Identity Management in Knowledge Graphs

Joe Raad

université PARIS-SACLAY

NNT : 2018SACLA028

Thèse de doctorat

AgroParisTech
INSTITUT DES SCIENCES ET INDUSTRIES DU VIVANT ET DE L'ENVIRONNEMENT
PARIS INSTITUTE OF TECHNOLOGY FOR LIFE, FOOD AND ENVIRONMENTAL SCIENCES

Gestion d'identité dans des graphes de connaissances

Thèse de doctorat de l'Université Paris-Saclay
préparée à AgroParisTech

École doctorale n°581 Agriculture, Alimentation, Biologie, Environnement et Santé (ABIES)
Spécialité de doctorat: Informatique appliquée

Thèse présentée et soutenue à Paris, le 30 novembre 2018, par

Joe Raad

Composition du Jury :

Mme Sarah Cohen Boulakia Professeure, Université Paris-Sud	Présidente
Mme Catherine Faron Zucker Maître de Conférences (HDR), Université Nice Sophia Antipolis	Rapporteur
M. Mathieu d'Aquin Professeur, National University of Ireland Galway	Rapporteur
M. Harry Halpin Chercheur, Massachusetts Institute of Technology	Examinateur
M. Pascal Molli Professeur, Université de Nantes	Examinateur
Mme Juliette Dibie Professeure, AgroParisTech	Directrice de thèse
Mme Nathalie Pernelle Maître de Conférences (HDR), Université Paris-Sud	Co-Directrice de thèse
Mme Fatiha Saïs Maître de Conférences, Université Paris-Sud	Co-Encadrante, Invitée
Mme Liliana Ibanescu Maître de Conférences, AgroParisTech	Co-Encadrante, Invitée

Identity Management in Knowledge Graphs

A Dissertation

Submitted in Partial Satisfaction of the

Requirements for the Degree of

Doctor of Philosophy

in

Computer Science

from the

University of Paris-Saclay

by

Joe Raad

November 2018

IDENTITY MANAGEMENT IN KNOWLEDGE GRAPHS

ABSTRACT

In the absence of a central naming authority in the Web of data, it is common for different knowledge graphs to refer to the same thing by different names (IRIs). Whenever multiple names are used to denote the same thing, owl:sameAs statements are needed in order to link the data and foster reuse. Such identity statements have strict logical semantics, indicating that every property asserted to one name, will also be inferred to the other, and vice versa. While such inferences can be extremely useful in enabling and enhancing knowledge-based systems such as search engines and recommendation systems, incorrect use of identity can have wide-ranging effects in a global knowledge space like the Web of data. With several studies showing that owl:sameAs is indeed misused for several reasons, a proper approach towards the handling of identity links is required in order to make the Web of data succeed as an integrated knowledge space.

This thesis investigates the identity problem at hand, and provides different, yet complementary solutions. Firstly, it presents the largest dataset of identity statements that has been gathered from the LOD Cloud to date, and a web service from which the data and its equivalence closure can be queried. Such resource has both practical impacts (it helps data users and providers to find different names for the same entity), as well as analytical value (it reveals important aspects of the connectivity of the LOD Cloud). In addition, by relying on this collection of 558 million identity statements, we show how network metrics such as the community structure of the owl:sameAs graph can be used in order to detect possibly erroneous identity assertions. For this, we assign an error degree for each owl:sameAs based on the density of the community(ies) in which they occur, and their symmetrical characteristics. One benefit of this approach is that it does not rely on any additional knowledge. Finally, as a way to limit the excessive and incorrect use of owl:sameAs, we define a new relation for asserting the identity of two ontology instances in a specific context (a sub-ontology). This identity relation is accompanied with an approach for automatically detecting these links, with the ability of using certain expert constraints for filtering irrelevant contexts. As a first experiment, the detection and exploitation of the detected contextual identity links are conducted on a knowledge graph for life sciences, constructed in a mutual effort with domain experts from the French National Institute of Agricultural Research (INRA).

To my Grandfather in heaven...

ACKNOWLEDGEMENTS

First I would like to express my sincere gratitude to my advisors Juliette, Liliana, Nathalie and Fatiha for giving me the opportunity to prepare my PhD under their supervision for the past three years. I still remember that sunny spring day in 2015 when I met Juliette and Nathalie for the first time in AgroParisTech for my interview. I was sure back then that this was the convenient place to prepare my PhD. Three years, with countless discussions, decisions and trips later, I have no doubt that I made the right choice. The readiness of Juliette and Liliana to offer advice, support and guidance whenever needed, is something I truly appreciate. Their experience and intervention at the right moments, have massively contributed to the current shape of this manuscript and the success of this project. On the other side of Paris, the positive energy, the patience, the incredible attention to details, and the equal treatment I received from Nathalie and Fatiha is something I will never forget. They have devoted so much of their personal time to guide me through this journey, and I will always be grateful for this. Also, having shared many moments together outside our usual work environment, I had the chance to discover their joyful personalities and their great sense of humour.

I thank the members of my thesis committee, Prof. Sarah Cohen Boulakia, Dr. Catherine Faron Zucker, Prof. Mathieu d'Aquin, Dr. Harry Halpin, and Prof. Pascal Molli for devoting their time to read the manuscript and for their insightful comments and remarks.

I also thank the members of my thesis supervisory committee Dr. Cédric Pruski and Prof. Sandra Bringay for their valuable guidance during these years.

I would like to thank all my colleagues at MIA-Paris, LRI and the computer science department of the IUT of Orsay for supporting me and for making me feel as a part of their team from the first moment. In particular, I would like to thank the head of the MIA-Paris research unit Prof. Liliane Bel, the head of the LInK team Prof. Antoine Cornuéjