, (ii) property completeness which is the extent of the missing property values of a specific kind of property, and (iii) population completeness that refers to the ratio of number of represented objects to total number of real-world objects. Later, a new type of completeness is introduced for Linked Data called (iv) interlinking completeness, which defines links losses between datasets via their linksets [Albertoni and Pérez 2013]. Thus, we classified the selected articles into one of these types of completeness as illustrated in Section 3.3.1.2. As a result of our SLR, we identified three new types of completeness, namely, (v) currency, (vi) metadata and (vii) labelling completeness making it a total of seven types of completeness. It should be noted that we do not assume an Open World Assumption [Drummond and Shearer 2006] since it contradicts completeness by definition, as a gold standard or a reference dataset is needed to be compared against the given dataset.

3.3.1.1 Systematic literature review methodology

In this section, we explain our SLR methodology to identify all articles related to Linked Data completeness and we summarize the proposed solutions in terms of (i) the problem addressed, (ii) approaches and metrics proposed and (iii) tools developed to assess the

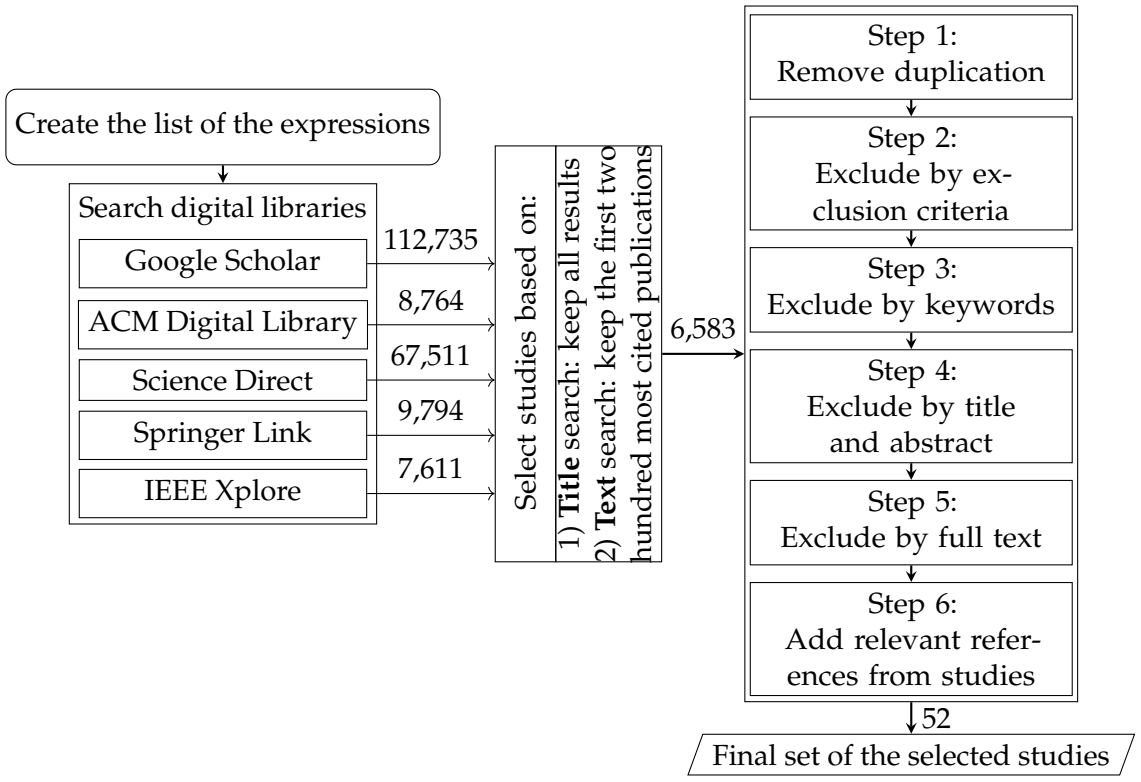

Figure 3.4: Overview of the systematic literature review methodology.

issue of completeness.

We conducted this systematic review by following the systematic review procedures described in [Kitchenham and Charters 2007]. According to [Kitchenham and Charters 2007], a systematic review is useful for several reasons, such as: (i) summarize and compare the various methodologies in a domain, (ii) identify open problems, (iii) contribute a hybrid concept comprising of various methodologies developed in a domain, or/and (iv) synthesize new ideas to address open problems. This systematic review tackles, in particular, problems (i), (ii) and (iii). It summarizes and compares various LD completeness data quality assessment methodologies as well as identifying open problems related to LD completeness. An overview of our search methodology including the number of retrieved articles at each step is shown in Figure 3.4 and described in detail below.

Research questions. In this SLR, we aim to answer the following general research question:

How can we assess the completeness of Linked Data, which includes different types of completeness considering several approaches?

We divide this general research question into sub-questions:

- what *types* of completeness currently exist in Linked Data?
- what are the proposed *approaches and metrics* to identify and measure the completeness of Linked Data?
- what are the data completeness *problems* being discussed by researchers?
- what *tools* are available to detect completeness of Linked Data?

Inclusion criteria

- articles published in English
- articles published between 2006[15]-2019
- articles that:
  - studied or measured completeness of Linked Data
  - proposed data completeness methodology or framework
  - proposed and applied metrics for completeness of Linked Data

Exclusion criteria

- articles that have not been peer-reviewed
- articles published in other languages
- master or doctorate thesis, poster, PowerPoint presentation or books
- articles that focused neither on Linked Data nor on Semantic Web technologies

[15] As the term of Linked Data first appeared in 2006 [Berners-Lee 2006a]

Generating a search strategy. Search strategies in a systematic review are usually iterative and are run separately to avoid bias and to maximize coverage coverage of all related articles. We performed a search on the following search engines: Google Scholar, IEEE Xplore, ACM Digital Library, Science Direct and Springer Link. Because it is impractical to accept all the returned results in the research engines when we search for the keywords in the full articles, we limit our research to the most cited 200 articles from each source.

From our perspective, searching only on the title is not efficient and does not always provide all the relevant articles. This is because authors are often inclined to use agile titles which do not express the real content of the article. Thus, we divide our search strategy into three steps:

- scan article titles based on inclusion/exclusion criteria
- search within text and determine fit based on inclusion/exclusion criteria

Figure 3.4 provides more details on the exact numbers of articles searched and obtained over the aforementioned steps.

One of the most important parts is defining the search terms. These expressions that collect the articles related to Linked Data completeness should be based on a defined search strategy. This strategy aims to find as many relevant articles as possible. Based on our discussions and testing to obtain the most related articles as possible in our domain, the search string that was proposed contains synonyms of the concept term. As our concept term is “Linked Data completeness”, we added the alternative spellings, synonyms, as well as terms related to quality. Finally, we connected them using the boolean operators **OR** and **AND**. The expressions that were used to extract the interested studies are:

- Exp. 1: ("Linked Data") **AND** (quality **OR** assessment **OR** evaluation **OR** methodology **OR** measuring **OR** completeness)
- Exp. 2: ("Linked Open Data") **AND** (quality **OR** assessment **OR** evaluation **OR** methodology **OR** measuring **OR** completeness)
- Exp. 3: LOD **AND** (quality **OR** assessment **OR** evaluation **OR** methodology **OR** measuring **OR** completeness)

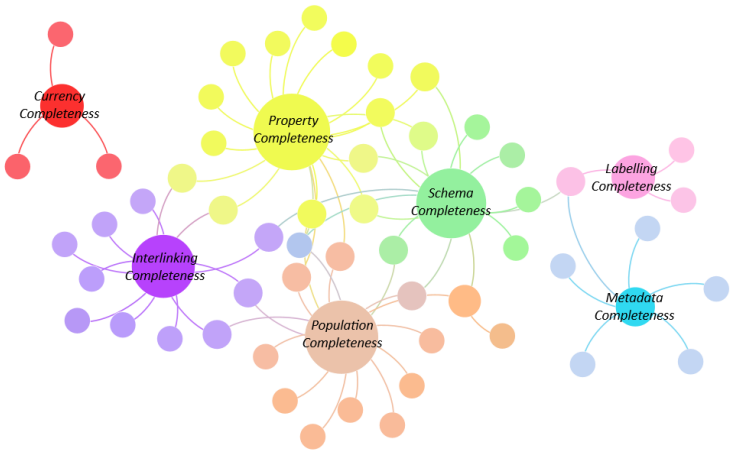

Figure 3.5: Classification of the 52 core articles by type of completeness.

After removing of duplicated papers, we excluded the papers based on exclusion criteria. After that, we excluded them based on the abstract of each paper then the full text of the paper in case that the abstract was not sufficiently clear to take a decision. Finally, we added the relevant papers from the references of the selected papers from the last step.

3.3.1.2 Linked Data completeness analysis

Quantitative analysis. As a result of our methodology, we identified 52 core articles that focus on Linked Data completeness. The final list of the selected articles is shown in Table 3.2. Table 3.3 shows the list with the type of completeness for each article addressed. We further categorized them with respect to the type of completeness, as shown in Figure 3.5. The unlabeled nodes represent publications and the edges link to the type of completeness that the publication covers. We can observe that a publication can address multiple types of completeness. Property completeness is the most addressed by the studies, with 18 publications. From the 52 core articles, 28 were published from 2016 till now 2019; hence about 52% of the studies are quite recent and the trend shows that more researchers are getting involved in this domain as the years go by as illustrated in Figure 3.6. Figure 3.7 shows where researchers are publishing their work, where the top journal is Journal of Web Semantics and the top conference for publishing being European Semantic Web Conference. We observe that researchers are now publishing more in conferences with 29

Table 3.1: Number of the articles retrieved in each search engine.

		GS	SD	ACM	SL	IEEE
Title	Exp. 1	173	7	557	3	221
	Exp. 2	35	1	103	-	77
	Exp. 3	27	1	41	1	191
Anywhere	Exp. 1	31,100	5,487	6,903	5,365	2,686
	Exp. 2	14,600	573	928	2,270	937
	Exp. 3	66,800	61,442	232	2,155	3,499

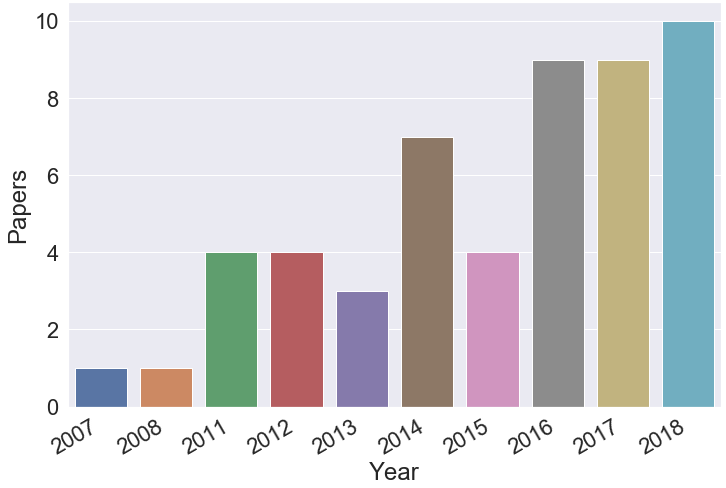

Figure 3.6: Number of core articles by year.

articles (54%) published at a conference even though no one conference seems to be over represented. 13 articles (24%) were published in journals and seven articles (13%) were published in workshop proceedings.

Qualitative analysis In this section, we analyze the 52 articles qualitatively to extract relevant information regarding Linked Data completeness. After analyzing the selected articles in detail, we identified and extracted 23 ubiquitous metrics which are presented in Table 3.4 that can be applied to assess the completeness of Linked Data, categorizing them based on the type of completeness covered. As mentioned previously, (i) schema, (ii) property, (iii) population and (iv) interlinking completeness were already identified by [Zaveri, Rula, Maurino, Pietrobon, Lehmann, and Auer 2016]. As part of our SLR, we identified three more types, namely, (v) currency, (vi) metadata and (vii) labelling

Table 3.2: List of the 52 core articles related to Linked Data completeness.

Article	Citation
A comprehensive quality model for Linked Data	[Radulovic, Mihindukulasooriya, García-Castro, and Gómez-Pérez 2018]
A framework for Linked Data fusion and quality assessment	[Nahari, Ghadiri, Jafarifard, Dastjerdi, and Sack 2017]
A linkset quality metric measuring multilingual gain in SKOS Thesauri	[Albertoni, De Martino, and Podestà 2015]
A Measure-Theoretic Foundation for Data Quality	[Bronselaer, De Mol, and De Tre 2018]
A metric-driven approach for interlinking assessment of RDF graphs	[Yaghouti, Kahani, and Behkamal 2015]
A metrics-driven approach for quality assessment of Linked Open Data	[Behkamal, Kahani, Bagheri, and Jeremic 2014]
A Model for Linked Open Data Acquisition and SPARQL Query Generation	[Alec, Reynaud-Delaître, and Safar 2016]
A Quality Model for Linked Data Exploration	[Cappiello, Di Noia, Marcu, and Matera 2016]
A Two-Fold Quality Assurance Approach for Dynamic Knowledge Bases: The 3cixty Use Case	[Mihindukulasooriya, Rizzo, Troncy, Corcho, and García-Castro 2016]
Analyzing Linked Data Quality with LiQuate	[Ruckhaus, Vidal, Castillo, Burguillos, and Baldizan 2014]
Are Linked Datasets fit for Open-domain Question Answering? A Quality Assessment	[Thakkar, Endris, Gimenez Garica, Debattista, Lange, and Auer 2016]
Assessing and Improving Domain Knowledge Representation in DBpedia	[Font, Zouaq, and Gagnon 2017]
Assessing Linked Data Mappings Using Network Measures	[Guéret, Groth, Stadler, and Lehmann 2012]
Assessing linkset quality for complementing third-party datasets	[Albertoni and Pérez 2013]
Assessing the Completeness Evolution of DBpedia: A Case Study	[Issa, Paris, and Hamdi 2017b]
Automated quality assessment of metadata across open data portals	[Neumaier, Umbrich, and Polleres 2016]
Automatically Generating Data Linkages Using a Domain-Independent Candidate Selection Approach	[Song and Heflin 2011]
BOUNCER: Privacy-aware Query Processing Over Federations of RDF Datasets	[Endris, Almhithawi, Lytra, Vidal, and Auer 2018]
Capturing the age of Linked Open Data: Towards a dataset-independent framework	[Rula, Palmonari, and Maurino 2012]
Co-evolution of RDF Datasets	[Faisal, Endris, Shekarpour, Auer, and Vidal 2016]
Comparing Index Structures for Completeness Reasoning	[Darari, Nutt, and Razniewski 2018]
Comparison of metadata quality in open data portals using the Analytic Hierarchy Process	[Kubler, Robert, Neumaier, Umbrich, and Le Traon 2018]
CROCUS: Cluster-based Ontology Data Cleansing	[Cherix, Usbeck, Both, and Lehmann 2014]
Data Quality Assessment in Europeana: Metrics for Multilinguality	[Charles, Stiller, Kiraly, and Bailer 2018]
Dataset Profiling - a Guide to Features, Methods, Applications and Vocabularies	[Ellefi, Bellahsene, Breslin, Demidova, Dietze, Szymanski, and Todorov 2016]
Enabling Fine-Grained RDF Data Completeness Assessment	[Darari, Razniewski, Prasojo, and Nutt 2016]
Enhancing answer completeness of SPARQL queries via crowdsourcing	[Acosta, Simperl, Flöck, and Vidal 2017]
Enhancing Dbpedia Quality Using Markov Logic Networks	[Ali and Alchaita 2018]
Ensuring the Completeness and Soundness of SPARQL Queries Using Completeness Statements about RDF Data Sources	[Darari, Nutt, Razniewski, and Rudolph 2017]
How Linked Data can aid machine learning-based tasks	[Mountantonakis and Tzitzikas 2017]
Improving Curated Web-Data Quality with Structured Harvesting and Assessment	[Feeney, O'Sullivan, Tai, and Brennan 2014]
Improving the Quality of Linked Data Using Statistical Distributions	[Paulheim and Bizer 2014]
Interlinking Linked Data Sources Using a Domain-Independent System	[Nguyen, Ichise, and Le 2013]
KBQ - A Tool for Knowledge Base Quality Assessment Using Evolution Analysis	[Rizzo, Torchiano, and Torino 2017]
Labels in the Web of Data	[Ell, Vrandečić, and Simperl 2011]
Linked Data Aggregation Algorithm: Increasing Completeness and Consistency of Data	[Knap and Michelfeit 2012]
Linked data quality of DBpedia, Freebase, OpenCyc, Wikidata, and YAGO	[Färber, Bartscherer, Menne, and Rettinger 2018]
Methodology for Linked Enterprise Data Quality Assessment Through Information Visualizations	[Gürdür, El-khoury, and Nyberg 2018]
Metrics-driven framework for LOD quality assessment	[Behkamal 2014]
Quality and Complexity Measures for Data Linkage and Deduplication	[Christen and Goiser 2007]
Recoin: Relative Completeness in Wikidata	[Balaraman, Razniewski, and Nutt 2018]
Sieve: Linked Data Quality Assessment and Fusion	[Mendes, Mühleisen, and Bizer 2012a]
SWIQA - a Semantic Web Information Quality Assessment Framework	[Fürber and Hepp 2011a]
Test-driven evaluation of Linked Data quality	[Kontokostas, Westphal, Auer, Hellmann, Lehmann, Cornelissen, and Zaveri 2014]
Towards a Data Quality Framework for Heterogeneous Data	[Micic, Neagu, Campean, and Zadeh 2017]
Towards a vocabulary for data quality management in semantic web architectures	[Fürber and Hepp 2011b]
Towards an objective assessment framework for Linked Data quality	[Assaf, Senart, and Troncy 2016]
Towards Ontology Quality Assessment	[McGurk, Abela, and Debattista 2017]
Towards the Russian Linked Culture Cloud: Data Enrichment and Publishing	[Mouromtsev, Haase, Cherny, Pavlov, Andreev, and Spiridonova 2015]
Towards unified and native enrichment in event processing systems	[Hasan, O'Riain, and Curry 2013]
URI Disambiguation in the Conext of Linked Data	[Jaffri, Glaser, and Millard 2008]
What's up LOD Cloud? Observing The State of Linked Open Data Cloud Metadata	[Assaf, Troncy, and Senart 2015]

Table 3.3: List of the 52 core articles classified according to the seven types.

Article/Type of completeness	Schema	Property	Population	Interlinking	Currency	Metadata	Labelling
Radulovic et al. [Radulovic, Mihindukulasooriya, García-Castro, and Gómez-Pérez 2018]		✓		✓			
Nahari et al. [Nahari, Ghadiri, Jafarifard, Dastjerdi, and Sack 2017]		✓		✓			
Albertoni et al. [Albertoni, De Martino, and Podestà 2015]				✓			
Bronselaer et al. [Bronselaer, De Mol, and De Tre 2018]	✓						
Yaghouti et al. [Yaghouti, Kahani, and Behkamal 2015]				✓			
Behkamal et al. [Behkamal, Kahani, Bagheri, and Jeremic 2014]	✓	✓					
Alec et al. [Alec, Reynaud-Delaître, and Safar 2016]		✓					
Cappiello et al. [Cappiello, Di Noia, Marcu, and Matera 2016]	✓						
Mihindukulasooriya et al. [Mihindukulasooriya, Rizzo, Troncy, Corcho, and García-Castro 2016]		✓	✓				
Ruckhaus et al. [Ruckhaus, Vidal, Castillo, Burguillos, and Baldizan 2014]			✓	✓			
Thakkar et al.[Thakkar, Endris, Gimenez Garica, Debattista, Lange, and Auer 2016]				✓			
Font et al. [Font, Zouaq, and Gagnon 2017]		✓					
Guéret et al. [Guéret, Groth, Stadler, and Lehmann 2012]				✓			
Albertoni et al. [Albertoni and Pérez 2013]				✓			
Issa et al. [Issa, Paris, and Hamdi 2017b]	✓						
Neumaier et al. [Neumaier, Umbrich, and Polleres 2016]						✓	
Song and Heflin [Song and Heflin 2011]			✓				
Endris et al. [Endris, Almhithawi, Lytra, Vidal, and Auer 2018]		✓					
Rula et al. [Rula, Palmonari, and Maurino 2012]					✓		
Faisal et al. [Faisal, Endris, Shekarpour, Auer, and Vidal 2016]					✓		
Darari et al. [Darari, Nutt, and Razniewski 2018]			✓				
Kubler et al. [Kubler, Robert, Neumaier, Umbrich, and Le Traon 2018]						✓	
Cherix et al. [Cherix, Usbeck, Both, and Lehmann 2014]			✓				
Charles et al. [Charles, Stiller, Kiraly, and Bailer 2018]						✓	
Ellefi et al. [Ellefi, Bellahsene, Breslin, Demidova, Dietze, Szymanski, and Todorov 2016]						✓	
Darari et al. [Darari, Razniewski, Prasojo, and Nutt 2016]		✓	✓				
Acosta et al. [Acosta, Simperl, Flöck, and Vidal 2017]			✓				
Ali and Alchaita [Ali and Alchaita 2018]		✓					
Darari et al. [Darari, Nutt, Razniewski, and Rudolph 2017]			✓				
Mountantonakis and Tzitzikas [Mountantonakis and Tzitzikas 2017]		✓					
Feeney et al. [Feeney, O'Sullivan, Tai, and Brennan 2014]		✓	✓				
Paulheim and Bizer [Paulheim and Bizer 2014]		✓					
Nguyen et al. [Nguyen, Ichise, and Le 2013]				✓			
Rizzo et al. [Rizzo, Torchiano, and Torino 2017]					✓		
Ell et al. [Ell, Vrandečić, and Simperl 2011]							✓
Knap and Michelfeit [Knap and Michelfeit 2012]	✓		✓				
Färber et al. [Färber, Bartscherer, Menne, and Rettinger 2018]	✓	✓	✓				
Gürdür et al. [Gürdür, El-khoury, and Nyberg 2018]							✓
Behkamal [Behkamal 2014]	✓	✓					
Christen and Goiser [Christen and Goiser 2007]		✓					
Balaraman et al. [Balaraman, Razniewski, and Nutt 2018]	✓						
Mendes et al. [Mendes, Mühleisen, and Bizer 2012a]	✓	✓					
Fürber and Hepp [Fürber and Hepp 2011a]	✓	✓	✓				
Kontokostas et al. [Kontokostas, Westphal, Auer, Hellmann, Lehmann, Cornelissen, and Zaveri 2014]	✓	✓					
Micic et al. [Micic, Neagu, Campean, and Zadeh 2017]		✓	✓				
Fürber and Hepp [Fürber and Hepp 2011b]		✓					
Assaf et al. [Assaf, Senart, and Troncy 2016]	✓					✓	✓
McGurk et al. [McGurk, Abela, and Debattista 2017]	✓						
Mouromtsev et al. [Mouromtsev, Haase, Cherny, Pavlov, Andreev, and Spiridonova 2015]			✓	✓			
Hasan et al. [Hasan, O'Riain, and Curry 2013]		✓					
Jaffri et al. [Jaffri, Glaser, and Millard 2008]				✓			
Assaf et al. [Assaf, Troncy, and Senart 2015]						✓	

completeness.

In the following, we describe each of the seven types by providing a definition and discussing the problems and approaches that they address. We summarize the problems and approaches found for each type of completeness and provide a few examples. The full list of approaches is reported in Table 3.4.

(i) Schema completeness

The schema of a dataset is considered complete, if it contains all the classes and properties needed for a given task. It is also called ontology completeness [Fürber and Hepp 2011a]. Fürber et al. [Fürber and Hepp 2011a] defined schema completeness as the degree to which classes and properties are represented in a schema. In a similar sense but under a different name, Mendes et al. [Mendes, Mühleisen, and Bizer 2012a] defined *intensional completeness* which is the existence of all the attributes in a dataset for a given task. For example, the dataset suffers from a schema completeness problem when the property *capital* is missed from the instance *France* .

Definition 3.3.1 (Schema completeness)**.** *Schema completeness is the degree to which the classes and properties of an ontology are represented in a LD dataset.*

Problems. Several articles address the challenge of development of new tools and frameworks to assess and improve completeness and other data quality dimensions [Behkamal 2014; Kontokostas, Westphal, Auer, Hellmann, Lehmann, Cornelissen, and Zaveri 2014; Assaf, Senart, and Troncy 2016]. The authors in [Bronselaer, De Mol, and De Tre 2018] were interested in the application of first order logic predicates and developed a capacity function (i.e., a fuzzy measure) to express completeness. [Balaraman, Razniewski, and Nutt 2018] investigated how to employ the similarity between entities in a dataset to determine completeness. In [Issa, Paris, and Hamdi 2017b], the authors built transaction vectors constituted of sequence of properties that deduced from instances to use them as an input to generate frequent patterns in order to compute the completeness.

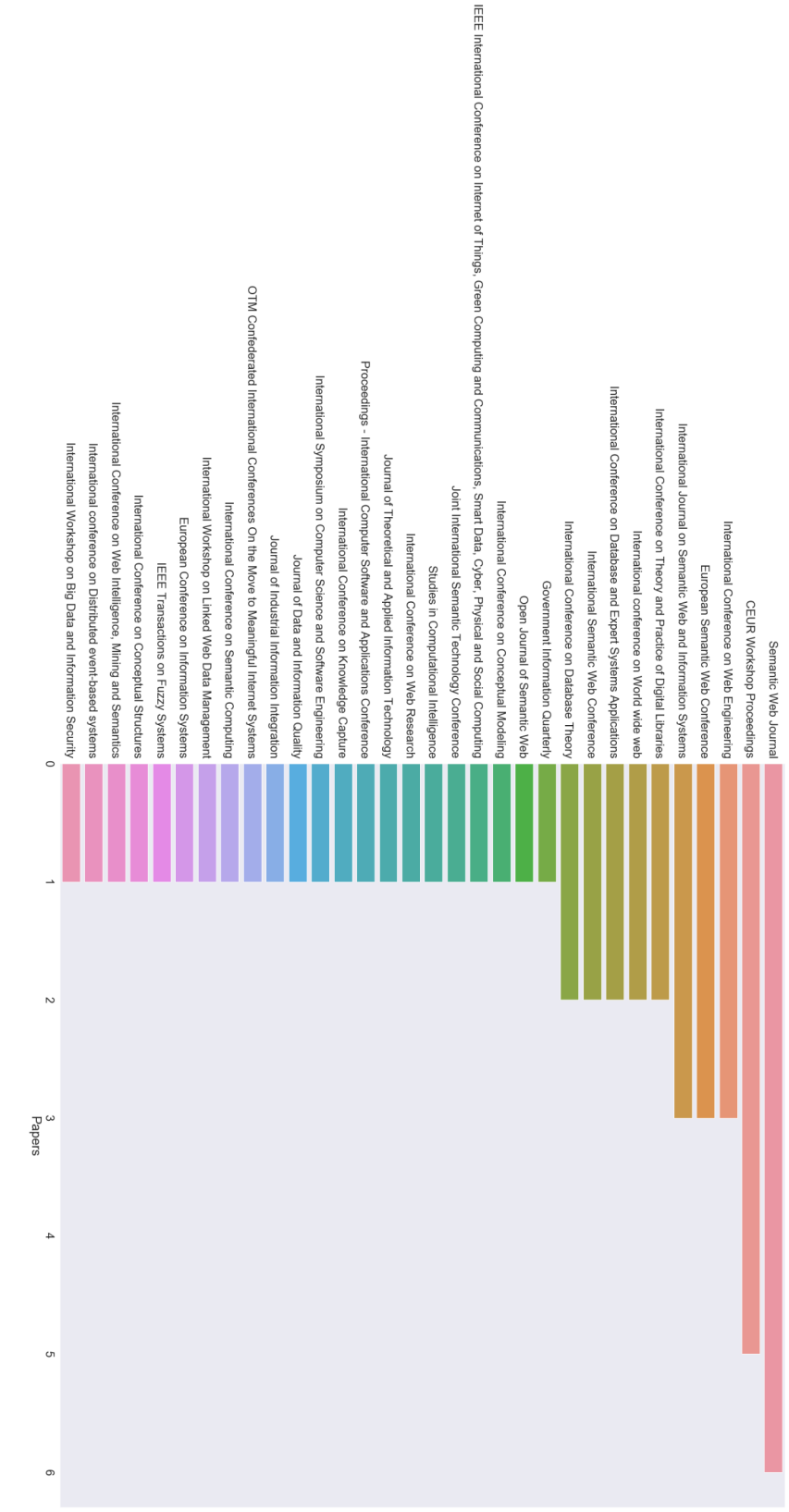

Figure 3.7: Classification of articles by conferences and journals.

Approaches and metrics. There are 13 articles that propose some approaches of metrics about schema completeness. These existing approaches defined a set of metrics to assess schema completeness [Behkamal, Kahani, Bagheri, and Jeremic 2014; Mendes, Mühleisen, and Bizer 2012a; Kontokostas, Westphal, Auer, Hellmann, Lehmann, Cornelissen, and Zaveri 2014; Assaf, Senart, and Troncy 2016; McGurk, Abela, and Debattista 2017] such as applying fusion methods or defining quality indicators, or assessing completeness based on extracting a set of frequent/required predicates [Bronselaer, De Mol, and De Tre 2018; Cappiello, Di Noia, Marcu, and Matera 2016; Issa, Paris, and Hamdi 2017b; Knap and Michelfeit 2012; Färber, Bartscherer, Menne, and Rettinger 2018; Balaraman, Razniewski, and Nutt 2018; Issa, Paris, Hamdi, and Cherfi 2019]. Several metrics measure completeness as the ratio of the number of classes/properties presented in a dataset to the total number of classes/properties [Behkamal, Kahani, Bagheri, and Jeremic 2014; Knap and Michelfeit 2012; Färber, Bartscherer, Menne, and Rettinger 2018; Mendes, Mühleisen, and Bizer 2012a; Kontokostas, Westphal, Auer, Hellmann, Lehmann, Cornelissen, and Zaveri 2014; Assaf, Senart, and Troncy 2016; McGurk, Abela, and Debattista 2017]. Other metrics take into account only the mandatory properties to assess the completeness [Cappiello, Di Noia, Marcu, and Matera 2016; Issa, Paris, and Hamdi 2017b; Issa, Paris, Hamdi, and Cherfi 2019]. [Balaraman, Razniewski, and Nutt 2018] measured ratio of similar instances/subjects missing same properties.

(ii) Property completeness

Property completeness as defined by [Zaveri, Rula, Maurino, Pietrobon, Lehmann, and Auer 2016] is the measure of the missing values for a specific property. This is similar to the definition of [Pipino, Lee, and Wang 2002] which referred to it as column completeness. Property completeness is measured by determining if a specific property has missing values. For example, the dataset suffers from a property completeness problem when the property *capital* of the instance *France* does not have a value, namely, *Paris*.

Definition 3.3.2 (Property completeness)**.** *Property completeness is the degree to which values for a specific property are available for a given task.*

Problems. The common research challenges that addresses property completeness is the development of data quality models, metrics and tools upon which benchmarking and evaluation may be carried out. [Radulovic, Mihindukulasooriya, García-Castro, and Gómez-Pérez 2018; Nahari, Ghadiri, Jafarifard, Dastjerdi, and Sack 2017; Behkamal, Kahani, Bagheri, and Jeremic 2014; Ali and Alchaita 2018; Mendes, Mühleisen, and Bizer 2012a; Behkamal 2014; Feeney, O'Sullivan, Tai, and Brennan 2014; Kontokostas, Westphal, Auer, Hellmann, Lehmann, Cornelissen, and Zaveri 2014] all proposed novel models, methodology and/or metrics for measuring property completeness as part of general data quality assessment. [Endris, Almhithawi, Lytra, Vidal, and Auer 2018] explored evaluation of query answer completeness in a privacy preserving manner and [Cappiello, Di Noia, Marcu, and Matera 2016] developed models for evaluating completeness employing automatic query generation.

Approaches and metrics. There are 21 articles that addressed property completeness either by proposing an approach or a metric to measure completeness. A prominent methodology for assessing property completeness focuses on the development of novel frameworks towards measuring the level of completeness of a knowledge base such as [Behkamal, Kahani, Bagheri, and Jeremic 2014] where the authors use the Goal Question Metric (GQM) method to define metrics for inherent qualities of a dataset. A set of metrics based on measurement-theory have been proposed for evaluating the inherent quality characteristics of a dataset where property completeness is evaluated the ratio of the sum of the number of presented properties per instance to the total number of instances in the dataset. Furthermore, [Mendes, Mühleisen, and Bizer 2012a] proposed a framework for flexibly expressing quality assessment methods as well as data fusion methods where property completeness was explored using the proportion of unique non-missing objects in the dataset. Other approaches include the application of aggregate functions [Darari, Razniewski, Prasojo, and Nutt 2016; Endris, Almhithawi, Lytra, Vidal, and Auer 2018] and statistical distributions [Paulheim and Bizer 2014; Ali and Alchaita 2018; Nahari,

Ghadiri, Jafarifard, Dastjerdi, and Sack 2017].

(iii) Population completeness

A dataset is complete if it contains all of real-world objects for a given task, which is also called the completeness at data (instance) level [Fürber and Hepp 2011a]. Population completeness is also termed *extensional completeness* [Mendes, Mühleisen, and Bizer 2012a]. For example, the dataset suffers from a population completeness problem if it does not have all the French cities.

Definition 3.3.3 (Population completeness)**.** *Population completeness is the degree to which all real-world objects of a particular type are represented in LD dataset.*

Problems. The popular challenge in this type is to check a KB to see whether it contains all entities of a given type. [Mihindukulasooriya, Rizzo, Troncy, Corcho, and García-Castro 2016] proposed an approach to maintain the quality of KB that evolves repeatedly. The proposed approach provides an overview of the change of a given KB and a fine-grained analysis. Moreover, the authors in [Mouromtsev, Haase, Cherny, Pavlov, Andreev, and Spiridonova 2015] used completeness metrics to assess the quality of newly published Linked Data for cultural heritage. [Song and Heflin 2011] was focused on disambiguation problem, the authors provided a method to scalably resolve entity co-reference in structured datasets. Moreover, efforts have been made to include completeness information in KBs. This is achieved by adding true facts such as *Adele has one brother*. This information is essential to evaluate query completeness and soundness [Darari, Razniewski, Prasojo, and Nutt 2016; Darari, Nutt, Razniewski, and Rudolph 2017; Darari, Nutt, and Razniewski 2018].

Approaches and metrics. A set of 14 articles have been proposed to enhance population completeness. Various metrics that check KBs to see whether they contain all entities of a given type in comparison to real-world data [Mihindukulasooriya, Rizzo, Troncy, Corcho, and García-Castro 2016] or deal with query completeness via hybrid computation [Acosta, Simperl, Flöck, and Vidal 2017], include completeness

information as part of the KB that can be used for validation [Darari, Razniewski, Prasojo, and Nutt 2016]. On the other hand, [Ruckhaus, Vidal, Castillo, Burguillos, and Baldizan 2014] used a Bayesian Network to model the dependencies among resources that belong to a set of linked datasets and represent the joint probability distributions of relationships among resources. The probability of an individual resource is considered the likelihood of redundancy or indicator of completeness regarding the resource.

(iv) Interlinking completeness This type particularly focuses on data integration which is a core tenet of Linked Data. It refers to the instances that are interlinked in the dataset for disambiguation with regards to a reference dataset [Guéret, Groth, Stadler, and Lehmann 2012]. For example, the instance *France* linked from French national dataset to another instance *French Republic* in the United Nations dataset.

Definition 3.3.4 (Interlinking completeness)**.** *Interlinking completeness is the degree to which instances are interlinked in a LD dataset.*

Problems. A standard data quality model for quality specification and assessment is presented in [Radulovic, Mihindukulasooriya, García-Castro, and Gómez-Pérez 2018]. In the first step, the authors defined base measures for quality evaluation, then combining different base measures to get derived measures and metrics that are obtained by integrating base and/or derived measures. These measures and metrics are used to assess data quality covering various quality dimensions. Albertoni et al. [Albertoni, De Martino, and Podestà 2015] explored how to assess the value of interlinks of datasets in terms of information gain via what they refer to as linkset importing. [Song and Heflin 2011; Jaffri, Glaser, and Millard 2008] proposed methods to resolve entity co-reference and completing links in KBs.

Approaches and metrics. We identified 11 articles that focus on the interlinking completeness in LD. Several approaches have been proposed to assess interlinking completeness [Radulovic, Mihindukulasooriya, García-Castro, and Gómez-Pérez 2018; Nahari, Ghadiri, Jafarifard, Dastjerdi, and Sack 2017; Yaghouti, Kahani, and

Behkamal 2015]. [Ruckhaus, Vidal, Castillo, Burguillos, and Baldizan 2014] analyzed the quality of data and links in LOD cloud using Bayesian Networks. Additionally, [Albertoni, De Martino, and Podestà 2015] estimated the completeness of a dataset by complementing SKOS thesauri with their *skos:exactMatch* related information. In [Nguyen, Ichise, and Le 2013], the authors gathered the essential predicates of data sources using their covering and discriminative abilities. Then, they selected the most suitable alignments based on their confidences and finally, comparing the instances based on the selected alignments.

(v) Currency completeness

Currency according to [Rula, Palmonari, and Maurino 2012] is the degree to which data is up-to-date; and in this work, the authors were interested in providing a model for assessing currency and as a result they developed a metric for currency completeness to evaluate the completeness of the currency measurement. Currency completeness is evaluated on the dataset as it is modified and updated over time. For instance, the population in *France* as it varies over the years.

Definition 3.3.5 (Currency completeness)**.** *Currency completeness is the degree to which elements of a knowledge base are available as it is updated over time.*

Problems. The development of frameworks and metrics for assessing the currency of RDF data is the focus of research such as [Rula, Palmonari, and Maurino 2012; Rula, Panziera, Palmonari, and Maurino 2014] and currency completeness is the by-product of evaluating the proposed framework for assessing currency. [Faisal, Endris, Shekarpour, Auer, and Vidal 2016] also dealt with currency completeness while investigating an approach to deal with the mutual propagation of the changes between a replica and its origin dataset termed as co-evolution.

Approaches and metrics. We found two articles that proposed metrics to measure currency completeness and another article [Rula, Palmonari, and Maurino 2012] proposed a new framework focusing on LD currency. While timeliness captures the freshness of a specific statement or entity [Ellefi, Bellahsene, Breslin, Demidova,

Dietze, Szymanski, and Todorov 2016], in other words, determines the extent to which data are sufficiently up-to-date for a task; currency completeness measures the completeness of the knowledge base as it is being updated over different versions. As such, currency completeness is the intersection between timeliness and completeness where the degree of completeness is measured as the data becomes more up-to-date. [Rula, Palmonari, and Maurino 2012; Rula, Panziera, Palmonari, and Maurino 2014] defined its currency completeness metric as the number of resources for which currency can be computed over the total number of resources occurring in a knowledge base. Furthermore, [Faisal, Endris, Shekarpour, Auer, and Vidal 2016] evaluated currency completeness as the ratio of the number of unique triples in the synchronized dataset to the count of unique triples in the two different versions of the dataset.

(vi) Metadata completeness

Descriptive metadata about datasets enables dataset discovery, and as such [Ellefi, Bellahsene, Breslin, Demidova, Dietze, Szymanski, and Todorov 2016] provided a comprehensive overview on metadata termed as *dataset profiling* where they also assess metadata completeness. Accordingly, metadata is considered complete if it contains all the fields with values required to properly describe a dataset e.g., a dataset with technical identifiers such as *title or description* without any metadata context is incomplete and reduce the quality of the dataset. Metadata/description of a dataset is expected to be Findable, Accessible, Interoperable and Reusable (FAIR) [Wilkinson, Dumontier, Aalbersberg, Appleton, Axton, Baak, Blomberg, Boiten, da Silva Santos, Bourne, et al. 2016]. For instance, indicating the description of the dataset about *France* that it captures intrinsic properties of the country or all the editors that modified the dataset, thus, making sure the metadata is available and complete.

Definition 3.3.6 (Metadata completeness)**.** *Metadata completeness is the degree to which metadata properties and values are not missing in a dataset for a given task.*

Problems. Open Data platforms such as LOD Cloud and Governmental Open Data terminal are becoming widespread and important data source for research. [Neumaier, Umbrich, and Polleres 2016] mentioned that metadata quality issues in Open Data portals have been identified as one of the core problems for wider adoption of Open Data and developed a quality assessment and evolution monitoring framework for web-based data portal platforms, which offers their metadata in different and heterogeneous models. Similarly, [Assaf, Senart, and Troncy 2016; Ellefi, Bellahsene, Breslin, Demidova, Dietze, Szymanski, and Todorov 2016] investigated the development of frameworks for assessing metadata quality in Linked Data sources and [Kubler, Robert, Neumaier, Umbrich, and Le Traon 2018; Assaf, Troncy, and Senart 2015] focused on assessing Government Open Data platforms.

Approaches and metrics. There are six articles that focus on metadata completeness. A lot of emphasis is currently placed on the completeness of the dataset itself only, but the importance of completeness of the metadata cannot be understated. Descriptive metadata about existing datasets are a substantial building block for facilitating entities and datasets linking, entity retrieval, distributed search or query federation. Ellefi et al. [Ellefi, Bellahsene, Breslin, Demidova, Dietze, Szymanski, and Todorov 2016] developed a framework, for dataset profiling, for the formal representation of a set of features that describes a dataset and allow the comparison of different datasets. They provided a taxonomy, formally represented as an RDF vocabulary of dataset profiling features. Furthermore, [Charles, Stiller, Kiraly, and Bailer 2018] presented an approach for capturing multilingualism as part of data quality dimensions, spanning completeness and [Kubler, Robert, Neumaier, Umbrich, and Le Traon 2018] proposed a framework for comparison of Open Data portals for metadata quality using analytic hierarchy process.

(vii) Labelling completeness

Labelling completeness is particular to RDF Data, where URIs are used for identification but are not very suitable for indexing purposes and human readability. This warrants that entities have a human readable label and the level for which it is not

missing is labelling completeness. For example, the existence of *rdfs:label* for instance of a city *Paris*.

Definition 3.3.7 (Labelling completeness)**.** *Labelling completeness refers to the degree to which entities in the dataset have a human readable label.*

Problems. [Ell, Vrandečić, and Simperl 2011] investigated internationalization of knowledge bases, existence of multiple labels for an entity, the computational costs associated with dereferencing URIs and proposed that all entities in a knowledge base have human readable labels. They also explored development of a metric for labelling completeness. [Gürdür, El-khoury, and Nyberg 2018] was focused on enterprise data integration and investigated the assessment of data quality for a Linked Enterprise Data in the automotive industry. Authors calculated the average number of resources that did not have a label property defined and stressed the need for it due to the heterogeneous nature of environments that generates different components of the dataset.

Approaches and metrics. Non-information resources are abstract ideas represented in LD dataset that may possess URI but cannot be directly accessed or downloaded via the internet, such as person. The importance of labels for non-information resources in Linked Data cannot be overstated. It helps indexing and searching the resources and displaying data to end-users that can be easily understood, rather than URIs [Ell, Vrandečić, and Simperl 2011]. Only three articles have examined labelling completeness. According to Ell et al. [Ell, Vrandečić, and Simperl 2011], Labelling completeness measures the degree to which Linked Data resources have labels, it can be defined as the ratio of URIs with at least one value for a labeling property to all URIs in a given knowledge base. [Gürdür, El-khoury, and Nyberg 2018] developed a data quality assessment tool in the form of a dashboard to manage data quality of a dataset integrated from various departments of an organization that is part of the automotive industry. Finally, [Assaf, Senart, and Troncy 2016] presented a quality measurement tool that helps data providers to rate the quality of

their datasets and get recommendations on possible improvements by developing standardized quality indicators to rate datasets.

Tools analysis. From the core articles, we identified nine most common used tools (listed in Figure 3.8) that automatically or semi-automatically assess completeness of datasets. An overview of different tools and their capabilities along with the type of completeness they focus on, is described below.

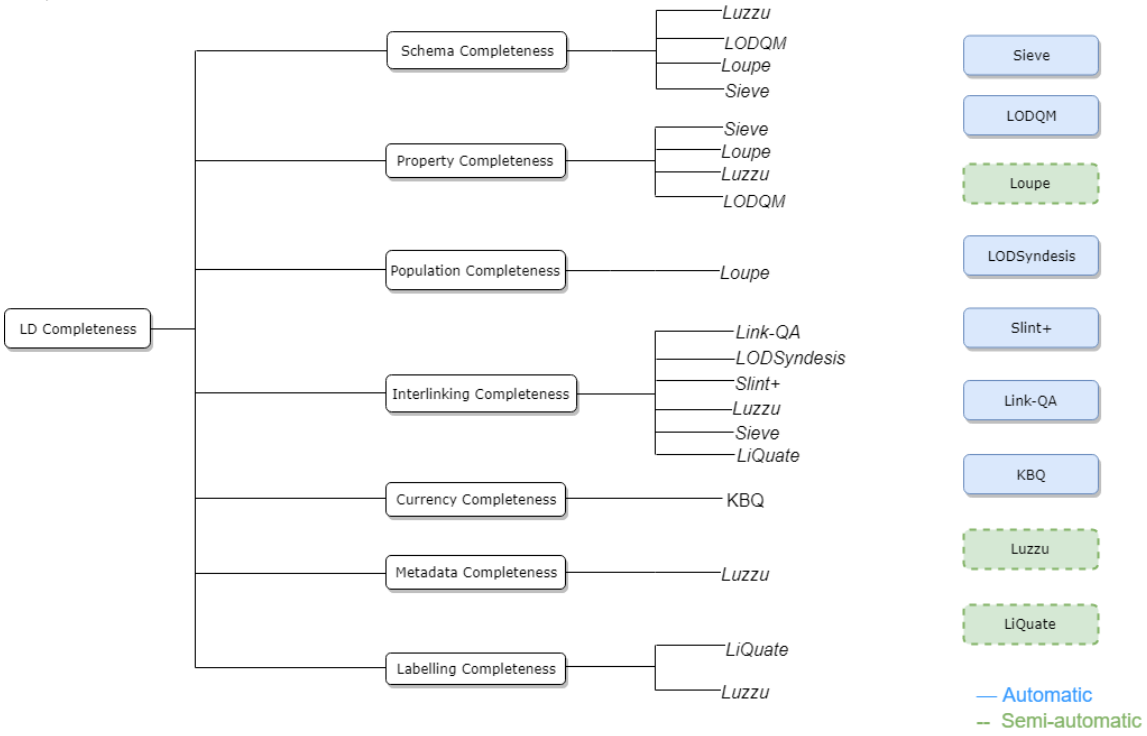

Figure 3.8: Summary of tools based on type of completeness.

- Sieve

  Sieve[16] [Mendes, Mühleisen, and Bizer 2012a] is the quality evaluation module within Linked Data Integration Framework (LDIF) [Schultz, Matteini, Isele, Mendes, Bizer, and Becker 2012] which enables automatic data quality assessment by a conceptual model composed of assessment metrics, indicators and scoring functions like (Set Membership, Threshold and Interval Membership) for completeness. It is suitable for assessing schema, property and interlinking completeness.

[16] http://sieve.wbsg.de

- Loupe

  Loupe[17] [Radulovic, Mihindukulasooriya, García-Castro, and Gómez-Pérez 2018] is a tool that can be used to inspect a dataset to understand which vocabularies (classes and properties) are used with statistics and frequent triple patterns. Starting from the high-level statistics, Loupe allows one to zoom into details down to the corresponding triple with its visual explorer and semi. It is a semi-automatic tool as metrics needs customization by the user. This can be used to assess schema and property completeness. Population completeness can also be assessed with external data.

- Luzzu

  Luzzu[18] [Thakkar, Endris, Gimenez Garica, Debattista, Lange, and Auer 2016] is also a Quality Assessment Framework for linked Open datasets based on the Dataset Quality Ontology (daQ), allowing users to define their own quality metrics. It provides a library of generic quality metrics that users can customize based on domain-specific tasks in a scalable manner, thus it is semi-automatic; it also provides queryable quality metadata on the assessed datasets and assembles detailed quality reports on assessed datasets. It is useful for assessing property, schema, interlinking, metadata and labelling completeness albeit it requires manual configurations by the user.

- Link-QA

  Link-QA[19] [Guéret, Groth, Stadler, and Lehmann 2012] specifies a framework for detection of the quality of linksets using network metrics (degree, clustering coefficient, open sameAs chains, centrality, description richness through sameAs). It is completely automatic and compatible with a set of resources, SPARQL endpoints and/or dereferencable resources and a set of triples as input. It is particularly useful for assessing interlinking completeness.

- LiQuate

[17] http://loupe.linkeddata.es/loupe/
[18] https://eis-bonn.github.io/Luzzu/
[19] https://github.com/cgueret/LinkedData-QA

LiQuate [Ruckhaus, Vidal, Castillo, Burguillos, and Baldizan 2014] is a tool that uses a Bayesian Network to learn the dependencies between properties in RDF data. It is particularly well suited to assess interlinking and labelling completeness. However, it is semi-automatic and requires configurations from the user.

- LODsyndesis

  LODsyndesis[20] [Mountantonakis and Tzitzikas 2017] uses novel lattice-based algorithms to find the intersection of datasets in the LOD cloud. The symmetric and transitive closure of the set of *owl:sameAs*, *owl:equivalentProperty* and *owl:equivalentClass* relationships from all datasets was computed for creating semantically enriched indexes. It is a reference for automatically assessing interlinking completeness.

- Slint+

  Slint+[21] [Nguyen, Ichise, and Le 2013] (Schema-Independent Linked Data Interlinking) is similar to LODsyndesis, in that, it detects all owl:sameAs links automatically between two given Linked Data sources.

- LODQM

  LODQM[22] [Behkamal 2014] is an automatic tool developed around goal-question-metric [Basili, Caldiera, and Rombach 1994] approach to soliciting metrics used for assessment of datasets. It is suited for assessing schema and property completeness.

- KBQ

  KBQ[23] [Rizzo, Torchiano, and Torino 2017] is a tool geared towards assessment of quality of datasets based on temporal analysis. It automatically computes the frequency of predicates and the frequency of entities of a given resource type, and compares the frequencies with the ones observed in previous versions of the dataset. It can be specifically used to assess currency completeness.

[20] http://83.212.101.188:8081/LODsyndesis/index.jsp
[21] http://ri-www.nii.ac.jp/SLINT/index.html
[22] https://bitbucket.org/behkamal/new-metrics-codes/src
[23] http://datascience.ismb.it/shiny/KBQ/

3.3.1.3 Discussion

Overview. In this SLR, we have analyzed 52 articles that focus on seven types of LD completeness. In total, we identified 23 metrics and nine tools that specifically deal with LD completeness. We have observed that some articles examine one type of completeness such as [Jaffri, Glaser, and Millard 2008; Issa, Paris, and Hamdi 2017b; Ell, Vrandečić, and Simperl 2011] or several types of completeness such as [Mendes, Mühleisen, and Bizer 2012a; Knap and Michelfeit 2012; Färber, Bartscherer, Menne, and Rettinger 2018]. Furthermore, research is rarely entirely focused on a single aspect of data quality dimension. Among the nine tools analyzed, we discovered six tools that are automatic and the remaining three tools are semi-automatic. Out of the tools LiQuate does not seem to be online or it is no longer supported while all other are online. Also, there was no formal validation of the methodologies that were implemented as tools. LD completeness challenges tackled in the literature is most often in the development of frameworks for assessment of data quality using various approaches, ranging from the application of network measures [Guéret, Groth, Stadler, and Lehmann 2012] to first order logic predicates [Bronselaer, De Mol, and De Tre 2018]. In other scenarios, researchers define certain constraints applicable to a LD dataset, such as in the case of a privacy aware assessment framework [Cappiello, Di Noia, Marcu, and Matera 2016]. Based on our analysis, we have identified several open challenges pertaining to Linked Data quality in general and also specifically for the completeness dimension, which we discuss in the following.

Open world assumption. Typically the Semantic Web follows an Open World Assumption (OWA) [Drummond and Shearer 2006], which does not allow inferring the truth of a statement only by checking whether the statement is known. OWA assumes that everything we do not know is not yet defined. For data quality assessment, we need to define metrics based on close-world assumption, i.e., assume that everything that is not known can be assumed as false. However, this assumption will most likely not hold in many cases since we often suffer from lack of gold standard and complete data. Consequently, when performing data quality assessment, the metrics have to be evaluated

and refined continuously [Fürber and Hepp 2011a].

Maintenance of data quality. After the assessment of data quality, the next step is to improve the quality taking into account the results from the assessment. This cycle of assessment and improvement should be done at regular intervals of time and/or when the data is updated. Additionally, a data quality issue in one dataset can ultimately affect the quality of multiple interlinked data sets, thus propagating the errors. Consequently, maintenance of quality becomes challenging in the Web of Data mainly because it is generated from existing data and thus its correction can be even more difficult and time consuming when meta-information provenance is not available anymore [Zaveri, Maurino, and Equille 2014].

Quality-based question answering. When existing Linked Data sets are published along with their quality information, it can be possible to design a new generation of quality-based question answer systems, which rely on this information to deliver useful and relevant results [Naumann 2002]. In order to provide the answer to user queries in a meaningful way, it is necessary to define what should be in the result, how it can be obtained and how one should represent the query result. In this case, the completeness, consistency (logical/formal), timeliness, etc. of the data affects the results considerably. For example, querying an integrated data set for a particular flight time, the time from the source with the higher update frequency and more complete information should be chosen. Thus, question answering can be increased in effectiveness and efficiency using data quality criteria as a leverage to filter the most relevant results.

Stream-lining future surveys. This SLR took eight months in total to be performed. In order to increase the efficiency and sustainability of such SLRs in the future, we propose (i) future surveys on Linked Data quality, specifically on completeness, tag their articles with the type of completeness (listed in Section 3.3.1.2) as keywords and (ii) we, as a community, think of combining human and machine effort towards stream-lining such SLRs. We resonate with the idea proposed in [Thomas, Noel-Storr, Marshall, Wallace, McDonald,

Mavergames, Glasziou, Shemilt, Synnot, Turner, and Elliott 2017] of *living systematic reviews* combining humans and machines. For some of the repetitive and labor-intensive tasks, machines can assist, such as, for searching relevant articles and eligibility analyzing. Then, humans can assist in extracting relevant information from within the text. Workflows can be developed in which human effort and machine automation can each enables the other to operate in more effective and efficient ways, offering substantial enhancements to the productivity of systematic reviews [Thomas, Noel-Storr, Marshall, Wallace, McDonald, Mavergames, Glasziou, Shemilt, Synnot, Turner, and Elliott 2017]. In this way, a systematic literature review can be continually updated incorporating new articles as they become available.

3.3.2 Linked Data conciseness

An ontology is an essential component of Semantic Web vision. It represents a set of concepts and the relationships as a description of knowledge. Ontologies generally provide certain elements such as classes, instances and relations in addition to axioms and rules.

Due to the decentralized nature of the Semantic Web, through creating ontologies in different formats and languages in order to represent the data, ontologies are highly heterogeneous. This heterogeneity is having a negative effect on the quality of the data itself such as inconsistencies. Data heterogeneity is considered at three categories; syntax, structure and semantics [Stuckenschmidt and Van Harmelen 2005] that occurs at both schema and instance levels. Ontology alignment is a solution to the semantic heterogeneity issue [Shvaiko and Euzenat 2011] by determining correspondences between entities that are semantically related from any two ontologies. Each correspondence consists of two entities (e.g., classes, properties or instances) with a relation such as equivalence ($\equiv$), subsumption ($\sqsubseteq$,$\sqsupseteq$) or disjointedness ($\perp$) between them. Figure 3.9 shows an example of ontology alignment as defined in [Shvaiko and Euzenat 2011]. Correspondences are illustrated as left-right arrows that link between an entity from O_1 and an entity from O_2 tagged by the relation.

In this thesis, we are interested in finding the equivalent properties (i.e., predicates) to

Table 3.4: List of completeness metrics.

Type	Metric
Schema completeness	* Ratio of number of classes/properties presented in dataset to total number of classes/properties * Ratio of number of properties of an instance to total number of mandatory properties * Capacity function to validate predicate of completeness at schema level * Ratio of similar subjects/instances missing properties
Property completeness	* Percentage of values for which a given property exists * Ratio of number of values presented for a specific property to total number of values for a specific property * Completeness measurement based on statistical distributions of properties * Count of property values * Ratio of concepts/predicate pairs
Population completeness	* Ratio of unique objects on the dataset to all available unique objects in the universe * Multiplicity of the resource and the aggregated multiplicity of all classes where the resource belongs to * Identify a fragment of completeness information to check completeness
Interlinking completeness	* Ratio of instances that are interlinked in the dataset to total number of instances in the dataset * Linkset importing * owl:sameAs frequency relative to co-reference * Ratio between the number of triples that are “in-links” and the total number of triples in the RDF graph served as a description of each resource * Extent of connectivity between the dataset under assessment and external sources
Currency completeness	* Ratio of unique triples in new version of KB to total unique triples over all versions of the KB * Difference between frequency of properties for a class between two KB releases
Metadata completeness	* Aggregate function on predicates on metadata * Existence , Count of metadata values and ratio of missing metadata values to total metadata properties
Labelling completeness	* Percentage of URIs with label * Existence, count of labels

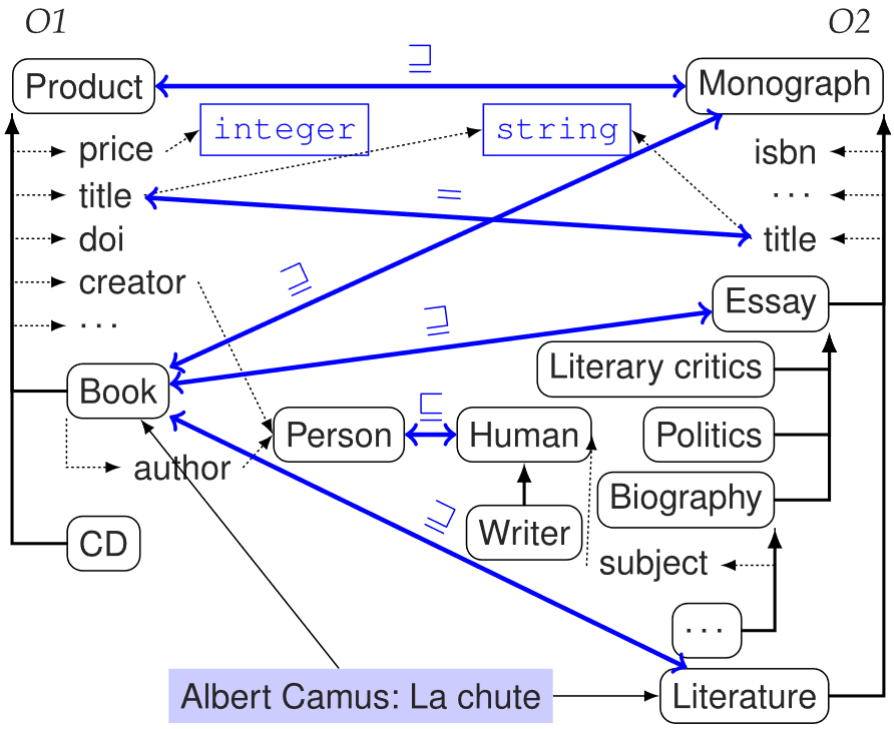

Figure 3.9: An alignment between two ontologies [Shvaiko and Euzenat 2011].

evaluate the conciseness dimension. Whereas researches in linking RDF at the schema and instance levels have been investigated in the recent past, property alignment has not received much attention yet [Gunaratna, Thirunarayan, Jain, Sheth, and Wijeratne 2013]. For this reason, in this work we focus only on the equivalent predicate. There are some studies on mapping properties across RDF sources [Fu, Wang, Jin, and Yu 2012; Gunaratna, Thirunarayan, Jain, Sheth, and Wijeratne 2013; Zhang, Gentile, Blomqvist, Augenstein, and Ciravegna 2017]. In [Fu, Wang, Jin, and Yu 2012], the authors identified similar relations using subject and object overlap of predicates. On the other hand, [Gunaratna, Thirunarayan, Jain, Sheth, and Wijeratne 2013] also used statistical measures to identify strictly equivalent relations rather than similarity in general.

In [Zhang, Gentile, Blomqvist, Augenstein, and Ciravegna 2017; Galárraga, Teflioudi, Hose, and Suchanek 2015], authors provided methods to group equivalent predicates by clustering synonymous relations. PARIS [Suchanek, Abiteboul, and Senellart 2011] is another well-known approach which needs to be mentioned in this context. It combined

instance and schema matching using probabilities with high accuracy.

Conciseness is an aspect of Linked Data quality dimensions which basically aims to avoid repetition elements. The eliminating of the synonymously used predicates aims to optimize the dataset to speed up processing. Mendes et al. [Mendes, Mühleisen, and Bizer 2012b] categorized the conciseness dimension into *intensional* and *extensional* conciseness. The first type, which is the intensional conciseness, measures a number of unique dataset attributes to the total number of schema attributes, thus, this measurement is represented at the schema level. In a similar manner but at the instance level, extensional conciseness measures the number of unique objects to the real number of objects in the dataset. In [Lei, Uren, and Motta 2007], the authors proposed an assessment metric by detecting the number of ambiguous instances according to those of semantic metadata sets in order to discover the duplicated instances. In the similar sense but under a different name, Füber et Hepp [Fürber and Hepp 2011] defined the elements of representation such as classes, predicates and objects under the domination of "uniqueness". Their definition suggested uniqueness of both breadth and depth at schema and instance level, respectively.

Indeed, most proposed metrics of conciseness assessment are based on a simple ratio, which compares the proportion of existing elements to the overall ones [Zaveri, Rula, Maurino, Pietrobon, Lehmann, and Auer 2016]. Whereas, the conciseness at schema level is measured by the number of unique predicates and dataset classes to the total number of those elements in a schema [Mendes, Mühleisen, and Bizer 2012b]. At instance level, it measures the number of unique instances to the total number of instances in the dataset [Mendes, Mühleisen, and Bizer 2012b; Fürber and Hepp 2011; Lei, Uren, and Motta 2007]. In [Abedjan and Naumann 2013], the authors proposed an algorithm to discover the synonym predicates for query expansions. This algorithm is based on mining similar predicates according to their subjects and objects. Abedjan and Naumann suppose that synonymous predicates do not co-occur for the same subject instance. For example, triples about an instance would not use equivalent predicates such as *residence* and *livesIn*. This assumption is based on the concept that usually a specific predicate among the analogous synonyms are used for the same instance, as it is uncommon that two synonym predicates to be used for the same instance. However, their approach evaluated on a small dataset

and proposed a lot of predicates that are not synonyms (false positives).

Recently, Kalo et al. [Kalo, Ehler, and Balke 2019] presented a classification method based on knowledge embedding technique to discover synonym predicates. They employed similarity metrics to detect semantic similarity of predicates and classify predicate pairs as synonymous or non-synonymous predicates. Their approach is limited to predicates that are *ObjectProperty* type which relates between resources. In this thesis, we focus on extracting equivalent predicates regardless of their types from RDF datasets to assess the conciseness at schema level.

3.4 Summary

In this chapter, we have explained some concepts that are related to the work in this thesis. Furthermore, we have shown the problems and approaches found to enhance a part of Linked Data quality. Two dimensions of Linked Data quality have been highlighted, the completeness dimension that refers to the amount of information present in a particular dataset, and conciseness dimension that refers to the redundancy of entities at schema and instance levels.

We can explore certain aspects of detecting and mitigating the completeness data quality issue in a more thorough fashion, for example, exploring whether completeness is better tackled in the data collection or integration processes. Completeness dimension includes different types of completeness as described above. One of our objectives is to assess Linked Data completeness at schema level. We propose an approach to discover a reference schema based on data sources. This discovered schema is used to assess the completeness of the dataset. Furthermore, we have developed a prototype to display the conceptual schemas according to user's requirements as it will be illustrated in Chapter 4.

Concerning the conciseness dimension, we have illustrated that this dimension is classified into schema and instance levels. Our proposed idea focuses only on a schema level and more specifically finding equivalent predicates. The conciseness of data is one of these dimensions that should be extensively examined as it prevents data redundancy. Our approach consists of three sequential analyses to discover the equivalent predicates as

it will be shown in Chapter 5.

In this thesis, our objective is to provide new approaches to assess and improve completeness and conciseness of Linked Data. The following chapters describe in detail our various contributions.

Chapter 4

Assessing Completeness of RDF Datasets

This chapter explains our proposed approach to calculate the completeness of RDF datasets. It starts with a motivating example that illustrates the main idea behind our approach. The second section explains the proposed approach that consists of two steps: properties mining and completeness calculation. Section 4.3 describes the objective of our prototype (called *LOD-CM*) and the steps taken to generate reference conceptual schemas. It also illustrates the user's interface of the prototype and presents two use cases of *LOD-CM*. Finally, Section 4.4 summarizes the main points.

4.1 Motivating scenario

We illustrate in this section an example that shows the issues and the difficulties encountered in the calculation of a dataset completeness. Let us consider the set of scientists described in the well-known open linked dataset, DBpedia, we would like to know when the user aims to query this dataset about a scientist (or a subset of scientists), if the information provided for this scientist are complete (well described) or not. Let us give an example a subset of 100 scientists, the pseudo-code 1 returns, for each scientist, the couples $\langle property, value \rangle$.

To evaluate the completeness of this subset, a first intuition could consist of comparing the properties used in the description of each scientist with a reference scientist schema

Algorithm 1 Scientists Descriptions.

```
String Query1 = "SELECT ?subject where{
                 ?subject rdf:type dbo:Scientist
                 } LIMIT 100"
Result S = ExecQuery(Query1)
for each subject ∈ S do
    String Query2 = "SELECT ?property ?value where{
                     subject ?property ?value}"
    Result R = ExecQuery(Query2)
    Descriptions.put(subject, < property, value >)
return Descriptions
```

(ontology). For example, in DBpedia, the class *Scientist*[24] has a number of properties (e.g., *doctoralAdvisor*), but these properties are not the only ones used in the description of a scientist (e.g., the property *birthDate* is not present in this list). Indeed, the super class of the class *Scientist* is called *Person*. Thus, the description of a scientist may also take into account the properties of this class. Therefore, to obtain an exhaustive list of the whole properties used in the description of a scientist, we have to calculate the union of the set of properties of the class *Scientist* and all its ancestors. For our example, the reference scientist schema that we called *Scientist_Schema* could be calculated as follows:

$$Scientist\_Schema = \{Properties\ on\ Scientist\} \cup$$
$$\{Properties\ on\ Person\} \cup \{Properties\ on\ Agent\} \cup$$
$$\{Properties\ on\ Thing\}$$

such that: $Scientist \sqsubseteq Person \sqsubseteq Agent \sqsubseteq Thing$

Thus, the completeness of a scientist description (e.g., *Albert_Einstein*) will be the proportion of properties used in the description of this scientist to the total number of properties in *Scientist_Schema*. In the case of DBpedia, with a simple SPARQL query[25], we can obtain the size of *Scientist_Schema*, which is equal to 664 (A-Box properties). Thus, the completeness of the description of the instance *Albert_Einstein* could be calculated as

[24] http://mappings.dbpedia.org/server/ontology/classes/
[25] Performed on: http://dbpedia.org/sparql

follows:

$$Comp(Albert\_Einstein) = \frac{|Properties\ on\ Albert\_Einstein|}{|Scientist\_Schema|} = \frac{21}{664} = 3,16\%$$

However, for example, the property *weapon* is in *Scientist_Schema*, but is not relevant to the instance *Albert_Einstein*.

We can finally conclude that, the completeness as calculated here, does not provide us with the relevant value regarding the real representation of scientists in DBpedia dataset. Hence, to overcome this issue, we may have to explore those instances. We want to find which properties are used more often than others to describe instances of a given type. Based on data mining, the approach that we propose in this work deals with this issue by extracting, from a set of instances (of the same class), the set of the most representative properties and calculates completeness in respect to this set.

4.2 Completeness computation: A mining-based approach

To assess the completeness of the different version of an LOD dataset (as DBpedia), we propose an approach that calculates the completeness of an input dataset by posing the problem as an itemset mining problem. In fact, the completeness at instances level assesses missing values [Pipino, Lee, and Wang 2002]. This requires a schema (e.g., a set of properties) to be inferred from the data source. However, it is not relevant to consider, for a subset of resources, the schema as the union of all properties used in their description as seen in Section 4.1. Indeed, this vision disregards the fact that missing values could express inapplicability.

1. **Properties mining**: Given a dataset $\mathcal{D}$, we first represent the properties, used for the description of the $\mathcal{D}$ instances, as a transaction vector. Then, we apply the well-known FP-growth algorithm [Han, Pei, Yin, and Mao 2004] for mining frequent itemsets. We have chosen FP-growth for efficiency reasons, but any other itemset mining algorithm could, obviously, be used. Only a subset of these frequent itemsets, called "Maximal" [Grahne and Zhu 2003], is captured. This choice is motivated

by the fact that, on the one hand, we are interested in important properties for a given class that should appear often and, on the other hand, the number of frequent patterns could be exponential when the transaction vector is very large (see Section 4.2.1 for details).

2. **Completeness calculation**: Once the set of Maximal Frequent Patterns $\mathcal{MFP}$ is generated, we use the apparition frequency of items (properties) in $\mathcal{MFP}$ to give each of them a weight that reflects how important the set of properties is to the description of instances. Weights are then exploited to calculate the completeness of each transaction (regarding the presence or absence of properties) and, hence, the completeness of the whole dataset.

In the following, we give a detailed description of each step.

4.2.1 Properties mining

In this step, the objective is to find the properties sets that are the most shared by the subset of instances extracted from a dataset. Our assumption is that a property often used by several instances of a given type is more important than less often used properties for the same instances. This set will be then used to calculate a completeness value.

More formally, let $\mathcal{D}(C, I_C, P)$ be a dataset, where C is the set of classes (e.g., rdf:type such as *Actor*, *City*), I_C is the set of instances for categories in C (e.g., *Ben_Affleck* is an instance of the class *Actor*), and $P = \{p_1, p_2, ..., p_n\}$ is the set of properties (e.g., *residence*(*Person*, *Place*)). Let $\mathcal{I}'$ be a subset of data (instances) extracted from $\mathcal{D}$ with $\mathcal{I}' \subseteq I_C$, we first initialize $\mathcal{T} = \phi$, $\mathcal{MFP} = \phi$. For each $i \in \mathcal{I}'$ we generate a transaction t. Indeed, each instance i is related to values (either resources or literals) through a set of properties. Therefore, a transaction t_k of an instance i_k is a set of properties such that $t_k \subseteq P$. Transactions generated for all the instances of $\mathcal{I}'$ are then added to the set $\mathcal{T}$.

Example 4.2.1. *Taking Table 4.1, let $\mathcal{I}'$ be a subset of instances such that:*

Table 4.1: A sample of triples from DBpedia.

Subject	Predicate	Object
The_Godfather	director	Francis_Ford_Coppola
The_Godfather	musicComposer	Nino_Rota
Goodfellas	director	Martin_Scorsese
Goodfellas	editing	Thelma_Schoonmaker
True_Lies	director	James_Cameron
True_Lies	editing	Conrad_Buff_IV
True_Lies	musicComposer	Brad_Fiedel

Table 4.2: Transactions created from triples.

Instance	Transaction
The_Godfather	director, musicComposer
Goodfellas	director, editing
True_Lies	director, editing, musicComposer

$\mathcal{I}' = \{The\_Godfather, Goodfellas, True\_Lies\}$. *The set of transaction* $\mathcal{T}$ *would be:*

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

The objective is then to compute the set of frequent patterns $\mathcal{FP}$ from the transaction vector $\mathcal{T}$.

Definition 4.2.1. *(Pattern) Let* $\mathcal{T}$ *be a set of transactions, a pattern* $\hat{P}$ *is a sequence of properties shared by one or several transactions t in* $\mathcal{T}$.

For any pattern $\hat{P}$ (e.g., a set of properties), let $T(\hat{P}) = \{t \in \mathcal{T} \mid \hat{P} \subseteq E(t)\}$ be the corresponding set of transactions. $|T(\hat{P})|$ is the *support* of $\hat{P}$ (e.g., the number of individuals having all properties of $\hat{P}$). A pattern $\hat{P}$ is frequent if $\frac{1}{|\mathcal{T}|}|T(\hat{P})| \geq \xi$, where ξ is a user-specified threshold.

Example 4.2.2. *Taking Table 4.2, let* $\hat{P} = \{director, musicComposer\}$ *and* $\xi = 60\%$. $\hat{P}$ *is frequent as its relative support (66.7%) is greater than* ξ.

To find all the frequent patterns $\mathcal{FP}$, we used, as mentioned above, the FP-growth itemsets mining algorithm. However, according to the size of the transactions vector, the

FP-growth algorithm generates a very large $\mathcal{FP}$ set. In itemset mining, a concept, called "Maximal" frequent patterns, allows us to find those subsets of properties. Thus, to reduce $\mathcal{FP}$, we generate a subset containing only "Maximal" patterns.

Definition 4.2.2. *($\mathcal{MFP}$) Let $\hat{P}$ be a frequent pattern. $\hat{P}$ is maximal if none of its proper superset is frequent. We define the set of Maximal Frequent Patterns $\mathcal{MFP}$ as:*

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

Example 4.2.3. *Taking Table 4.2, let ξ = 60% and the set of frequent patterns $\mathcal{FP}$ = {{director},{musicComposer},{editing}, {director,musicComposer},{director,editing}}. The $\mathcal{MFP}$ set would be: $\mathcal{MFP}$={{director,musicComposer},{director, editing}}*

4.2.2 Completeness calculation

In this step, we carry out for each transaction a comparison between its corresponding properties and each pattern of the $\mathcal{MFP}$ set, regarding the presence or the absence of the pattern. An average is, therefore, calculated to obtain the completeness of each transaction $t \in \mathcal{T}$. Finally, the completeness of the whole $t \in \mathcal{T}$ will be the average of all the completeness values calculated for each transaction.

Definition 4.2.3. *(Completeness) Let $\mathcal{I}'$ a subset of instances, $\mathcal{T}$ the set of transactions constructed from $\mathcal{I}'$, and $\mathcal{MFP}$ a set of maximal frequent pattern. The completeness of $\mathcal{I}'$ corresponds to the completeness of its transaction vector $\mathcal{T}$ obtained by calculating the average of the completeness of $\mathcal{T}$ regarding each pattern in $\mathcal{MFP}$. Therefore, we define the completeness $\mathcal{CP}$ of a subset of instances $\mathcal{I}'$ as follows:*

$$\mathcal{CP}(\mathcal{I}') = \frac{1}{|\mathcal{T}|} \sum_{k=1}^{|\mathcal{T}|} \sum_{j=1}^{|\mathcal{MFP}|} \frac{\delta(E(t_k), \hat{P}_j)}{|\mathcal{MFP}|} \tag{4.1}$$

$$\textit{such that: } \hat{P}_j \in \mathcal{MFP}, \textit{ and } \delta(E(t_k), \hat{P}_j) = \begin{cases} 1 & \textit{if } \hat{P}_j \subset E(t_k) \\ 0 & \textit{otherwise} \end{cases}$$

Example 4.2.4. *Let ξ = 60%, the completeness of the subset of instances in Table 4.2 regarding $\mathcal{MFP}$ = {{director,musicComposer}, {director, editing}} , would be:*

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

This value corresponds to the completeness average value for the whole dataset regarding the inferred patterns in $\mathcal{MFP}$.

4.3 Prototype: LOD-CM

Recognizing that conceptual modeling is a powerful tool for data understanding, our proposal addresses the problem of deriving a conceptual schema from RDF data. By exploring instances, our approach integrates a completeness measurement as a quality criterion to ensure the relevancy of the derived schema, because data from RDF datasets is the result of a free individual publication effort. The result would be a conceptual schema enriched with completeness values.

4.3.1 Overview

In this section, we are interested in conceptual modeling of RDF Data (Resource Description Framework) [Klyne and Carroll 2006]. Our objective is to define an approach for deriving conceptual schemas from existing data. The proposed solution should cope with the essential characteristics of a conceptual schema that have the ability to make an abstraction of relevant aspects from the universe of discourse and meeting the user's requirements [Rolland and Prakash 2000].

The approach that we propose takes into account the two facets; namely the universe of discourse represented by the data sources, and the user's needs represented by the user's decisions during the conceptual schema construction. As the model should express the meaningful state of the considered dataset, we rely on a mining approach leading to taking into consideration the data model from a more frequent combination of properties. The relevancy of properties is handled by integrating a completeness measurement solution that drives the identification of relevant properties [Hamdi and Cherfi 2015; Issa, Paris, and Hamdi 2017a]. To meet the user's requirements, we propose to construct the conceptual schema on a *scratch card* manner where the user decides about the parts of the conceptual schema to reveal according to her needs and constraints. The main contributions are:

1. We use a mining approach to infer a model from data, as we consider that no

predefined schema exists. The underlying assumption is that the more frequent a schema is, the more representative for the dataset it is.

2. We introduce a novel approach, called *LOD-CM*, for Conceptual Model mining based on quality measures and on completeness measures as a way to drive the conceptual schema mining process.

3. We experimentally assess our approach by using real-world datasets.

4.3.2 Conceptual schema derivation

To illustrate our proposed approach, let us consider a user willing to obtain a list of artists with their names and birth places from an RDF data source; to do so, she can write the following SPARQL query[26]:

Listing 4.1 – Retrieve a list of artists with their birth places

```
SELECT * WHERE {
       ?actor rdf:type dbo:Actor .
       ?actor foaf:name ?name .
       ?actor dbo:birthPlace ?birthPlace .
       }
```

Writing such a query is much more difficult in a Linked Open Data (LOD) source context than in a relational database one. In a relational context, the database schema is predefined and the user writing the query is aware of it. In an LOD context, in addition to the fact that the schema does not exist, there is another problem related to data completeness. Actually, the expressed query returns only the list of actors having values for all the properties listed in the query. In our example, only actors having values for both *foaf:name* and *dbo:birthPlace* are included in the result. Knowing that at most 74% of actors have a value for *dbo:birthPlace*, the user should probably appreciate getting this information to add `OPTIONAL` parameters to the second pattern of the query and obtain more results. Besides, she would be aware of the fact that the result is complete to a certain degree (i.e., *dbo:birthPlace* is present in only 74% of actors).

[26]Performed on: `http://dbpedia.org/sparql`

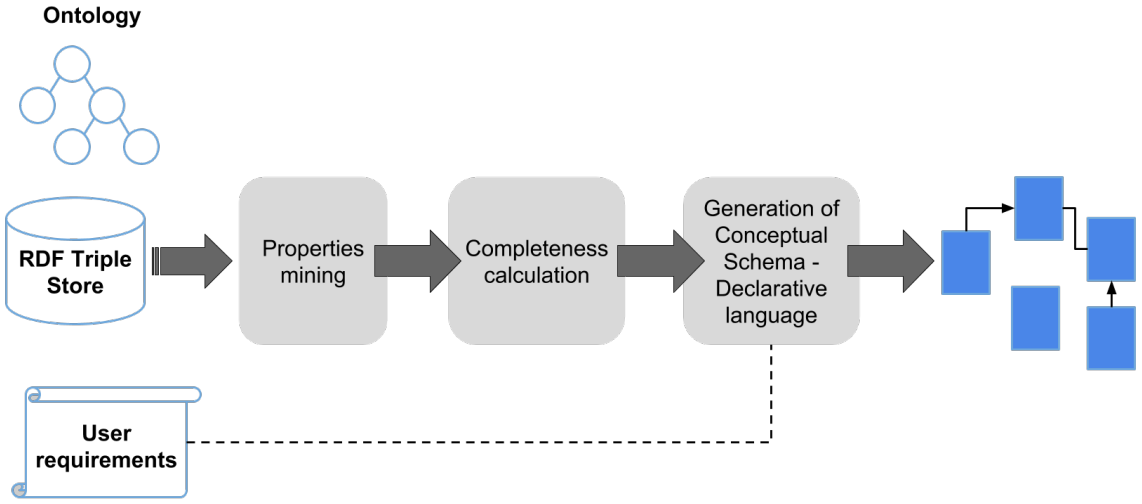

Figure 4.1: The *LOD-CM* workflow.

To tackle these two problems, we propose an approach that aims to help "revealing" a conceptual schema from a LOD RDF source. This conceptual schema is driven by the user for both its content and completeness quality values.
In the context of the Web of Data, most of the datasets published are described by models called, in the linked data jargon, vocabularies (or ontologies). However, these models are not used in a prescriptive manner. Consequently, a person who publishes data is not constrained by the underlying ontology leading to sparse descriptions of concepts. For example, the category *Actor* from DBpedia has around 532 properties that are not equally relevant.

From these observations, it is clear that checking data (instances) is necessary to infer a relevant model that can be used to guarantee, for example, an expected completeness value. The approach that we propose deals with this issue through an iterative process which infers a conceptual schema complying the expected completeness. Figure 4.1 gives an overview of this process.

The process of inferring a conceptual schema goes through four steps: First, a subset of data that corresponds to the user's scope is extracted from the triple store (cf. Section 4.3.3). This subset is then transformed into transactions and a mining algorithm is applied. In our approach, for efficiency reasons, we have chosen the well-known FP-growth algorithm

[Han, Pei, and Yin 2000; Han, Pei, Yin, and Mao 2004] as illustrated in Section 4.2.1. From the generated frequent itemsets, only a subset of these maximal frequent itemsets is considered as previously explained in Section 4.2.2 .

4.3.3 Scope and completeness specification

To generate the conceptual schema, a subset of data is extracted from the triple store. This subset could correspond to a category or a set of categories such as *Actor*, *Film* or *Organization*. This defines what we call the user's scope that corresponds to the categories that the user plans to use in a query, to the information she wants to explore or any kind of usage based on data consumption.

The user is also asked to indicate the degree of the desired completeness. Indeed, properties for a given category are not equally valuated. For example, for the category *Artist*, the property *foaf:name* has a value for 99% of the instances whereas the property *dbo:birthPlace* has a value for at most 74% of the instances. Our approach gives the possibility to express a constraint on the completeness values desired for mined properties and associations. Once the categories are identified, the data is converted into transaction vectors and a mining algorithm is applied to obtain a set of frequent itemsets.

Table 4.1 illustrates some instances of the category *Film* in the form of triples, taken from DBpedia. Each category is described by a set of properties (predicates) and each instance of this category could have a value for all the properties or only for a subset of these properties, this subset is called transaction. Table 4.2 represents the set of transactions constructed from the triples of Table 4.1.

Let $\mathcal{T} = \{t_1, t_2, ..., t_m\}$ be a set of transactions with $\forall k, 1 \leq k \leq m : t_k \subseteq P$ be a vector of transactions over P, and $E(t_k)$ be the set of items in transaction t_k. Each transaction is a set of properties used in the description of the instances of the subset $\mathcal{I}' = \{i_1, i_2, ..., i_m\}$ with $\mathcal{I}' \subseteq I_C$ (e.g., properties used to describe the *The_Godfather* instance are: *director* and *musicComposer*). We consider $\mathcal{CP}$ the completeness of $\mathcal{I}'$ against properties used in the description of each of its instances.

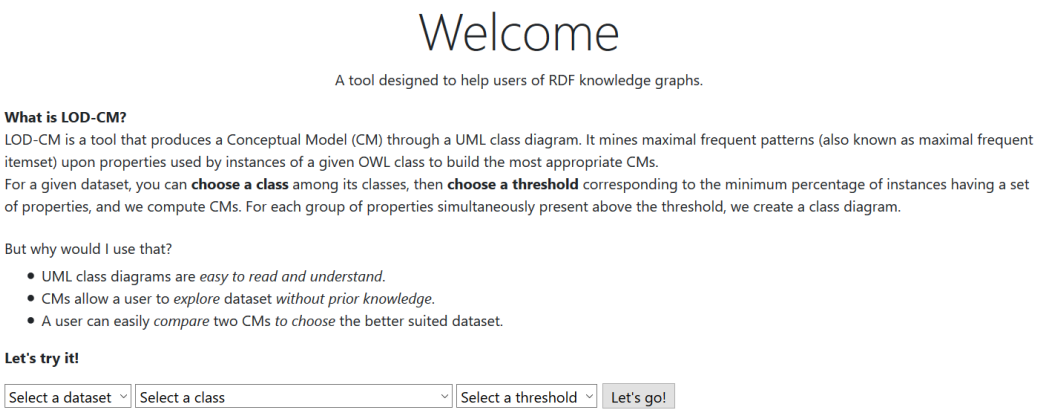

Figure 4.2: LOD-CM main interface.

4.3.4 UI description

In this step, the goal is to generate a conceptual schema enriched with the completeness values calculated in the previous step. The $\mathcal{MFP}$ used to get the completeness values are transformed into a class diagram. Figure 4.2 illustrates the user's interface of our LOD-CM web service. Using the graphical interface[27], the user can choose her own constraints. The web service permits the user to choose the class name in the drop-down list and the user may select the threshold completeness she wants to apply. Currently, our demo supports only DBpedia dataset.

After the user selects the category name and desired completeness and clicks button "*let's go!*", the algorithm runs to find the attributes, relationships and the missed domains/ranges based on the user's constraints.

The structure of the model is constructed regarding the definitions of the patterns properties in the ontology describing the dataset. Figure 4.4 represents a class diagram derived by our approach, from a set of films extracted from DBpedia.

4.3.4.1 A first iteration

In this example, the expectation of the user is a model that guarantees at least 50% of completeness. To generate the model, the first step consists of obtaining the set of

[27] `http://cedric.cnam.fr/lod-cm`

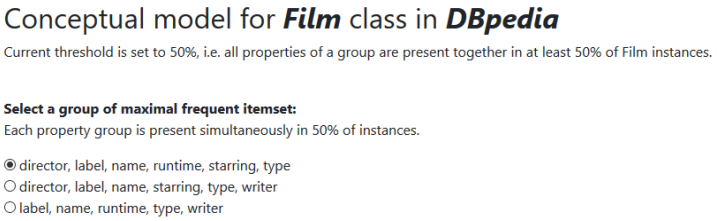

Figure 4.3: List of proposed groups of maximal frequent itemsets.

properties $p \in \bigcup_{j=1}^{n} E(\hat{P}_j)$, and $\hat{P}_j \in \mathcal{MFP}$ that composes the union of all the $\mathcal{MFP}$, mined from the extracted subset, with a minimum support $\xi = 50\%$. For this example, the set of properties are: {*director, label, name, runtime, starring, type*}, {*director, label, name, starring, type, writer*} and {*label, name, runtime, type, writer*} as illustrated in Figure 4.3. OWL distinguishes between two main categories of properties: (i) datatype properties, where the value is a data literal, and (ii) object properties, where the value is an individual (i.e., an other instance with its own type). Each property is considered as an attribute (e.g., name) of the class or a relationship (e.g., director) with another class, depending on the nature of the value. Therefore, according to the nature of the value of each property, it is considered as an attribute of the class or a relationship with another class.

Two types of links will be used during generating of conceptual schemas: inheritance and association links. Inheritance link describes the relation between the class and the superclass, and association link describes the relation between two classes and point to the property. A dotted link was added to illustrate that a class has been inferred to complete the relationship. For this reason, we use a simple heuristic inference step, assuming that the most frequently appearing type in the subject and object position are the property domain and range, similar to the approaches introduced in [Völker and Niepert 2011; Töpper, Knuth, and Sack 2012].

In our example, the names of the classes and the inheritance links between the classes are derived from categories names and organization described in the ontology of the data source DBpedia. We do not derive new names nor new organization of the classes as the conceptual schema should conform to the data used. Indeed, even if the derived conceptual schema is not satisfactory from conceptual modeling principles, it should

faithfully reflect the reality of data while taking into account the user preferences. Finally, the diagram is enriched by the completeness values calculated in the previous step. These values follow appropriate properties.

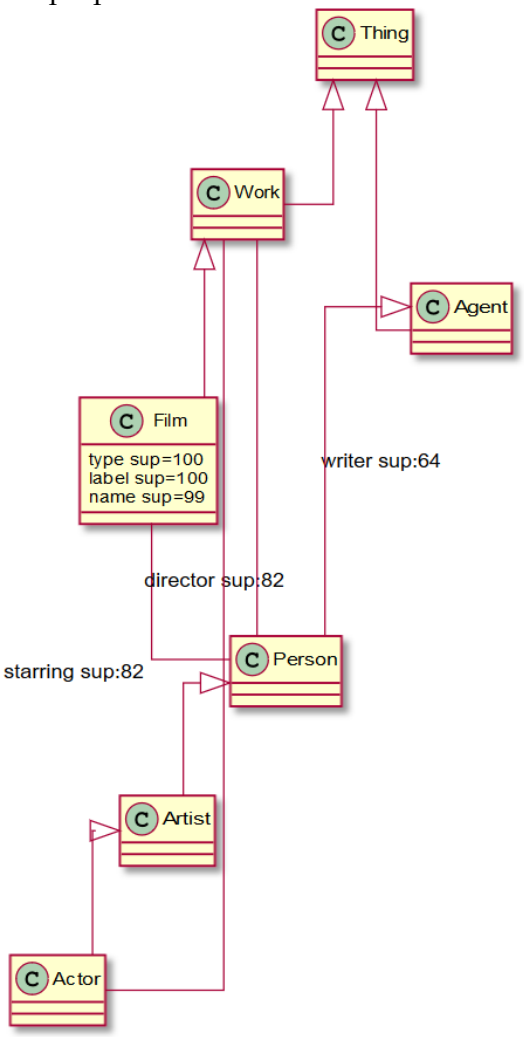

Figure 4.4: The *Film* conceptual schema as a class diagram.

4.3.4.2 A second iteration

A new iteration is triggered when the user chooses to get more details about a part of the model (e.g., the class *Artist*). In this case, a new query is executed on the triple store to extract data corresponding to this part. The previous three, as illustrated in Figure 4.1 steps are then re-executed in order to generate a new model integrating the new desired details.

Figure 4.5 shows an example that details a part of the model from Figure 4.4. In this example, a set of classes, relationships and attributes are added to the category *Artist* with corresponding completeness value. This way of revealing the conceptual schema is similar to a *magnifying glass* that allows the user navigating around a targeted concept, the category *Film* in our example.

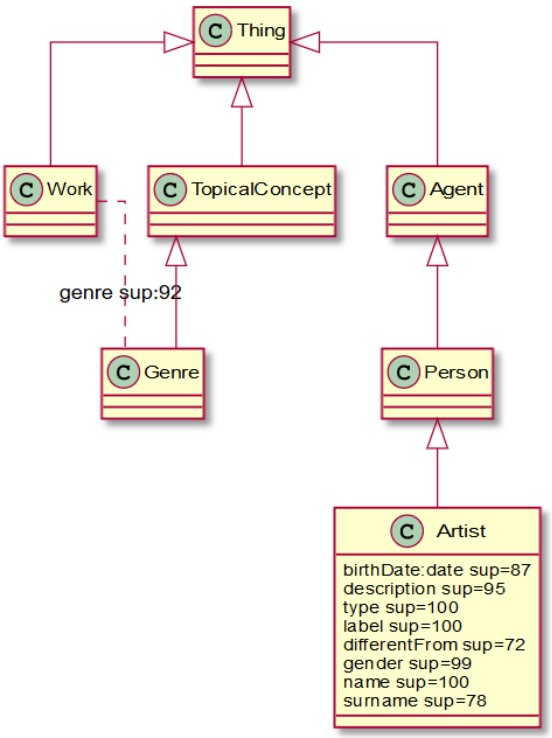

Figure 4.5: The *Artist* diagram class.

The output of our algorithm is a file written in a declarative language. This file includes the chosen category, the attributes, and the relationships tagged by completeness values. We use PlantUML[28] to transfer this generated file into a picture to be illustrated for the user on the Web page.

4.3.5 Use cases

The objective of the Linked Open Data cloud is to enable large-scale data integration, so that we can have a contextually relevant Web and find quick answers to a much wider range of questions. LOD-CM is a web-based completeness demonstrator for linked datasets. It

[28] `http://plantuml.com/`

is used to display data related to the chosen class of a dataset. In this section, we provide a brief summary of two use cases related to schema discovery based on the user's needs. The displayed model could help the user to understand the schema and discover the related properties.

1. **Class diagram to facilitate data browsing**

   LOD-CM aims basically to visualize the discovered schema based on the user's requirements. Suppose a user wants to find the directors and budgets of a list of films. Actually, 82% of films have a director in DBpedia dataset. In addition, only 15% of films have budget value for the same dataset. Only the list of films that have the properties (director and budget) will be displayed (i.e., at most 15% of the films). The outcome model helps the user to present the properties that are related to the chosen class and that are greater than a specified threshold. Besides, it illustrates the relation between the concerned classes such as the classes *Person* and *Film* that are linked by the property *director*. Furthermore, the model illustrates the inheritance relationship such as *Artist* is a subclass of *Person*.

2. **Discovering a subset of MFP**

   As mentioned in Section 4.2.1, our goal is also to find the set of properties that can be used together in the query and does not exceed the selected threshold. For example, for the class *Film* with 60% of completeness, there are four sets of properties that are greater than 60% {{type, name, label, director, writer}, {type, name, label, director, runtime}, {type, name, label, director, starring},{type, name, label, runtime, starring}}. For this reason, our LOD-CM interface enables the user to check the desired set of properties that appear in the returned model. It should be noted that the property which does not achieve the completeness threshold with other selected properties will disappear, such as *starring* and *writer* in our previous example. This feature confirms that the returned results for the query with this set of properties are equal or greater than the desired threshold.

Finally, Table 4.3 shows the number of properties we get at the end of our stage for several classes according to several thresholds. The lower the threshold is, the more properties there are, obviously. Thus, lower thresholds produce more complex conceptual

Table 4.3: DBpedia number of properties by classes and thresholds.

Class/threshold	0.1	0.3	0.5	0.7	0.9
Film	18	12	7	6	3
Settlement	18	14	8	5	4
Organisation	18	4	4	3	3
Scientist	19	16	12	9	5

schemas but with more noise. Hence, this tool can help the user to find the right balance between those two.

4.4 Summary

In this chapter, we have first presented a completeness calculation approach based on data mining. Our proposed approach aims to discover a reference schema from a given data source through properties sets that are the most shared. This discovered schema is used to calculate the completeness of each instance and the whole dataset.

Furthermore, we have presented an approach for revealing conceptual schemas from RDF data sources. The approach is an iterative process that computes a plausible model from the data values. We have shown how to automatically extract schema and represent it as a model from a data source using a user-specified threshold. The inferred model takes into account the data and the user quality expectations. The product is a conceptual schema enriched by both completeness values as a relevancy indicator on the elements of the models, and existence constraints that inform about how often these elements co-exist or co-appear in the real data.

The elements composing the model (classes, relationships and properties) are obtained by applying a mining algorithm with an underlying assumption stating that the more frequent a schema is, the more relevant it is. The user can choose the desired completeness, the parts of the data for which the model will be inferred and the possibility to focus on a different category through an iterative process. Currently, our demo supports only the DBpedia dataset.

We have provided several use cases to demonstrate the usefulness of such a tool.

We believe that it can help in the recognition of a new dataset and its internal structure, therefore, it can help in the adoption of LD datasets. Our analysis revealed some interesting characteristics allowing the characterization of the sources and the behavior of the community that maintains each of the data sources.

Chapter 5

Assessing the Conciseness of Linked Datasets

In this chapter, we address the problem of identifying synonym predicates for linked datasets. We propose an approach to discover equivalent properties to evaluate the conciseness dimension of Linked Data. Section 5.1 shows the drawback of Abedjan and Naumann's approach that was evaluated on a small dataset. Section 5.2 explains our proposed approach that consists of three sequential phases. Finally, Section 5.3 includes a brief summary of the key points of our approach.

5.1 Motivating scenario

Semantic data is usually collected from heterogeneous sources through a variety of tools for different purposes [Lei, Sabou, Lopez, Zhu, Uren, and Motta 2006; Mika 2005]. Unfortunately, this sort of mixing could lead to decreasing the quality of data, so that we have proposed this approach to evaluate the conciseness dimension of Linked Data.

Our research on the conciseness dimension was inspired by Abedjan and Naumann's work "Synonym Analysis for Predicate Expansion" [Abedjan and Naumann 2013]. The authors proposed a data-driven synonym discovery algorithm for a predicate expansion by applying both schema analysis and range content filtering.

Range content filtering aims to represent a transaction as a distinct object with several

predicates. For example, the object *Lyon*[29] city is connected with several predicates such as (*birthPlace*, *deathPlace* and *location*). The authors supposed that synonym predicates share a similar group of object values. For this reason, the proposed approach seeks the frequent patterns of predicates that share a significant number of object values.

In fact, it is not sufficient to synonymously discover the used predicates depending only on range content filtering. For example, the predicates *birthPlace* and *deathPlace* share significant co-occurrences with the same object values but they are definitely used differently. For this reason, the authors added another filter called "schema analysis" in order to overcome this problem. This filter is better in finding suitable synonym predicates. The authors supposed that the synonym predicates should not co-exist for the same instance. According to schema analysis, transactions of distinct subjects with several predicates are represented. By applying negative association rules [Brin, Motwani, and Silverstein 1997], the synonym predicates appear in different transactions. For example, the subject *Michael_Schumacher* does not have two synonym predicates such as *born* and *birthPlace* in the same dataset.

Now, we clarify the drawbacks of Abedjan and Naumann's approach through applying the following example (see Table 5.2). We use a sample of facts from DBpedia dataset to discover the synonym predicates.

Table 5.1: Six configurations of context and target [Abedjan and Naumann 2011].

Conf.	Context	Target
1	Subject	Predicate
2	Subject	Object
3	Predicate	Subject
4	Predicate	Object
5	Object	Subject
6	Object	Predicate

Based on range content filtering (Conf. 6 as illustrated in Table 5.1), all the predicates will be gathered into groups by each distinct object. Thus, in order to retrieve frequent candidates, results could be as in Table 5.3.

As a result, we can see that *nationality* and *sourceCountry* are already in the same

[29] Lyon is a French city

Table 5.2: Facts in SPO structure from DBpedia.

Subject	Predicate	Object
Adam_Hadwin	*type*	*GolfPlayer*
Adam_Hadwin	*birthPlace*	*Moose_Jaw*
Adam_Hadwin	*nationality*	*Canada*
White_River	*sourceCountry*	*Canada*
White_River	*riverMouth*	*Lake_Superior*
White_River	*state*	*Ontario*

Table 5.3: Range content filtering.

Object	Predicate
GolfPlayer	*type*
Moose_Jaw	*birthPlace*
Canada	*nationality*, *sourceCountry*
Lake_Superior	*riverMouth*
Ontario	*state*

transaction. By applying FP-growth algorithm [Han, Pei, Yin, and Mao 2004], or any other itemset mining algorithm, for mining frequent itemsets, *nationality* and *sourceCountry* will be found as a frequent pattern.

The next step is to perform schema analysis (Conf. 1 as illustrated in Table 5.1) by considering subjects as a context to get transactions as illustrated in Table 5.4. By applying negative association rules, Abedjan and Naumann's algorithm shows that there is no co-occurrence between *sourceCountry* and *nationality* predicates. Therefore, it will propose *nationality* and *sourceCountry* as a synonym predicate pair, which is not correct because *nationality* and *sourceCountry* have different intentions.

Table 5.4: Schema analysis.

Subject	Predicate
Adam_Hadwin	*type*, *birthPlace*, *nationality*
White_River	*sourceCountry*, *riverMouth*, *state*

5.2 Discovering synonym predicates

In the next subsections, we explain our proposed approach that consists of three phases. In addition to the statistical study through schema analysis and range content filtering, we basically intend to perform a semantic analysis to understand the meaning of the candidates. Finally, we use learning algorithms to filter the results of the two previous phases.

5.2.1 Phase 1: statistical analysis

As we have already mentioned, our goal is to start with statistical analysis in order to discover potential equivalent predicates. We are interested, in this part, in studying the appearance of each predicate by finding the frequent pattern with negative association rules. This part is basically inspired from Abedjan and Naumann's work [Abedjan and Naumann 2013] which proposed a data-driven synonym discovery algorithm for predicate expansion. Based on mining configuration of contexts and targets [Abedjan and Naumann 2011], the authors applied Conf. 1 and Conf. 6 as illustrated in Table 5.1 that represents schema analysis and range content filtering, respectively.

We extent the method that is explained in Section 5.1 to be suitable not only to generate candidate pairs of synonym predicates, but also to remove those that are actually not by semantic analysis. In the next subsection, we look forward to study the candidates depending on semantics features to decrease the number of predicates by identifying strictly equivalent predicates and eliminating non-equivalent ones.

5.2.2 Phase 2: semantic analysis

Actually, some predicates are not easy to understand, share the same meaning with different identifiers, or have several meanings. For these reasons, calculating string similarity or synonym based measurements on predicate names alone does not suffice. Indeed, the first phase proposes candidate pairs as synonyms but also too many false positive results, especially in the case of large cross-domain datasets. As the previous example illustrated in Section 5.1, the predicates *nationality* and *sourceCountry* could have

the same object (Conf. 6) such as *Canada*. They also never co-occur together for the same subject (Conf. 1). However, *nationality* is a predicate of an instance that its type is *Person* class and *sourceCountry* is a predicate of an instance that its type is *Stream* class. Thus, they should not be considered as synonyms as clarified below.

We add an extension to Abedjan and Naumann's work by studying the meaning of each candidate. Indeed, we examine the semantic representations of the synonym candidates that, under certain conditions, provide us with useful conclusions; for example, a predicate could not be equivalent to another predicate if they have disjoint domains or ranges. Taking the previous example of *nationality* and *sourceCountry* predicates, according to the DBpedia ontology, *Stream* class is a subclass of *Place* class, and *Place* and *Person* classes are completely disjointed. As a consequence, we cannot consider *nationality* and *sourceCountry* to be equivalent predicates.

Thus, in this phase we take into account the semantic part of the predicates. This allows us to detect the incompatibility of the predicates that have opposite features such as symmetric and asymmetric. OWL2 supports declaring two classes to be disjointed. It also supports declaring that a predicate is symmetric, reflexive, transitive, functional, or inverse functional. We take into account these features for each predicate in addition to the *max* cardinality restriction.

We prove the disjointness of predicates, that could not be synonyms, using $\mathcal{SROIQ}$ description logic that models constructors which are available in OWL 2 DL [Horrocks, Kutz, and Sattler 2006]. We depend on studying the meaning of predicates by analyzing their features.

Domain and range disjointness

In the following paragraphs, we give an example about the disjointness of domain and range between two predicates. For a given RDF triple, the property *rdfs:domain* indicates the class that appears as its subject and the property *rdfs:range* indicates the class or data value that appears as its object.

- **Domain of property**

  We use here the property *rdfs:domain* to check whether the domains of the two compared predicates are disjointed or not. If yes, we can state that these predicates cannot be synonyms.

  Theorem 5.1. *Let p_1 & p_2 be two predicates and C_1 & C_2 be two classes, p_1 & p_2 cannot be synonyms if:*

  $$\exists p_1.\top \sqsubseteq C_1 \tag{5.1}$$

  $$\exists p_2.\top \sqsubseteq C_2 \tag{5.2}$$

  $$C_1 \sqcap C_2 \sqsubseteq \bot \tag{5.3}$$

  Proof. Assume $\exists x$, that:

  $$p_1(x, y_1) \tag{5.4}$$

  $$p_2(x, y_2) \tag{5.5}$$

  We assert that:

  $$(5.1) + (5.4) \Rightarrow C_1(x) \tag{5.6}$$

  $$(5.2) + (5.5) \Rightarrow C_2(x) \tag{5.7}$$

  $$(5.6) + (5.7) \Rightarrow C_1 \sqcap C_2 \not\sqsubseteq \bot \tag{5.8}$$

  $$(5.3) + (5.8) \Rightarrow \bot \textit{ absurd} \tag{5.9}$$

  □

  As a result, we conclude that predicates that have disjointed domains are disjointed. In the same manner, we prove the other features discussed previously.

- **Range of property**

  For a given RDF triple, the property *rdfs:range* indicates the class or the data value that appears as its object (the predicate range).

  Theorem 5.2. *Let p_1 & p_2 be two predicates and C_1 & C_2 be two classes, p_1 & p_2 cannot be*

synonyms if:

$$\top \sqsubseteq \forall p_1.C_1 \tag{5.10}$$

$$\top \sqsubseteq \forall p_2.C_2 \tag{5.11}$$

$$C_1 \sqcap C_2 \sqsubseteq \bot \tag{5.12}$$

Proof. Assume $\exists y$*, that:*

$$p_1(x_1, y) \tag{5.13}$$

$$p_2(x_2, y) \tag{5.14}$$

We assert that:

$$(5.10) + (5.13) \Rightarrow C_1(y) \tag{5.15}$$

$$(5.11) + (5.14) \Rightarrow C_2(y) \tag{5.16}$$

$$(5.15) + (5.16) \Rightarrow C_1 \sqcap C_2 \not\sqsubseteq \bot \tag{5.17}$$

$$(5.12) + (5.17) \Rightarrow \bot \textit{ absurd} \tag{5.18}$$

□

Symmetric/asymmetric property

Symmetric property indicates that the relationship between two instances is bi-directional, even if the relationship is only declared in one direction, Sara *sisterOf* Lara as an example.

Asymmetric property means that the object property which is expressed between two instances *a* and *b* cannot be expressed between *b* and *a*, Sara *hasFather* Tim as an example.

Theorem 5.3. *Let* p_1 *&* p_2 *be two predicates,* p_1 *&* p_2 *cannot be synonyms if:*

$$p_1 \textit{ is a SymmetricProperty where } p_1(x, y) \Rightarrow p_1(y, x) \tag{5.19}$$

$$p_2 \textit{ is an AsymmetricProperty where } p_2(x, y) \nRightarrow p_2(y, x) \tag{5.20}$$

Proof. Assume that $p_1 \equiv p_2$*, then:*

$$p_1 \sqsubseteq p_2 \wedge p_2 \sqsubseteq p_1 \tag{5.21}$$

We assert that:

$$(5.21) \Rightarrow p_1(x, y) \Rightarrow p_2(x, y) \tag{5.22}$$

$$(5.19) \Rightarrow p_1(x, y) \Rightarrow p_2(y, x) \tag{5.23}$$

$$(5.19) + (5.21) \Rightarrow p_2(x, y) \Rightarrow p_2(y, x) \tag{5.24}$$

$$(5.20) + (5.23) \Rightarrow \perp \textit{ absurd} \tag{5.25}$$

which is impossible because p_2 *is* $AsymmetricProperty$ □

Transitive property

A property P is transitive, this means if *a P b* and *b P c* then *a P c*. For example, if Adam *hasNeighbor* Saly and Saly *hasNeighbor* Taylor then Adam *hasNeighbor* Taylor. Therefore, if the property is not transitive, that means the relation does not allow to bind the first individual to the last one. For example, Alice *hasFriend* Elsie and Elsie *hasFriend* Bob, so it is not necessarily that Alice *hasFriend* Bob.

Theorem 5.4. *Let* p_1 *&* p_2 *be two predicates,* p_1 *&* p_2 *cannot be synonyms if:*

$$p_1 \textit{ is a } TransitiveProperty \textit{ where } p_1(x, y) \wedge p_1(y, z) \Rightarrow p_1(x, z) \tag{5.26}$$

$$p_2 \textit{ is a Non } TransitiveProperty \textit{ where } p_2(x, y) \wedge p_2(y, z) \nRightarrow p_2(x, z) \tag{5.27}$$

Proof. Assume that $p_1 \equiv p_2$*, then:*

$$p_1 \sqsubseteq p_2 \wedge p_2 \sqsubseteq p_1 \tag{5.28}$$

We assert that $\forall x, y, z$*:*

$$(5.28) \Rightarrow p_1(x, y) \Rightarrow p_2(x, y) \tag{5.29}$$

$$(5.28) \Rightarrow p_1(y, z) \Rightarrow p_2(y, z) \tag{5.30}$$

$$(5.28) \Rightarrow p_1(x, z) \Rightarrow p_2(x, z) \tag{5.31}$$

$$(5.26) + (5.29) + (5.30) + (5.31) \Rightarrow p_2 \textit{ is a TransitiveProperty} \tag{5.32}$$

$$(5.27) + (5.32) \Rightarrow \perp \textit{ absurd} \tag{5.33}$$

□

Functional property

A property P is functional, this means that it can have only one unique range value y (individuals or data values) for each instance x. For example, a person has only one biological mother, Toni *hasBoilogicalMother* Yos. On the contrary, a non-functional property can have, for the same instance x, several range values. For example, a person may have several children. Yos *hasChild* Toni and Yos *hasChild* Tara.

Theorem 5.5. *Let* p_1 *&* p_2 *be two predicates,* p_1 *&* p_2 *cannot be synonyms if:*

$$p_1 \textit{ is a FunctionalProperty} \tag{5.34}$$

$$p_2 \textit{ is a Non FunctionalProperty} \tag{5.35}$$

Proof. Assume that $p_1 \equiv p_2$*, then:*

$$p_1 \sqsubseteq p_2 \wedge p_2 \sqsubseteq p_1 \tag{5.36}$$

We assert that:

$$(5.35) \Rightarrow \exists y_1, y_2 \mid p_2(x, y_1) \wedge p_2(x, y_2) \Rightarrow y_1 \nsim y_2 \tag{5.37}$$

$$(5.36) \Rightarrow p_2(x, y_1) \Rightarrow p_1(x, y_1) \tag{5.38}$$

$$(5.36) \Rightarrow p_2(x, y_2) \Rightarrow p_1(x, y_2) \tag{5.39}$$

$$(5.34) + (5.38) + (5.39) \Rightarrow y_1 \sim y_2 \tag{5.40}$$

$$(5.37) + (5.40) \Rightarrow \perp \textit{ absurd} \tag{5.41}$$

□

Inverse functional property

This property is simply the opposite of the functional property. It means that it can have only one unique domain value x (individuals) for each object y.

Theorem 5.6. *Let p_1 & p_2 be two predicates, p_1 & p_2 cannot be synonyms if:*

$$p_1 \text{ is a } InverseFunctionalProperty \tag{5.42}$$

$$p_2 \text{ is a Non } InverseFunctionalProperty \tag{5.43}$$

Proof. Assume that $p_1 \equiv p_2$, then:

$$p_1 \sqsubseteq p_2 \wedge p_2 \sqsubseteq p_1 \tag{5.44}$$

We assert that $\exists x_1, x_2$:

$$(5.43) \Rightarrow \ p_2(x_1, y) \wedge p_2(x_2, y) \Rightarrow x_1 \nsim x_2 \tag{5.45}$$

$$(5.44) \Rightarrow p_2(x_1, y) \Rightarrow p_1(x_1, y) \tag{5.46}$$

$$(5.44) \Rightarrow p_2(x_2, y) \Rightarrow p_1(x_2, y) \tag{5.47}$$

$$(5.42) + (5.46) + (5.47) \Rightarrow x_1 \sim x_2 \tag{5.48}$$

$$(5.45) + (5.48) \Rightarrow \perp \textit{ absurd} \tag{5.49}$$

□

Cardinality restrictions

OWL2 supports not only Functional property in order to retain just one unique object value, but also goes beyond that by providing the users an authority to specify restrictions to the number of objects values. In the following subsections, we will explain the repercussions of the *max* and *min* cardinality restrictions to find the synonym predicates.

- **Max cardinality restriction**

  The *max* cardinality restriction allows describing a class of individuals that have at

most N value for a given property P. For example, the plane A380-800 has a seating capacity of 868 passengers, so we cannot have more than 868 passengers in the same flight. However, a number less than the value of cardinality constraint is for sure accepted. Formally, we can express this constraint as following: *passengerA380-800 owl:maxCardinality "868"^^ xsd:nonNegativeInteger*. Therefore, we restrict to 868 the instantiation of the *passengerA380-800* for the same individual.

Theorem 5.7. *Let p_1 & p_2 be two predicates and $(a, b) \in \mathbb{N}$, p_1 & p_2 cannot be synonyms if:*

$$p_1 \text{ has maxCardinality} \leqslant a \tag{5.50}$$

$$p_2 \text{ has maxCardinality} \leqslant b \tag{5.51}$$

$$a > b \tag{5.52}$$

Proof. Assume that $p_1 \equiv p_2$, then:

$$p_1 \sqsubseteq p_2 \wedge p_2 \sqsubseteq p_1 \tag{5.53}$$

We assert that $(\forall i, j \in [1, a],\ i \neq j)$:

$$(5.50) \Rightarrow p_1(x, y_1), ..., p_1(x, y_a) \Rightarrow\ y_i \nsim y_j \tag{5.54}$$

$$\begin{aligned}(5.50) + (5.53) \Rightarrow\ & p_1(x, y_1) \Rightarrow p_2(x, y_1)\\ & p_1(x, y_2) \Rightarrow p_2(x, y_2)\\ & \vdots\\ & \vdots\\ & p_1(x, y_b) \Rightarrow p_2(x, y_b)\\ & \vdots\\ & p_1(x, y_a) \Rightarrow p_2(x, y_a)\end{aligned} \tag{5.55}$$

$$(5.55) \Rightarrow |\{(p_2(x, y_1), \ldots, p_2(x, y_a)\}| = a \tag{5.56}$$

$$(5.51) + (5.56) \Rightarrow \perp \textit{ absurd} \tag{5.57}$$

□

- **Min cardinality restriction**

  The *min* cardinality restriction allows describing a class of individuals that have at least N value for a given property P. For example, a father must have at least one child, and there is no limit for the number of children. We do not consider this restriction due to the Open World Assumption [Drummond and Shearer 2006], that states that the lack of knowledge does not imply falsity. In the previous example, the fact that a father has no children is not considered as an inconsistency because it is possible that this father has a child (or children) that is merely unknown.

5.2.3 Phase 3: NLP-based analysis

The returned candidates have shown that some predicates are semantically similar but non-equivalent such as *composer* and *artist*. The Domain and Range types of instances of these predicates are the same and share same features (e.g., asymmetric, non-functional). Thus, statistical and semantic analyses are not sufficient to detect that *composer* and *artist* are non-equivalent predicates. To address such issue, we have used a learning algorithm to map words or phrases from the vocabulary to vectors of numbers called "word embedding". Word embedding uses an efficient technique to vectorize the text by converting strings to numbers, where similar words have similar encodings. We transfer this technique from synonym detection in natural language processing [Pennington, Socher, and Manning 2014; Weeds, Clarke, Reffin, Weir, and Keller 2014] into the field of KGs. Word2Vec that was developed by [Mikolov, Sutskever, Chen, Corrado, and Dean 2013] is one of the most popular techniques to learn word embeddings.

We apply Word2vec tool[30] for learning word embeddings based on a training dataset. The idea behind this tool is to assign a vector space to each unique word in the corpus where any words sharing common contexts are located close to each other in the space. Therefore, if there are two words that used about the same in the context, then these words are probably quite similar in meaning (e.g., *wrong* and *incorrect*) or are at least related (e.g., *France* and *Paris*).

[30] `https://code.google.com/archive/p/word2vec/`

Word2vec uses training algorithms to generate word vectors (embeddings) based on a dataset. As it would be extremely expensive to calculate the similarity of all predicate pairs of the dataset, we run Word2vec only on the resulting pairs from applying the statistical and semantic analyses in Phase 1 and Phase 2. Then, we apply cosine similarity [Salton 1988] for comparing two vectors, which is defined as follows:

$$\cos(\mathbf{t}, \mathbf{e}) = \frac{\mathbf{te}}{||\mathbf{t}||||\mathbf{e}||} = \frac{\sum_{i=1}^{n} \mathbf{t}_i \mathbf{e}_i}{\sqrt{\sum_{i=1}^{n} (\mathbf{t}_i)^2} \sqrt{\sum_{i=1}^{n} (\mathbf{e}_i)^2}}$$

Where similarity score will always be between 0.0 and 1.0.

A high similarity value indicates that two words are closely related and the maximum similarity (1.0) indicates that they are identical. This phase helps to decrease the number of false positive results, through including the candidates that have a significant similarity score and excluding them otherwise.

Finally, we have followed the metric identified by [Mendes, Mühleisen, and Bizer 2012b] to assess the conciseness dimension at schema level. The conciseness at schema level is measured as follows:

$$\frac{number\ of\ unique\ predicates\ of\ a\ dataset}{number\ of\ predicates\ in\ a\ target\ schema}$$

.

5.3 Summary

In this work, we have proposed a new approach to evaluate the conciseness of linked datasets by discovering synonym predicates. This approach consists of three phases: (1) performing a statistical analysis to obtain an initial set of synonyms predicates, (2) performing a semantic analysis of obtained set by exploring OWL2 features (*functional, transitive, cardinality, etc.*), and (3) finally, performing a similarity-based comparison between contextual vectors representing each candidate predicate. The main objective of the last two phases is to reduce the false positive candidates generated in the first phase.

We believe that our approach helps to enhance the conciseness dimension of a dataset

through eliminating one of the equivalent predicate pairs. This could be done by replacing the predicate that is used less in the dataset to describe instances, or by enabling the user to choose the preferred one.

Chapter 6

Experimental Evaluation

In the previous chapters, we have described new approaches to assess completeness and conciseness dimensions of Linked Data quality. In this chapter, we present experimental evaluations carried out in the context of this work. We have performed different experiments and classified them into two main parts.

In the first part of this chapter, we initially conduct a set of experiments upon three relatively timely spaced versions of DBpedia to study the evolution of completeness of this dataset. Then, Several experiments have been conducted to evaluate the robustness of the completeness measure regarding the number of instances and the user-specified threshold at several different parameters.

In the second part, we present two experiments performed on real-world datasets to evaluate our proposed approach that aims to assess conciseness of Linked Data. For evaluating pair accuracy, we use the standard precision, recall and F-measure on DBpedia and YAGO datasets.

6.1 Completeness dimension

The experiments were performed on the well-known real-world datasets, DBpedia[31] and YAGO[32], publicly available on the Linked Open Data (LOD). DBpedia is a large

[31] https://wiki.dbpedia.org

[32] https://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/yago/

Table 6.1: Number of resources/class.

	Film	Organisation	PopulatedPlace	Scientist
v3.6(2013)	53,619	147,889	340,443	9,726
v2015-04	90,060	187,731	455,398	20,301
v2016-10	106,613	275,077	505,557	23,373

knowledge base composed of structured information extracted collaboratively from Wikipedia. It describes currently about 6 million entities. The second dataset, YAGO, is a semantic knowledge base derived from Wikipedia, WordNet and GeoNames. Currently, YAGO has more than 10 million entities and contains more than 120 million facts about these entities.

6.1.1 Experimental setup

We present a set of experiments aiming (i) to analyze the evolution of completeness quality values over several versions of DBpedia, and (ii) to investigate subsets of instances from different categories (i.e., classes) and provide the discovered schema that can be generated from data values under completeness constrains. This experiment was performed on DBpedia and YAGO datasets. For evaluating the completeness of different versions of DBpedia, we have chosen three relatively timely spaced versions. The first one (v3.6) was generated in March/April 2013, the second one (v2015-04) in February/March 2015, and the third one (v2016-10) in October 2016. For each dataset, we have chosen various classes of different natures. We have studied the completeness of resources that have the following ones: $C = \{dbo{:}Film, dbo{:}Organisation, dbo{:}Scientist, dbo{:}PopulatedPlace\}$ as classes. For the properties used in the resources descriptions, we have chosen English datasets "mapping-based properties", "instance types" and "labels". The number of triples (statements) of each class is given in Table 6.1.

In the first experiment, we constructed the set of corresponding transactions $\mathcal{T}$. A transaction vector is constituted of sequences of properties deduced from instances belonging to a single class (e.g., the set of *Film* in DBpedia). The set of transactions is then used as an input to generate frequent patterns and compute the completeness. All experiments have been performed on a Dell XPS 27 with an Intel Core $i7-4770S$ processor

Table 6.2: Statistics on the size of each category in both DBpedia and YAGO.

DBpedia	Film	Organisation	PopulatedPlace	Scientist
	106,613	275,077	505,557	23,373
YAGO	movie	organization	urban_area	scientist
	309,630	628,786	90,869	63,384

and 16 GB of *DDR*3 RAM. The execution time of each experiment is about two minutes. The evaluation methodology consists of calculating, regarding the same inferred schema (for us the same $\mathcal{MFP}$), the completeness of different versions. We have chosen for our experiments, as a reference schema, the one inferred from the older version. Thus, we can observe the completeness evolution over versions.

In the second experiment, we have generated for each transaction file 10 random samples where each one containing $10,000$ transactions, and computed the completeness values for each transaction at different support values. The generation of random samples serves to test the robustness of our approach which is evaluated on DBpedia and YAGO datasets. Table 6.2 shows statistics about the number of instances in each category.

6.1.2 Completeness evolution over several versions of DBpedia

We have performed a set of experiments where completeness calculation was performed only on equivalent resources belonging to the three versions (intersection of triples subjects). Therefore, we focus on how the completeness is impacted by updated (not inserted) data. Figure 6.1 shows the completeness results obtained from the chosen classes of DBpedia v3.6, v2015-04 and v2016-10 at different minimum supports ξ. The completeness is calculated for the three versions regarding the same $\mathcal{MFP}$ inferred from the v3.6 version.

The results show that the diagrams of the classes *Film* and *PopulatedPlace* are roughly the same. Thus, we can conclude that for these classes, either the resources were almost not updated, or the updates did not alter the completeness. For the class *Scientist* and *Organisation*, the results show that completeness seems better in 2013 version. This may be due to the addition of new widespread properties across those classes but not added for enough individuals, or maybe some individuals have lost some important properties while DBpedia evolving.

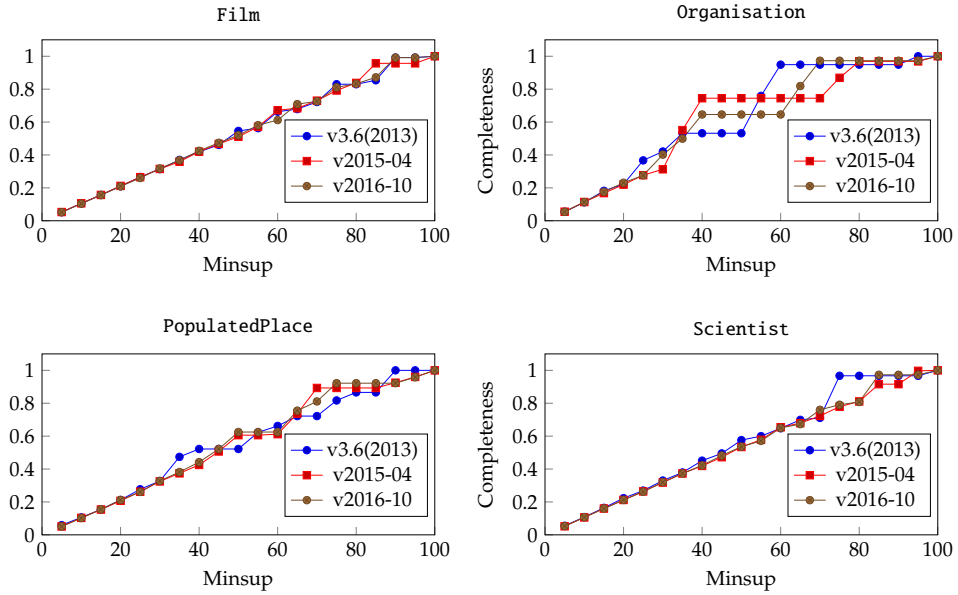

Figure 6.1: Completeness of equivalent resources from DBpedia v3.6, v2015-04 and v2016-10.

Then we have reproduced the same experiment but this time by taking all instances of a given class, not only instances belonging to the three datasets. The results of this new experiment are given in Figure 6.2.

We observe for *Scientist* and *Organisation* that completeness values are almost the same, except completeness for ξ between 30% and 50%. As completeness was better in the first experiment in 2013 for those two classes, we may think that added individuals in 2015 and 2016 versions have raised the completeness. However, for *Film* and especially for *PopulatedPlace*, there is a clear difference in completeness values and surprisingly 2013 and 2016 versions are very close. For the class *Film* the completeness values became less in v2015-04 compared to the v3.6 and the v2016-10. Either new *Film* instances lack important properties when added, or updated *Film* instances have lost some of their important properties. Since in the first series of experiments, we see there are no differences for common instances for the three DBpedia versions, we can conclude that the first justification must be the right one. For the class *PopulatedPlace*, as updated instances have not changed the completeness, the result we can observe may be caused by instances added in 2015 that were more complete than those added in 2016, because the completeness has

dropped down in 2016 to the level in 2013 after a rise in 2015.

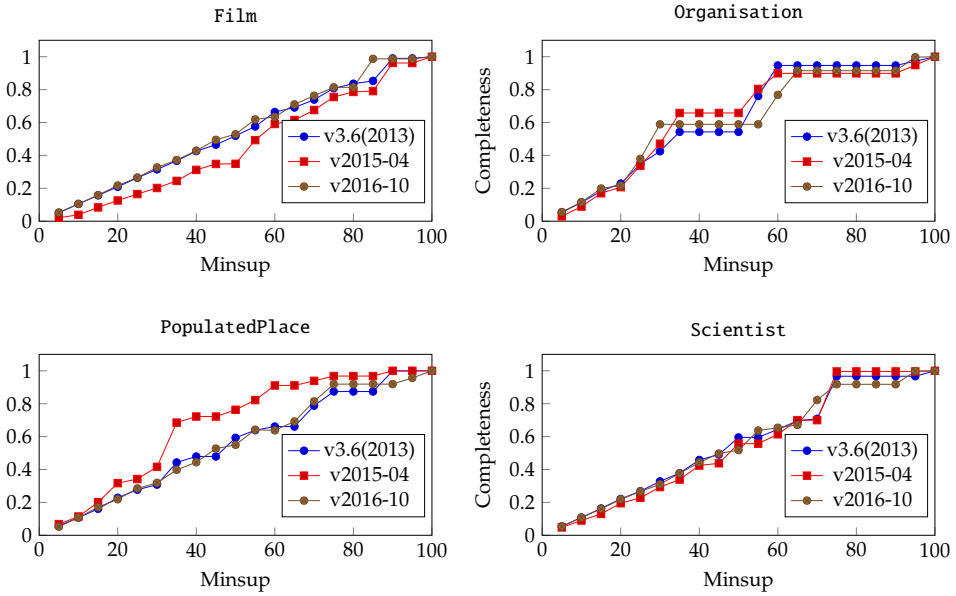

Figure 6.2: Completeness of DBpedia v3.6, v2015-04 and v2016-10.

6.1.3 Generate schema from data values under completeness constrains

In this experiment, we aim to evaluate the completeness measure robustness regarding the number of instances and the user-specified threshold.

In the first step of our experiments, queries on local dump of DBpedia and YAGO datasets are performed to extract data concerning each category. Then, the set of corresponding transactions $\mathcal{T}$ is constructed. A transaction vector is constituted of sequences of properties inferred from instances belonging to a single category (e.g., the set of *Film* in DBpedia). The set of transactions is then used as an input to generate the frequent patterns of properties.

The results for the two datasets DBpedia and YAGO are detailed in Tables 6.3 and 6.4, respectively. Remember that our approach derives a set of properties P, where P is the union of the set of $\mathcal{MFP}$, computed from the most frequent patterns having the same minimal support. We try to extract properties that co-occur to be sure that the mined schema has a certain degree of coherence. Concerning the robustness, the standard deviation of each population for the different categories (for both DBpedia and YAGO) is

Table 6.3: The completeness values and the number of properties for DBpedia categories at different minimum supports ξ.

	Film			Organisation			PopulatedPlace			Scientist		
ξ	$\mathcal{CP}$	$\|P\|$	$\|\mathcal{MFP}\|$	$\mathcal{CP}$	$\|P\|$	$\|\mathcal{MFP}\|$	$\mathcal{CP}$	$\|P\|$	$\|\mathcal{MFP}\|$	$\mathcal{CP}$	$\|P\|$	$\|\mathcal{MFP}\|$
5%	0.053	18	141	0.056	40	54	0.052	24	95	0.054	15	85
						⋮						
40%	0.429	10	8	0.589	4	1	0.444	8	2	0.443	8	5
45%	0.495	9	6	0.589	4	1	0.527	8	2	0.497	7	6
50%	0.529	7	5	0.589	4	1	0.550	8	6	0.518	7	4
						⋮						
95%	0.987	3	1	0.997	2	1	0.957	4	2	0.998	2	1
100%	1.0	1	1	1.0	1	1	1.0	1	1	1.0	1	1

Table 6.4: The completeness values and the number of properties for YAGO categories at different minimum supports ξ.

	movie			organization			urban_area			scientist		
ξ	$\mathcal{CP}$	$\|P\|$	$\|\mathcal{MFP}\|$	$\mathcal{CP}$	$\|P\|$	$\|\mathcal{MFP}\|$	$\mathcal{CP}$	$\|P\|$	$\|\mathcal{MFP}\|$	$\mathcal{CP}$	$\|P\|$	$\|\mathcal{MFP}\|$
5%	0.051	8	1	0.067	7	6	0.057	11	6	0.054	11	19
						⋮						
40%	0.482	7	1	0.470	5	1	0.513	6	2	0.432	9	3
45%	0.482	7	1	0.470	5	1	0.525	6	2	0.474	9	3
50%	0.639	6	2	0.775	4	1	0.592	6	1	0.552	8	5
						⋮						
95%	0.980	5	1	1.0	2	1	1.0	4	1	0.955	5	1
100%	1.0	2	1	1.0	2	1	1.0	2	1	1.0	3	1

close to 0. Therefore, we conclude that the completeness measure used in the our proposed approach is robust.

In both tables, we analyze the variation of the number of properties $|P|$ for a given category regarding the number of frequent patterns $|\mathcal{MFP}|$ which has been computed. From the two tables, we notice that for high values of minimal support that allows guaranteeing high completeness values (100% and 95%), the schema size is insignificantly low regardless of the nature of data (category and source of data). This is due to the absence of the schema during data publishing. For certain categories, the situation is even worse. For example *Organisation* in DBpedia are described by 4 properties in only 40% of cases. We also notice that there is a relative stability of the schema size around 40%

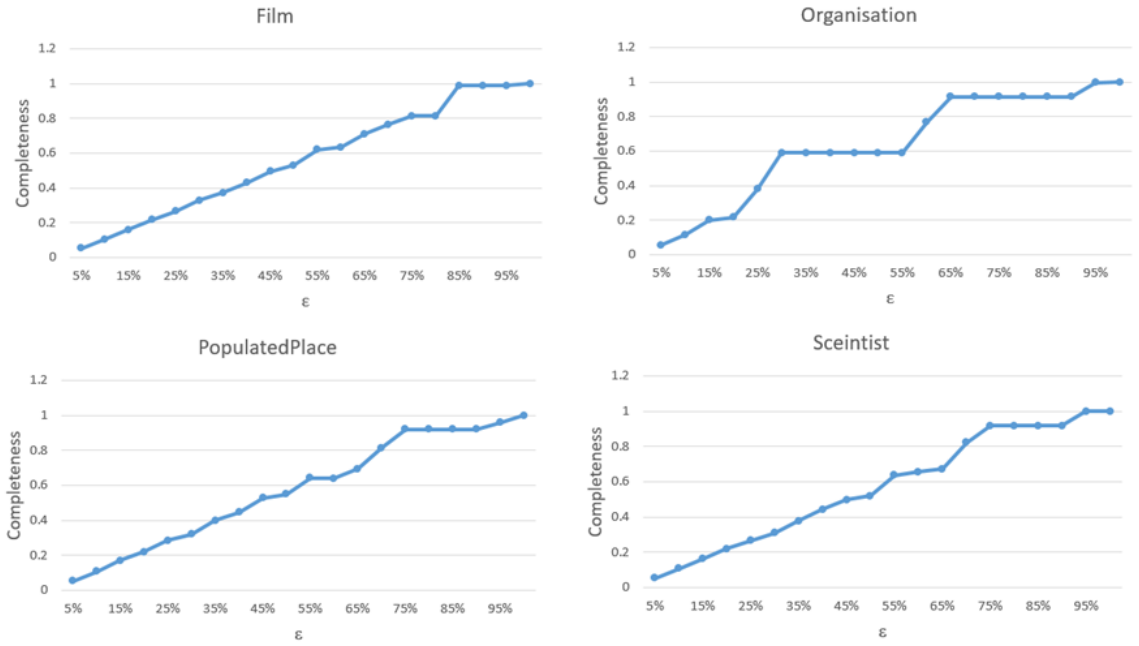

Figure 6.3: Completeness values of DBpedia categories at different minimum supports ξ.

and 50%. We have analyzed the related schemes and found a reasonable variation of properties.

The two tables show that in a user defined content environments, data is not equally defined. Some categories attract more publication efforts than others. The *Film* and *Scientist* categories have richer descriptions, in terms of the number of properties, compared to *Organisation* and this phenomenon is enhanced in DBpedia too. The number of frequent patterns is also an interesting measurement as it helps to analyze the differences between the two analyzed data sources. If we analyze this value for each of the two data sources, we notice a notably different behavior for this column. The number of frequent patterns at each level of minimal support is relatively stable for YAGO compared to DBpedia. The results of DBpedia categories are illustrated in Figure 6.3 and Figure 6.4 while Figure 6.5 and Figure 6.6 clarify the results of YAGO categories.

In DBpedia, we notice that there is a variety of representations (frequent patterns) describing the data, corresponding to various specific visions on the described category. In YAGO, the descriptions are less disparate. This could be explained by the way these two datasets have been constructed. DBpedia is populated from Wikipedia and thus reflects

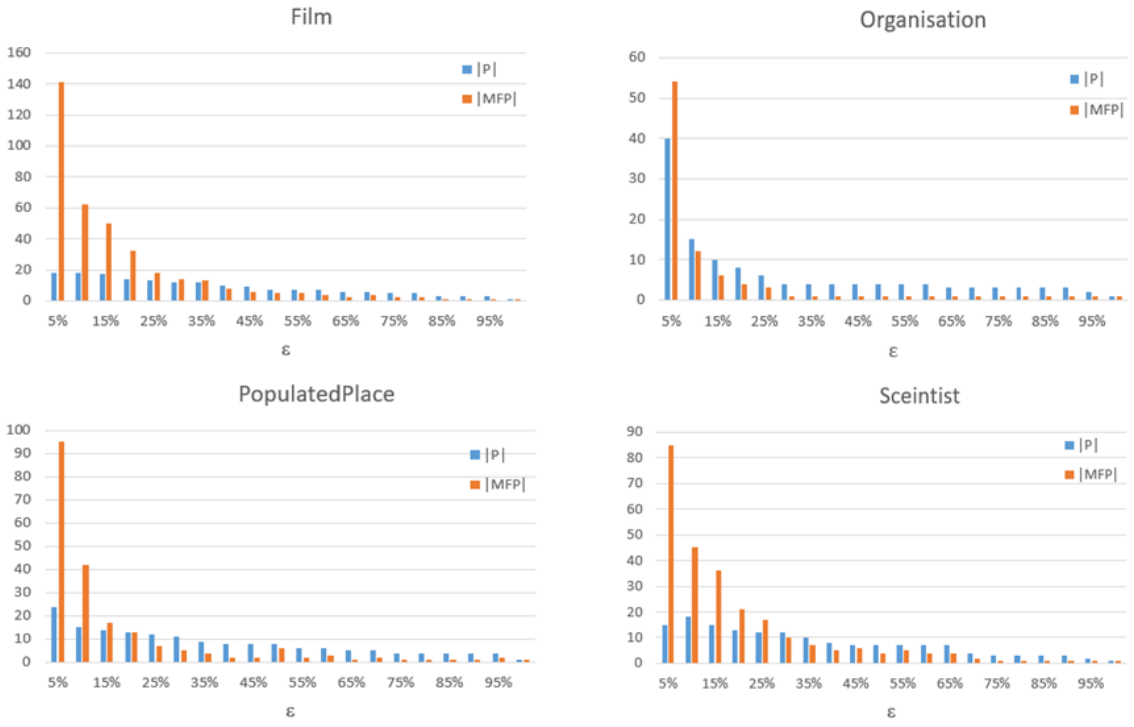

Figure 6.4: The number of properties $|P|$ and the number of frequent patterns $|\mathcal{MFP}|$ of DBpedia categories at different minimum supports ξ.

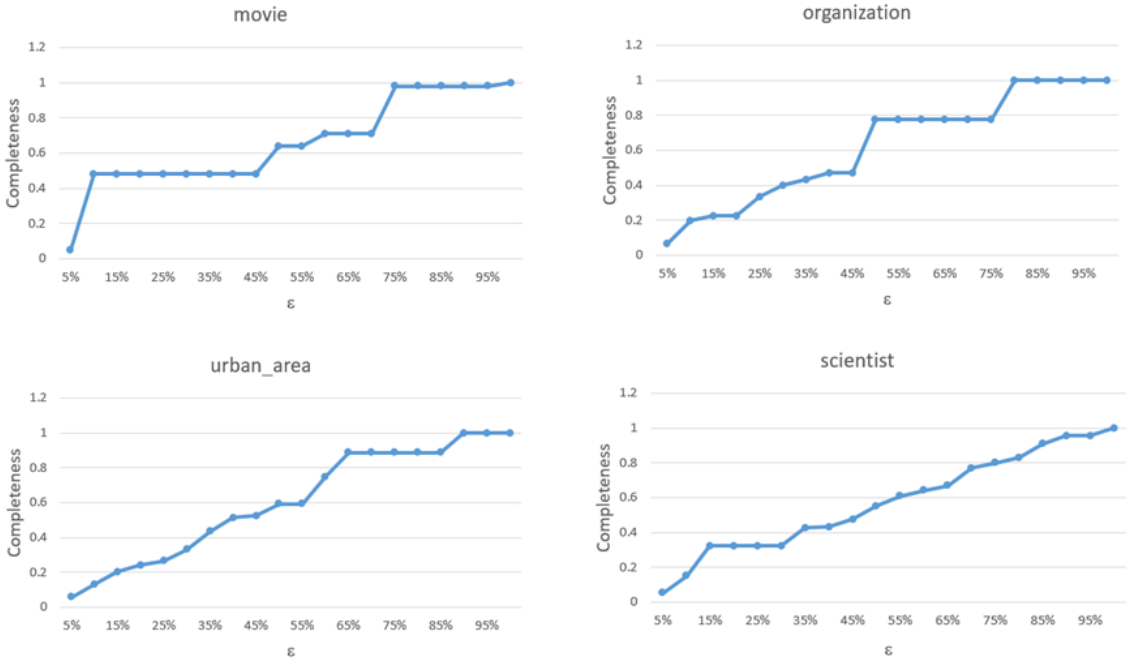

Figure 6.5: Completeness values of YAGO categories at different minimum supports ξ.

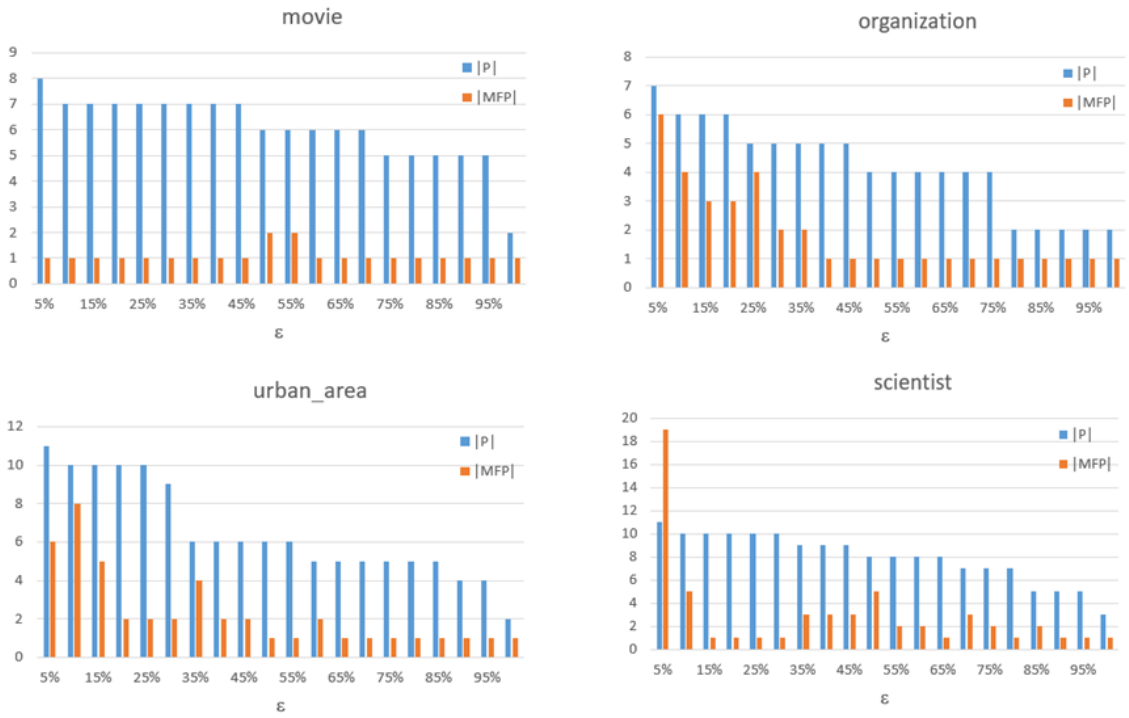

Figure 6.6: The number of properties $|P|$ and the number of frequent patterns $|\mathcal{MFP}|$ of YAGO categories at different minimum supports ξ.

enabling the user to publish data freely. YAGO, however, is a result of a unifying process of Wikipedia and WordNet with an effort on coverage and accuracy of the integrated data [Suchanek, Kasneci, and Weikum 2007a]. This verification effort led consequently to more cohesive descriptions of categories captured by our approach through a relatively low number of frequent patterns for a given level of desired completeness.

Finally, the experiments show the ability of our approach to derive models by taking into account user-specified completeness constraints. It also shows the ability to analyze data from both its semantic content by providing a comprehensive conceptual model, as well as from its structural configuration through adding a set of detailed measurement.

6.2 Conciseness dimension

In this section, we present two experiments performed on DBpedia and YAGO datasets in order to evaluate our approach. The metrics we have used for evaluating pair accuracy

are the standard precision, recall and F-measure (harmonic mean of precision and recall) that are calculated as follows:

- True Positive (TP): the number of discovered predicates by our approach that are actually equivalent predicates.
- False Positive (FP): the number of discovered predicated by our approach that are actually non-equivalent predicates.
- False Negative (FN): the number of equivalent predicates that are not discovered by our approach.
- $Precision = \frac{TP}{TP + FP}$
- $Recall = \frac{TP}{TP + FN}$
- $F - measure = \frac{2 \times Precision \times Recall}{Precision + Recall}$

As it is obvious that eliminating equivalent predicates will improve the conciseness of the dataset, we only focus on evaluating the accuracy of our approach to identify synonym predicates.

6.2.1 Experimental setup

As we have already pointed out that DBpedia project extracts structured knowledge from Wikipedia, it should represent a good challenge for our approach to find equivalent predicates. There is a great chance that some of data entered by different contributors is equivalent. Unfortunately, as long as many datasets are available in LOD, DBpedia suffers a lack of expressive OWL2 features such as *functional* properties, *transitive* properties, etc. Therefore, in this case it is difficult for our approach to perform a semantic analysis. As illustrated in Table 6.5, only 30 functional properties that represent 1% have been defined. Furthermore, DBpedia neither use *max* cardinality nor *transitive* or *symmetric* properties. In addition, we have noted that 16.3% of predicates are represented without domains and 10.2% without ranges.

Table 6.5: Features predicates of DBpedia dataset (v10-2016).

Feature	existence
Domain	83.7%
Range	89.8%
Functional properties	1%
Transitive properties	0%
Symmetric properties	0%
Max cardinality	0%

To address the lack of domains and ranges, we infer, based on the approach proposed in [Töpper, Knuth, and Sack 2012], the missed predicate domains (and/or ranges). By studying the instances that occur with each predicate which has no *rdfs:domain* value (and/or *rdfs:range* value), we have found that some of these instances may belong to different classes. In this case, only the class having a number of instances greater than a selected threshold will be defined as a domain (or range) of the predicate. In case the number of instances is smaller than the threshold, *owl:Thing* will be selected as domain (or range) value. Besides, we have applied [Töpper, Knuth, and Sack 2012; Fleischhacker, Völker, and Stuckenschmidt 2012] to enrich DBpedia ontology with the other OWL2 properties (e.g., *functional*, *transitive*, etc.).

On the other hand, due to the fact that some predicates share the same features such as *artist*, *composer* and *writer*, we have decided to use Word2vec tool, as we explained in Section 5.2.3. Our goal is to convert each predicate to a vector based on its context, and then calculate the similarity between predicate pairs. For this reason, we need a training dataset that contains all the predicate candidates. To create this dataset, we have chosen to merge data from Large Movie Review Dataset that contains 50,000 reviews from IMDb[33], and Polarity Dataset v2.0[34] that have 2,000 movie reviews. This choice is motivated by the fact that both datasets include the majority of the candidates according to our experiments. For missed predicates or when the frequency of the predicate is very low, we have generated paragraphs from the DBpedia dataset itself and we have added them to the training dataset. Actually, we have taken into account the suggestion of Carlson et al. [Carlson, Betteridge, Kisiel, Settles, Hruschka, and Mitchell 2010] which proposes

[33] http://ai.stanford.edu/~amaas/data/sentiment/
[34] http://www.cs.cornell.edu/people/pabo/movie-review-data/

that 10-15 examples are typically sufficient to learn the meaning of the predicate from Natural Language texts. We have generated these paragraphs through the text that exists in *rdfs:comment* of both the subject and the object of the triple, and *rdfs:label* of the predicate.

Example 6.2.1. *The English phrases about dbo:residence predicate is generated using the following Listing 6.1.*

Listing 6.1 – Sample query returning English phrases about *dbo:residence* predicate

```
SELECT DISTINCT ?s1 ?p1 ?o1 WHERE {
    ?s dbo:residence ?o .
    ?s rdfs:comment ?s1 .
    ?o rdfs:comment ?o1 .
    dbo:residence rdfs:label ?p1 .
    FILTER (lang(?o1) = 'en')
    FILTER (lang(?s1) = 'en')
    FILTER (lang(?p1) = 'en') }
```

As a result, we have got a paragraph that contains the missed candidate to be added to our training dataset. For example, the previous query generates the following paragraph (an excerpt): "*Lena Headey is an English actress...residence London is the capital of England and the United Kingdom...*". Adding this paragraph to the dataset will help the training process to generate a vector for the predicate *residence*, and according to the context *residence* connects between *Person* and *Place*.

6.2.2 First experiment

The objective of this first experiment is to show the improvement in detecting synonyms brought by the semantic analysis and NLP phases. As a reminder, the main goal of our approach is to evaluate the statistical analysis of synonyms predicates, which is the core of Abedjan and Naumann's approach. To evaluate our results, we have used a gold standard of DBpedia synonyms predicates generated by [Zhang, Gentile, Blomqvist, Augenstein, and Ciravegna 2017]. This gold standard contains 473 true positive pairs of 10 classes in DBpedia. Compared to [Zhang, Gentile, Blomqvist, Augenstein, and Ciravegna 2017], our approach gives, for a support threshold equals 0.01%, a slightly better F-measure (0.76 for our approach and 0.75 for [Zhang, Gentile, Blomqvist, Augenstein, and Ciravegna 2017]).

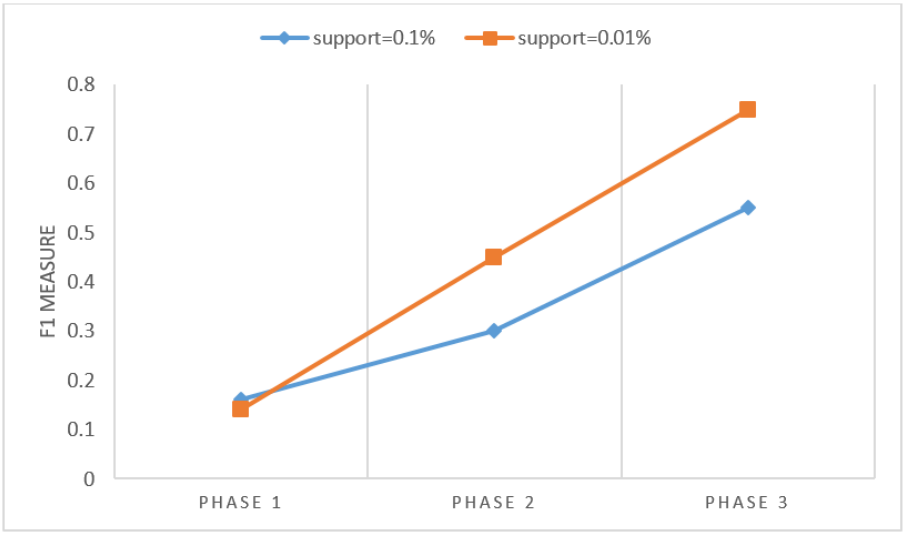

Figure 6.7: F1-measure values at each phase based on support threshold.

But as we have explained just before, the objective here is to show the interest of using semantic analysis and NLP. Thus, our approach would be rather complementary with [Zhang, Gentile, Blomqvist, Augenstein, and Ciravegna 2017] instead of being a direct concurrent.

Figure 6.7 and Figure 6.8 illustrate the number of equivalent predicates pairs and the F-measure at each phase. To show the improvement that Phase 2 and 3 make, we have chosen a low support threshold value for the statistical analysis to obtain a large number of candidates. This will increase the number of true and false positive results at Phase 1, and will show how Phase 2 and Phase 3 decrease false positive results to enhance the precision value. Besides, at Phase 3, our approach filters the candidates regarding their similarity scores that should be less than a user-specific threshold. In this experiment, we have set the value on 50%.

Figure 6.8 shows that with a support threshold equals 0.01%, we obtain after applying the statistical analysis (Phase 1) 4197 candidate pairs. Then, by performing a semantic analysis (Phase 2), the number decreases to 2006 which represents the elimination of 52.2% of false positive results. For example, the predicates *owner* and *employer* have been proposed as equivalent predicates by the statistical analysis phase; because on the one hand, they share a significant number of object values in their range *dbo:Organisation*, and

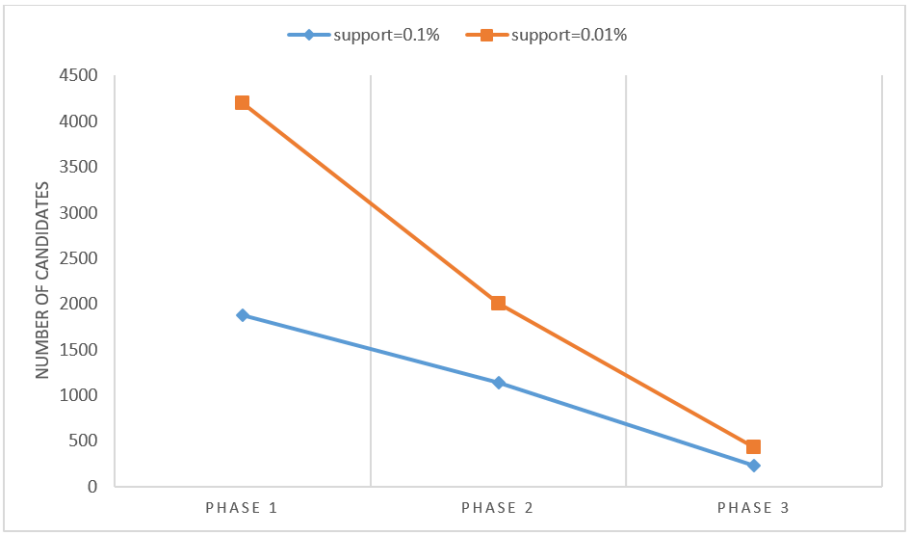

Figure 6.8: Number of candidate pair at each phase.

on the other hand, they rarely co-occur for the same instance. By applying the semantic analysis, this pair of predicates will be excluded due to a domain disjointness. Indeed, the domain of *employer* is *dbo:Person* and the domain of *owner* is *dbo:Place*, and *dbo:Person* and *dbo:Agent*, that is a super class of *dbo:Person*, are disjoint. Thus, as explained in Section 5.2.2, *owner* and *employer* cannot be synonyms. Finally, by performing an NLP-based analysis (Phase 3), the number of candidate pairs decreases to 429, which represents the elimination of 78.6% of false positive results. This phase was able to filter the predicates that share the same semantic features but are non-equivalents such as *author* and *composer*.

Our approach works well to achieve our objective through decreasing the number of false positive results. The experiment shows that we can increase the precision value without affecting the recall.

6.2.3 Second experiment

The gold standard of synonyms predicates of the first experiment is the only one that we have found in LOD. Thus, to perform more tests on our approach, we have created a new gold standard from mappings established between the predicates of different datasets. In fact, the mechanism consists of combining two or more datasets that have similar equivalence predicates. These predicates will therefore be the gold standard of the new

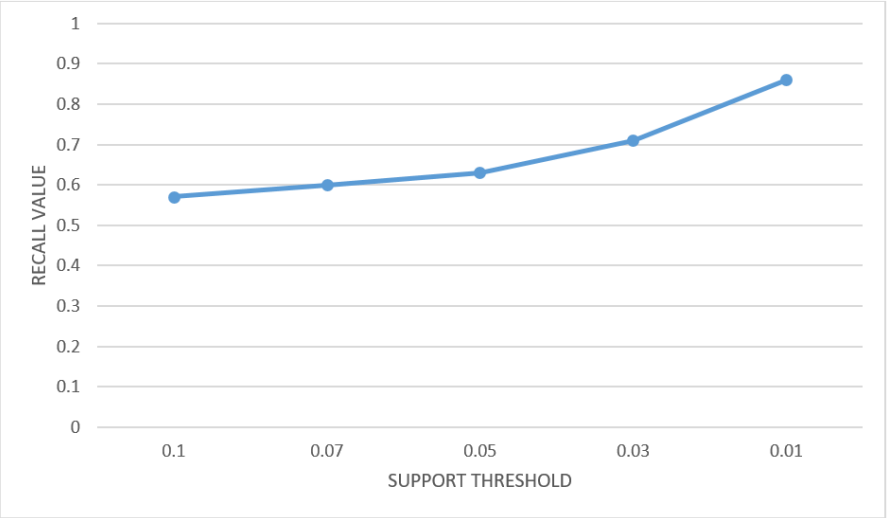

Figure 6.9: Recall value based on support threshold values.

dataset. For this experiment, we have chosen to merge DBpedia and YAGO that share a significant number of equivalent predicates. PARIS tool [Suchanek, Abiteboul, and Senellart 2011] proposes a gold standard containing 173 mappings between the predicates of these two datasets. However, this gold standard is incomplete and, thus, can only serve to see if our approach is able to find all equivalent predicates. Due to the huge number of instances that uses the predicates of this gold standard, we have demanded from three experts to manually extract 35 mappings (equivalent predicates). Then, for each dataset, we have chosen a couple of categories from different natures to cover all of the 35 predicates. For DBpedia, we have selected instances that have the following categories: C = {*dbo:Person*,*dbo:Film*, *dbo:Organisation*, *dbo:PopulatedPlace*}, and for YAGO those that are semantically close to those of DBpedia: C = {*yago:wordnet_person_105217688*, *yago:wordnet_movie_106613686*, *yago:wordnet_organization_1008008335*, *yago:wordnet_urban_area_108675967*}. Figure 6.9 shows the obtained recall at different support thresholds of Phase 1. The maximum value is obtained when the support threshold equals 0.01%. This is logical since a great number of candidates is generated. However, certainly a lot of false positives will also be generated. The reason why we do not find all the synonyms (e.g., *isbn* and *hasISBN*) is due to the fact that some predicate pairs share insufficient number of objects. The interesting result of this experiment is that our approach finds a good number of synonyms (recall at roughly 60%) even if the support

threshold is 0.1% which is relatively high in comparison to 0.01%.

6.3 Summary

To summarize, in the first part of this chapter, we have conducted a set of experiments upon three relatively timely spaced versions of DBpedia. These experiments have revealed that datasets completeness could vary due to changes made to existing data or newly added data. We have also noticed that this evolution often does not benefit from the initial data cleaning as the set of properties continue evolving over time. Our approach could be helpful for data source providers to improve, or at least to keep, a certain completeness of their datasets over different versions. It could be particularly useful for datasets constructed collaboratively, by imposing to contributors some rules when they update or add new resources.

Next, the approach has been evaluated on two datasets, DBpedia and YAGO. Several experiments have been conducted to evaluate the completeness measure robustness regarding the number of instances and the user-specified threshold. We have also analyzed how the approach behaves with several varied parameters. For example, we have studied the impact of the threshold on the number of properties that compose the inferred schema and the configuration of the frequent properties patterns.

In the second part of this chapter, we have evaluated our proposed approach on DBpedia and YAGO datasets. The experiment results show that our approach is highly promising, as it allows eliminating about 78.6% of false positives compared to Abedjan and Naumann's approach [Abedjan and Naumann 2013]. They also show good results in terms of F-measure. However, it has failed to discover some synonym predicates that are rarely used. The two tables show that in a user defined content environments, data is not equally defined. Some categories attract more publication efforts than others. The *Film* and *Scientist* categories have richer descriptions, in terms of the number of properties, compared to *Organisation* and this phenomenon is enhanced in DBpedia too.

Chapter 7

Conclusion and Perspectives

This chapter rearranges our objectives, summarizes what we have done, and highlights the implementations. At the end, we will give some perspective points for the future work of this research.

7.1 Thesis summary

Because a large amount of information is being daily generated and information needs to be of a high quality to be useful, the need for quality assessment of this data on the Web is more urgent than ever before. This thesis has addressed two dimensions of Linked Data quality, it is about assessing completeness and conciseness of Linked Data. Since these dimensions are related to other dimensions such as accuracy, we proposed two new approaches to assess both dimensions. In the following parts, we present a summary of the main contributions of this thesis.

SLR on Linked Data completeness

In this part, we have surveyed the research topic on completeness of LD, analyzed 52 studies, and classified seven types of completeness. We have also provided definitions for each type, identified the different kinds of problems that they address, provided approaches and metrics for assessment, and analyzed the tools available for assessment of LD completeness. In this SLR, we have addressed the research question: *How can we assess the completeness of Linked Data, which includes different types of completeness considering several*

approaches?. There are a number of different reasons why we have done this SLR:

- to summarize existing approaches concerning Linked Data completeness.
- to identify problems, approaches, metrics and tools for assessing LD completeness.
- to realize gaps in existing studies regarding LD completeness, in order to help the researchers find the field where they should work in.
- to serve as a starting document for future researchers interested in this topic.

The proposal of a completeness calculation approach

Data completeness has two facets. The first one analyzes whether all data is available, such completeness is known as structural completeness [Ballou and Pazer 2003] and the second facet requires a reference benchmark or a gold standard dataset as completeness reference [Zaveri, Rula, Maurino, Pietrobon, Lehmann, and Auer 2016]. To assess the completeness of LOD dataset, we propose a mining-based approach that includes two steps. The first step aims to find the properties patterns that are most shared by the subset of instances extracted from the triple store related to the same category. This set, called *transaction*, will be then used to calculate a completeness value regarding these patterns. The second step carries out for each transaction a comparison between its corresponding properties and each pattern of the $\mathcal{MFP}$ set regarding the presence or the absence of the pattern. An average is, therefore, calculated to obtain the completeness of each transaction $t \in \mathcal{T}$ and, hence, the completeness of the whole dataset. We have evaluated the proposed approach on different versions of a real-world dataset (DBpedia) from four different categories to evaluate the completeness of DBpedia over time. We have also assessed our approach experimentally using DBpedia and YAGO datasets.

Derive conceptual schemas from RDF datasets

Data is becoming a strategic asset in the information-driven world. One of the challenges facing companies and researchers is to improve their visibility and understandability of the data they manage and use. We have used a mining approach to infer a conceptual schema from RDF data. We have introduced a novel approach to automatically extract

schema and represent it as a model from a data source using a user-specified threshold. The derived schema includes the chosen class, the attributes and the relationships between classes tagged by completeness values. This can be achieved using *LOD-CM* that is a web-based completeness demonstrator for DBpedia that is available online.

Evaluate the conciseness of Linked Data

Linking heterogeneous resources is not only a key research question but also a challenge, as long as it is clear that interchangeable data could be described in different vocabularies making it confusing. For this purpose, we have proposed an approach to discover equivalent predicates in linked datasets. Our proposed approach consists of three sequential phases. The first phase, which is a statistical analysis, discovers potential equivalent predicates. The next two phases exclude non-equivalent predicates based on the meaning of the predicates through the semantic features and the context where the predicate is used. Extensive experimental evaluation of real-world RDF knowledge bases (DBpedia and YAGO) has been done.

In the following section, we describe some perspectives on the current work.

7.2 Future directions

We expect several potential avenues for future work. First, as we have already indicated that a comprehensive Systematic Literature Review on the research topic in completeness of Linked Data has been conducted, one limitation of our work is that we have explored a state of the art using only the keywords relating to *Linked Data* as specified in Section 3.3.1. Therefore, this may lead to the omission of interesting approaches that do not use these keywords. As a future work, we plan to expand our research by adding more relative keywords such as *Knowledge Graph*.

Second, as mentioned in Section 6.1, we have evaluated our proposed approach on different versions of DBpedia datasets to evaluate the completeness. We would like to enrich our investigation with other datasets such as YAGO and IMDb. Furthermore, we look forward to study the reasons why some categories improved their completeness over

time while others did not.

Likewise, we plan to investigate the role of conceptual modeling through integrated system upon several Linked Open Data. We plan to add more linked datasets available and allow the user to compare easily two conceptual schemas side by side. We believe that the ability to compare two conceptual schemas of two datasets can help to choose the one that suits better for use.

Finally, we plan to further investigate the implementation of our proposed approach to assess the conciseness of Linked Data as illustrated in Chapter 5. In particular, we will focus on adding a new phase to deal with uncommon predicates and data-type predicates besides to object-type predicates. We will also study some ways of improving the F-measure by combing our approach with others, for example, the unsupervised data-driven method proposed by [Zhang, Gentile, Blomqvist, Augenstein, and Ciravegna 2017].

List of publications

Journal Publication

- 2019 – Subhi Issa, Onaopepo Adekunle, Fayçal Hamdi, Samira Si-said Cherfi, Michel Dumontier, Amrapali Zaveri: **Linked Data Completeness: A Systematic Literature Review**. *Semantic Web journal*. (under review)

Doctoral Consortium

- 2018 – Subhi Issa: **Linked Data Quality.** *The 17th International Semantic Web Conference (DC@ ISWC 2018), October 2018, pp.37–45, Monterey, California, USA.*

International Publications

- 2019 – Subhi Issa, Fayçal Hamdi, Samira Si-said Cherfi: **Enhancing the Conciseness of Linked Data by Discovering Synonym Predicates.** *The 12th International Conference on Knowledge Science, Engineering and Management (KSEM 2019), August 2019, pp.739–750, Athens, Greece.*

- 2019 – Subhi Issa, Pierre-Henri Paris, Fayçal Hamdi, Samira Si-said Cherfi: **Revealing the Conceptual Schema of RDF Datasets.** *31st International Conference on Advanced Information Systems Engineering (CAiSE 2019), June 2019, pp.312–327, Rome, Italy.*

- 2017 – Subhi Issa, Pierre-Henri Paris, Fayçal Hamdi: **Assessing the Completeness Evolution of DBpedia: A Case Study.** *The 36th International Conference on Conceptual Modeling (ER 2017) Workshops AHA, MoBiD, MREBA, OntoCom, and QMMQ, November 2017, pp.238–247, Valencia, Spain.*

National Publication

- 2016 – Fayçal Hamdi, Samira Si-said Cherfi, Subhi Issa: **Evaluation de l'évolution de la complétude de DBpedia: une étude de cas.** *Ingénierie des Connaissances: des Sources Ouvertes au WEb de DOnnées (IC-SoWeDo 2016), June 2016, pp.23-31, Montpellier, France.*

Bibliography

Rdf 1.1 concepts and abstract syntax, 2014. URL `http://www.w3.org/TR/rdf11-concepts/`.

Ziawasch Abedjan and Felix Naumann. Context and target configurations for mining rdf data. In *Proceedings of the 1st international workshop on Search and mining entity-relationship data*, pages 23–24. ACM, 2011.

Ziawasch Abedjan and Felix Naumann. Synonym analysis for predicate expansion. In *Extended Semantic Web Conference*, pages 140–154. Springer, 2013.

Maribel Acosta, Elena Simperl, Fabian Flöck, and Maria Esther Vidal. Enhancing answer completeness of SPARQL queries via crowdsourcing. *Journal of Web Semantics*, 45:41–62, 2017.

Riccardo Albertoni and Asunción Gómez Pérez. Assessing linkset quality for complementing third-party datasets. In *International Conference on Database Theory*, page 52, 2013.

Riccardo Albertoni, Monica De Martino, and Paola Podestà. A linkset quality metric measuring multilingual gain in SKOS Thesauri. In *CEUR Workshop Proceedings*, volume 1376, 2015.

Céline Alec, Chantal Reynaud-Delaître, and Brigitte Safar. A model for Linked Open Data acquisition and SPARQL query generation. In *International Conference on Conceptual Structures*, pages 237–251. Springer, 2016.

Mahmoud Ali and Mohammed Alchaita. Enhancing dbpedia quality using markov logic networks. *Journal of Theoretical and Applied Information Technology*, 96(12):3924–3936, 2018.

Ahmad Assaf, Raphaël Troncy, and Aline Senart. What's up LOD cloud? Observing the state of Linked Open Data cloud metadata. In *European Semantic Web Conference*, volume 9341, pages 247–254, 2015.

Ahmad Assaf, Aline Senart, and Raphaël Troncy. Towards An Objective Assessment Framework for Linked Data Quality. *International Journal on Semantic Web and Information Systems*, 12(3):111–133, 2016.

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. Dbpedia: A nucleus for a web of open data. In *The semantic web*, pages 722–735. Springer, 2007.

Vevake Balaraman, Simon Razniewski, and Werner Nutt. Recoin: Relative Completeness in Wikidata. In *International conference on World wide web 2018 (WWW'18)*, pages 1787–1792, 2018.

Donald P. Ballou and Harold L. Pazer. Modeling completeness versus consistency tradeoffs in information decision contexts. *Knowledge and Data Engineering, IEEE Transactions on*, 15(1):240–243, 2003.

Victor R. Basili, Gianluigi Caldiera, and Dieter H. Rombach. *The Goal Question Metric Approach*, volume I. John Wiley & Sons, 1994.

Carlo Batini, Cinzia Cappiello, Chiara Francalanci, and Andrea Maurino. Methodologies for data quality assessment and improvement. *ACM computing surveys (CSUR)*, 41(3):16, 2009.

Sean Bechhofer, Iain Buchan, David De Roure, Paolo Missier, John Ainsworth, Jiten Bhagat, Philip Couch, Don Cruickshank, Mark Delderfield, Ian Dunlop, et al. Why linked data is not enough for scientists. *Future Generation Computer Systems*, 29(2):599–611, 2013.

Behshid Behkamal. Metrics-driven framework for LOD quality assessment. In *International Semantic Web Conference*, volume 8465 LNCS, pages 806–816, 2014.

Behshid Behkamal, Mohsen Kahani, Ebrahim Bagheri, and Zoran Jeremic. A metrics-driven

approach for quality assessment of Linked Open Data. In *International Conference on Database and Expert Systems Applications*, volume 9, pages 64–79, 2014.

T. Berners-Lee. Linked Data., 2006a. URL `http://www.w3.org/DesignIssues/LinkedData.html`.

Tim Berners-Lee. Linked data-design issues. *http://www. w3. org/DesignIssues/LinkedData. html*, 2006b.

Tim Berners-Lee, James Hendler, Ora Lassila, et al. The semantic web. *Scientific american*, 284(5):28–37, 2001.

Christian Bizer and Richard Cyganiak. Quality-driven information filtering using the wiqa policy framework. *Journal of Web Semantics*, 7(1):1–10, 2009.

Christian Bizer, Tom Heath, and Tim Berners-Lee. Linked data: The story so far. In *Semantic services, interoperability and web applications: emerging concepts*, pages 205–227. IGI Global, 2011.

Sergey Brin, Rajeev Motwani, and Craig Silverstein. Beyond market baskets: Generalizing association rules to correlations. *Acm Sigmod Record*, 26(2):265–276, 1997.

Antoon Bronselaer, Robin De Mol, and Guy De Tre. A Measure-Theoretic Foundation for Data Quality. *IEEE Transactions on Fuzzy Systems*, 26(2):627–639, 2018.

Cinzia Cappiello, Tommaso Di Noia, Bogdan Alexandru Marcu, and Maristella Matera. A quality model for Linked Data exploration. In *International Conference on Web Engineering*, volume 9671, pages 397–404, 2016.

Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R Hruschka, and Tom M Mitchell. Toward an architecture for never-ending language learning. In *Twenty-Fourth AAAI Conference on Artificial Intelligence*, 2010.

Valentine Charles, Juliane Stiller, Peter Kiraly, and Werner Bailer. Data Quality Assessment in Europeana: Metrics for Multilinguality. *International Conference on Theory and Practice of Digital Libraries*, 2038(September):11, 2018.

Didier Cherix, Ricardo Usbeck, Andreas Both, and Jens Lehmann. CROCUS: Cluster-based ontology data cleansing. *CEUR Workshop Proceedings*, 1240:7–14, 2014.

Peter Christen and Karl Goiser. Quality and complexity measures for data linkage and deduplication, 2007.

Fariz Darari, Simon Razniewski, Radityo Eko Prasojo, and Werner Nutt. Enabling fine-grained RDF data completeness assessment. In *International Conference on Web Engineering*, volume 9671, pages 170–187, 2016.

Fariz Darari, Werner Nutt, Simon Razniewski, and Sebastian Rudolph. Ensuring the Completeness and Soundness of SPARQL Queries Using Completeness Statements about RDF Data Sources. *Semantic Web Journal*, 0(0), 2017.

Fariz Darari, Werner Nutt, and Simon Razniewski. Comparing Index Structures for Completeness Reasoning. *International Workshop on Big Data and Information Security, IWBIS 2018*, pages 49–56, 2018.

Jeremy Debattista, Christoph Lange, Sören Auer, and Dominic Cortis. Evaluating the quality of the lod cloud: An empirical investigation. *Semantic Web*, (Preprint):1–43, 2017.

Nick Drummond and Rob Shearer. The open world assumption. In *eSI Workshop: The Closed World of Databases meets the Open World of the Semantic Web*, volume 15, 2006.

Basil Ell, Denny Vrandečić, and Elena Simperl. Labels in the web of data. In *International Semantic Web Conference*, volume 7031 LNCS, pages 162–176, 2011.

Mohamed Ben Ellefi, Zohra Bellahsene, John G Breslin, Elena Demidova, Stefan Dietze, Julian Szymanski, and Konstantin Todorov. Dataset Profiling -a Guide to Features, Methods, Applications and Vocabularies. *Semantic Web*, 1:1–5, 2016.

Kemele M Endris, Zuhair Almhithawi, Ioanna Lytra, Maria Esther Vidal, and Sören Auer. BOUNCER: Privacy-Aware Query Processing over Federations of RDF Datasets. In *International Conference on Database and Expert Systems Applications*, volume 11029 LNCS, pages 69–84, 2018.

Sidra Faisal, Kemele M. Endris, Saeedeh Shekarpour, Sören Auer, and Maria Esther Vidal. Co-evolution of RDF datasets. *International Conference on Web Engineering*, 9671(June): 225–243, 2016.

Michael Färber, Frederic Bartscherer, Carsten Menne, and Achim Rettinger. Linked Data quality of DBpedia, Freebase, OpenCyc, Wikidata, and YAGO. *Semantic Web*, 9(1):77–129, 2018.

Michael Färber, Frederic Bartscherer, Carsten Menne, and Achim Rettinger. Linked data quality of dbpedia, freebase, opencyc, wikidata, and yago. *Semantic Web*, 9(1):77–129, 2018.

Kevin Chekov Feeney, Declan O'Sullivan, Wei Tai, and Rob Brennan. Improving Curated Web-Data Quality with Structured Harvesting and Assessment. *International Journal on Semantic Web and Information Systems*, 10(2):35–62, 2014.

Daniel Fleischhacker, Johanna Völker, and Heiner Stuckenschmidt. Mining rdf data for property axioms. In *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, pages 718–735. Springer, 2012.

Ludovic Font, Amal Zouaq, and Michel Gagnon. Assessing and Improving Domain Knowledge Representation in DBpedia. *Open Journal of Semantic Web*, 4(1), 2017.

Linyun Fu, Haofen Wang, Wei Jin, and Yong Yu. Towards better understanding and utilizing relations in dbpedia. *Web Intelligence and Agent Systems: An International Journal*, 10(3):291–303, 2012.

Christian Fürber and Martin Hepp. SWIQA - a Semantic Web Information Quality Assessment Framework. *European Conference on Information Systems (ECIS2011)*, page 76, 2011a.

Christian Fürber and Martin Hepp. Towards a vocabulary for data quality management in semantic web architectures. In *International Workshop on Linked Web Data Management*, LWDM '11, page 1, New York, NY, USA, 2011b. ACM.

Christian Fürber and Martin Hepp. Swiqa-a semantic web information quality assessment framework. In *ECIS*, volume 15, page 19, 2011.

Luis Galárraga, Christina Teflioudi, Katja Hose, and Fabian M Suchanek. Fast rule mining in ontological knowledge bases with amie+. *The VLDB Journal—The International Journal on Very Large Data Bases*, 24(6):707–730, 2015.

Gösta Grahne and Jianfei Zhu. Efficiently using prefix-trees in mining frequent itemsets. In *Proceedings of the ICDM 2003 Workshop on Frequent Itemset Mining Implementations, 19 December 2003, Melbourne, Florida, USA*, 2003.

Christophe Guéret, Paul Groth, Claus Stadler, and Jens Lehmann. Assessing Linked Data mappings using network measures. *European Semantic Web Conference*, 7295 LNCS: 87–102, 2012.

Kalpa Gunaratna, Krishnaprasad Thirunarayan, Prateek Jain, Amit Sheth, and Sanjaya Wijeratne. A statistical and schema independent approach to identify equivalent properties on linked data. In *Proceedings of the 9th International Conference on Semantic Systems*, pages 33–40. ACM, 2013.

Didem Gürdür, Jad El-khoury, and Mattias Nyberg. Methodology for linked enterprise data quality assessment through information visualizations. *Journal of Industrial Information Integration*, (November):0–1, 2018.

Fayçal Hamdi and Samira Si-Said Cherfi. An approach for measuring rdf data completeness. *BDA 2015 Gestion de Données–Principes, Technologies et Applications 29 septembre au 2 octobre 2015 ^Ile de Porquerolles*, page 32, 2015.

Jiawei Han, Jian Pei, and Yiwen Yin. Mining frequent patterns without candidate generation. In Weidong Chen, Jeffrey F. Naughton, and Philip A. Bernstein, editors, *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, May 16-18, 2000, Dallas, Texas, USA.*, pages 1–12. ACM, 2000. doi: 10.1145/342009.335372.

Jiawei Han, Jian Pei, Yiwen Yin, and Runying Mao. Mining frequent patterns without candidate generation: A frequent-pattern tree approach. *Data Min. Knowl. Discov.*, 8(1): 53–87, January 2004. ISSN 1384-5810. doi: 10.1023/B:DAMI.0000005258.31418.83.

Steve Harris, Andy Seaborne, and Eric Prud'hommeaux. Sparql 1.1 query language. *W3C recommendation*, 21(10):778, 2013.

Souleiman Hasan, Sean O'Riain, and Edward Curry. Towards unified and native enrichment in event processing systems. In *International conference on Distributed event-based systems*, DEBS '13, page 171, New York, NY, USA, 2013. ACM.

Ian Horrocks, Oliver Kutz, and Ulrike Sattler. The even more irresistible sroiq. *Kr*, 6:57–67, 2006.

Subhi Issa, Pierre-Henri Paris, and Fayçal Hamdi. Assessing the completeness evolution of dbpedia: A case study. In *ER Workshops*, volume 10651 of *Lecture Notes in Computer Science*, pages 238–247. Springer, 2017a.

Subhi Issa, Pierre Henri Paris, and Fayçal Hamdi. Assessing the completeness evolution of DBpedia: A case study. In *International Conference on Conceptual Modeling*, volume 10651 LNCS, pages 238–247, Cham, 2017b. Springer International Publishing.

Subhi Issa, Pierre-Henri Paris, Fayçal Hamdi, and Samira Si-Said Cherfi. Revealing the conceptual schemas of rdf datasets. In *International Conference on Advanced Information Systems Engineering*, pages 312–327. Springer, 2019.

Afraz Jaffri, Hugh Glaser, and Ian Millard. URI disambiguation in the context of Linked Data. In *CEUR Workshop Proceedings*, volume 369, 2008.

J.M. Juran, F.M. Gryna, and R.S. Bingham. *Quality control handbook*. McGraw-Hill handbooks. McGraw-Hill, 1974. ISBN 9780070331754. URL `https://books.google.fr/books?id=YtdTAAAAMAAJ`.

Jan-Christoph Kalo, Philipp Ehler, and Wolf-Tilo Balke. Knowledge graph consolidation by unifying synonymous relationships. 2019.

B. Kitchenham and S Charters. Guidelines for performing systematic literature reviews in software engineering. 2007.

Graham Klyne and Jeremy J Carroll. Resource description framework (rdf): Concepts and abstract syntax. 2006.

Tomas Knap and Jan Michelfeit. Linked Data Aggregation Algorithm: Increasing Completeness and Consistency of Data. *Provided by Charles University*, 2012.

Dimitris Kontokostas, Patrick Westphal, Sören Auer, Sebastian Hellmann, Jens Lehmann, Roland Cornelissen, and Amrapali Zaveri. Test-driven evaluation of Linked Data quality. In *International conference on World wide web - WWW '14*, pages 747–758, 2014.

Sylvain Kubler, Jérérmy Robert, Sebastian Neumaier, Jürgen Umbrich, and Yves Le Traon. Comparison of metadata quality in open data portals using the Analytic Hierarchy Process. *Government Information Quarterly*, 35(1):13–29, 2018.

Ora Lassila and Ralph R Swick. Resource description framework (rdf) model and syntax specification, w3c recommendation 22 february 1999, 1999. URL `https://www.w3.org/TR/1999/REC-rdf-syntax-19990222/`.

Yuangui Lei, Marta Sabou, Vanessa Lopez, Jianhan Zhu, Victoria Uren, and Enrico Motta. An infrastructure for acquiring high quality semantic metadata. In *European Semantic Web Conference*, pages 230–244. Springer, 2006.

Yuangui Lei, Victoria Uren, and Enrico Motta. A framework for evaluating semantic metadata. In *Proceedings of the 4th international conference on Knowledge capture*, pages 135–142. ACM, 2007.

Merkourios Margaritopoulos, Thomas Margaritopoulos, Ioannis Mavridis, and Athanasios Manitsaris. Quantifying and measuring metadata completeness. *Journal of the American Society for Information Science and Technology*, 63(4):724–737, 2012a. doi: 10.1002/asi.21706. URL `https://onlinelibrary.wiley.com/doi/abs/10.1002/asi.21706`.

Merkourios Margaritopoulos, Thomas Margaritopoulos, Ioannis Mavridis, and Athanasios Manitsaris. Quantifying and measuring metadata completeness. *Journal of the American Society for Information Science and Technology*, 63(4):724–737, 2012b. doi: 10.1002/asi.21706. URL `https://onlinelibrary.wiley.com/doi/abs/10.1002/asi.21706`.

Silvio McGurk, Charlie Abela, and Jeremy Debattista. Towards Ontology Quality Assessment. *CEUR Workshop Proceedings*, 2017.

Pablo N Mendes, Hannes Mühleisen, and Christian Bizer. Sieve : Linked Data quality assessment and fusion. In *International Conference on Database Theory*, number March, 2012a.

Pablo N Mendes, Hannes Mühleisen, and Christian Bizer. Sieve: linked data quality assessment and fusion. In *Proceedings of the 2012 Joint EDBT/ICDT Workshops*, pages 116–123. ACM, 2012b.

Natasha Micic, Daniel Neagu, Felician Campean, and Esmaeil Habib Zadeh. Towards a Data Quality Framework for Heterogeneous Data. In *IEEE International Conference on Internet of Things, IEEE Green Computing and Communications, IEEE Cyber, Physical and Social Computing, IEEE Smart Data, iThings-GreenCom-CPSCom-SmartData*, volume 2018-Janua, pages 155–162, 2017.

Nandana Mihindukulasooriya, Giuseppe Rizzo, Raphaël Troncy, Oscar Corcho, and Raúl García-Castro. A two-fold quality assurance approach for dynamic knowledge bases: The 3cixty use case. In *CEUR Workshop Proceedings*, volume 1586, pages 1–12, 2016.

Peter Mika. Flink: Semantic web technology for the extraction and analysis of social networks. *Web Semantics: Science, Services and Agents on the World Wide Web*, 3(2):211–223, 2005.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

Amihai Motro and Igor Rakov. Estimating the quality of databases. In *International Conference on Flexible Query Answering Systems*, pages 298–307. Springer, 1998.

Michalis Mountantonakis and Yannis Tzitzikas. How Linked Data can aid machine learning-based tasks. In *International Conference on Theory and Practice of Digital Libraries*, volume 10450 LNCS, pages 155–168, Cham, 2017. Springer International Publishing.

Dmitry Mouromtsev, Peter Haase, Eugene Cherny, Dmitry Pavlov, Alexey Andreev, and Anna Spiridonova. Towards the Russian Linked Culture Cloud: Data Enrichment and Publishing. In *European Semantic Web Conference*, volume 9088, 2015.

Mohammad Khodizadeh Nahari, Nasser Ghadiri, Zahra Jafarifard, Ahmad Baraani Dastjerdi, and Joerg R. Sack. A framework for Linked Data fusion and quality assessment. In *International Conference on Web Research*, pages 67–72, 2017.

Felix Naumann. *Quality-driven Query Answering for Integrated Information Systems*. Springer-Verlag, Berlin, Heidelberg, 2002. ISBN 3-540-43349-X.

Felix Naumann. *Quality-driven query answering for integrated information systems*, volume 2261. Springer, 2003.

Sebastian Neumaier, Jürgen Umbrich, and Axel Polleres. Automated Quality Assessment of Metadata across Open Data Portals. *Journal of Data and Information Quality*, 8(1):1–29, 2016.

Khai Nguyen, Ryutaro Ichise, and Bac Le. Interlinking Linked Data sources using a domain-independent system. In *Joint International Semantic Technology Conference*, volume 7774 LNCS, pages 113–128, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

Heiko Paulheim. Knowledge graph refinement: A survey of approaches and evaluation methods. *Semantic web*, 8(3):489–508, 2017.

Heiko Paulheim and Christian Bizer. Improving the Quality of Linked Data Using Statistical Distributions. *International Journal on Semantic Web and Information Systems*, 10(2):63–86, 2014.

Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

Leo L Pipino, Yang W Lee, and Richard Y Wang. Data quality assessment. *Communications of the ACM*, 45(4):211–218, 2002.

Filip Radulovic, Nandana Mihindukulasooriya, Raúl García-Castro, and Asunción Gómez-Pérez. A comprehensive quality model for Linked Data. *Semantic Web*, 9(1):3–24, 2018.

Giuseppe Rizzo, Marco Torchiano, and Politecnico Torino. KBQ - A Tool for Knowledge Base Quality Assessment Using Evolution Analysis. In *International Conference on Knowledge Capture*, pages 1–6, 2017.

Colette Rolland and Naveen Prakash. From conceptual modelling to requirements engineering. *Annals of Software Engineering*, 10(1-4):151–176, 2000.

Edna Ruckhaus, Maria Esther Vidal, Simón Castillo, Oscar Burguillos, and Oriana Baldizan. Analyzing Linked Data quality with LiQuate. *OTM Confederated International Conferences "On the Move to Meaningful Internet Systems"*, 8798:488–493, 2014.

Anisa Rula, Matteo Palmonari, and Andrea Maurino. Capturing the age of Linked Open Data: Towards a dataset-independent framework. In *International Conference on Semantic Computing*, ICSC '12, pages 218–225, Washington, DC, USA, 2012. IEEE Computer Society.

Anisa Rula, Luca Panziera, Matteo Palmonari, and Andrea Maurino. Capturing the currency of dbpedia descriptions and get insight into their validity. In *Proceedings of the 5th International Conference on Consuming Linked Data - Volume 1264*, COLD'14, pages 61–72, Aachen, Germany, Germany, 2014. CEUR-WS.org. URL `http://dl.acm.org/citation.cfm?id=2877789.2877795`.

Gerald Salton. Automatic text processing. 1988.

Monica Scannapieco and Carlo Batini. Completeness in the relational model: a comprehensive framework. In *ICIQ*, pages 333–345, 2004.

Andreas Schultz, Andrea Matteini, Robert Isele, Pablo N. Mendes, Christian Bizer, and Christian Becker. LDIF - A Framework for Large-Scale Linked Data Integration. In *21st International World Wide Web Conference (WWW2012), Developers Track*, page to appear, April 2012.

Pavel Shvaiko and Jérôme Euzenat. Ontology matching: state of the art and future challenges. *IEEE Transactions on knowledge and data engineering*, 25(1):158–176, 2011.

Dezhao Song and Jeff Heflin. Automatically generating data linkages using a domain-independent candidate selection approach. *Semantic Web*, 7031 LNCS(PART 1):649–664, 2011.

Heiner Stuckenschmidt and Frank Van Harmelen. *Information sharing on the semantic web*. Springer Science & Business Media, 2005.

Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: a core of semantic knowledge. In Carey L. Williamson, Mary Ellen Zurko, Peter F. Patel-Schneider, and Prashant J. Shenoy, editors, *Proceedings of the 16th International Conference on World Wide Web, WWW 2007*, pages 697–706. ACM, 2007a. doi: 10.1145/1242572.1242667. URL http://doi.acm.org/10.1145/1242572.1242667.

Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*, pages 697–706. ACM, 2007b.

Fabian M Suchanek, Serge Abiteboul, and Pierre Senellart. Paris: Probabilistic alignment of relations, instances, and schema. *Proceedings of the VLDB Endowment*, 5(3):157–168, 2011.

Harsh Thakkar, Kemele Endris, Jose Gimenez Garica, Jeremy Debattista, Christoph Lange, and Sören Auer. Are Linked Datasets fit for Open-domain Question Answering? A Quality Assessment. In *International Conference on Web Intelligence, Mining and Semantics*, number June, pages 1–12, 2016.

James Thomas, Anna Noel-Storr, Iain Marshall, Byron Wallace, Steven McDonald, Chris Mavergames, Paul Glasziou, Ian Shemilt, Anneliese Synnot, Tari Turner, and Julian Elliott. Living systematic reviews: 2. combining human and machine effort. *Journal of Clinical Epidemiology*, 91:31 – 37, 2017. ISSN 0895-4356. doi: https://doi.org/10.1016/j.jclinepi.2017.08.011. URL http://www.sciencedirect.com/science/article/pii/S0895435617306042.

Gerald Töpper, Magnus Knuth, and Harald Sack. Dbpedia ontology enrichment for

inconsistency detection. In *Proceedings of the 8th International Conference on Semantic Systems*, pages 33–40. ACM, 2012.

Johanna Völker and Mathias Niepert. Statistical schema induction. In *Extended Semantic Web Conference*, pages 124–138. Springer, 2011.

Denny Vrandečić and Markus Krötzsch. Wikidata: a free collaborative knowledge base. 2014.

Yair Wand and Richard Y Wang. Anchoring data quality dimensions in ontological foundations. *Communications of the ACM*, 39(11):86–95, 1996.

Richard Y Wang and Diane M Strong. Beyond accuracy: What data quality means to data consumers. *Journal of management information systems*, 12(4):5–33, 1996.

Julie Weeds, Daoud Clarke, Jeremy Reffin, David Weir, and Bill Keller. Learning to distinguish hypernyms and co-hyponyms. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 2249–2259. Dublin City University and Association for Computational Linguistics, 2014.

Mark D Wilkinson, Michel Dumontier, IJsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, Jan-Willem Boiten, Luiz Bonino da Silva Santos, Philip E Bourne, et al. The fair guiding principles for scientific data management and stewardship. *Scientific data*, 3, 2016.

Najme Yaghouti, Mohsen Kahani, and Behshid Behkamal. A metric-driven approach for interlinking assessment of RDF graphs. In *International Symposium on Computer Science and Software Engineering*, pages 1–8, aug 2015.

Amrapali Zaveri, Andrea Maurino, and Laure-Berti Equille. Web data quality: Current state and new challenges. *Int. J. Semant. Web Inf. Syst.*, 10(2):1–6, apr 2014. ISSN 1552-6283. doi: 10.4018/ijswis.2014040101. URL `http://dx.doi.org/10.4018/ijswis.2014040101`.

Amrapali Zaveri, Anisa Rula, Andrea Maurino, Ricardo Pietrobon, Jens Lehmann, and Sören Auer. Quality assessment for linked data: A survey. *Semantic Web*, 7(1):63–93, 2016.

Ziqi Zhang, Anna Lisa Gentile, Eva Blomqvist, Isabelle Augenstein, and Fabio Ciravegna. An unsupervised data-driven method to discover equivalent relations in large linked datasets. *Semantic web*, 8(2):197–223, 2017.

le cnam

Subhi ISSA

Linked Data Quality: Completeness and Conciseness

le cnam

Abstract:

The wide spread of Semantic Web technologies such as the Resource Description Framework (RDF) enables individuals to build their databases on the Web, write vocabularies, and define rules to arrange and explain the relationships between data according to the Linked Data principles. As a consequence, a large amount of structured and interlinked data is being generated daily. A close examination of the quality of this data could be very critical, especially, if important research and professional decisions depend on it. The quality of Linked Data is an important aspect to indicate their fitness for use in applications. Several dimensions to assess the quality of Linked Data are identified such as accuracy, completeness, provenance, and conciseness. This thesis focuses on assessing completeness and conciseness of Linked Data. In particular, we first proposed a completeness calculation approach based on a generated schema. Indeed, as a reference schema is required to assess completeness, we proposed a mining-based approach to derive a suitable schema (i.e., a set of properties) from data. This approach distinguishes between essential properties and marginal ones to generate, for a given dataset, a conceptual schema that meets the user's expectations regarding data completeness constraints. We implemented a prototype called "LOD-CM" to illustrate the process of deriving a conceptual schema of a dataset based on the user's requirements. We further proposed an approach to discover equivalent predicates to assess the conciseness of Linked Data. This approach is based, in addition to a statistical analysis, on a deep semantic analysis of data and on learning algorithms. We argue that studying the meaning of predicates can help to improve the accuracy of results. Finally, a set of experiments was conducted on real-world datasets to evaluate our proposed approaches.

Keywords:

Semantic Web, Linked Data, Knowledge Graph, Data Quality, Data Completeness, Data Conciseness, RDF, Schema Mining, Quality Evaluation.

Résumé :

La diffusion à large échelle des technologies du Web Sémantique telles que le *Resource Description Framework* (RDF) a permis aux individus de construire leurs bases de données sur le Web, d'écrire des vocabulaires et de définir des règles pour organiser et expliquer les relations entre les données selon les principes des Données Liées. En conséquence, une grande quantité de données structurées et interconnectées est générée quotidiennement. Un examen attentif de la qualité de ces données pourrait s'avérer très critique, surtout si d'importantes recherches et décisions professionnelles en dépendent. La qualité des Données Liées est un aspect important pour indiquer leur aptitude à être utilisées dans des applications. Plusieurs dimensions permettant d'évaluer la qualité des Données Liées ont été identifiées, telles que la précision, la complétude, la provenance et la concision. Dans cette thèse nous nous sommes focalisés sur l'étude de la complétude et de la concision des Données Liées. Dans un premier temps, nous avons proposé une approche de calcul de complétude fondée sur un schéma généré. En effet, comme un schéma de référence est nécessaire pour évaluer la complétude, nous avons proposé une approche fondée sur la fouille des données pour obtenir ce schéma (c.-à-d. un ensemble de propriétés). Cette approche permet de distinguer les propriétés essentielles des propriétés marginales pour générer, pour un jeu de données, un schéma conceptuel qui répond aux attentes de l'utilisateur par rapport aux contraintes de complétude souhaitées. Nous avons implémenté un prototype appelé "LOD-CM" pour illustrer le processus de dérivation d'un schéma conceptuel d'un jeu de données fondé sur les besoins de l'utilisateur. Dans un second temps, nous avons proposé une approche pour découvrir des prédicats synonymes afin d'évaluer la concision des Données Liées. Cette approche s'appuie, en plus d'une analyse statistique, sur une analyse sémantique approfondie des données et sur des algorithmes d'apprentissage. Ces analyses permettent de mieux capter le sens des prédicats et ainsi améliorer la précision des résultats obtenus. Enfin, un ensemble d'expériences a été mené sur des jeux de données réelles afin d'évaluer nos différentes approches.

Mots clés :

Web Sémantique, Données Liées, Graphe de Connaissances, Qualité des Données, Complétude des Données, Concision des Données, RDF, Extraction de Schéma, Evaluation de la Qualité.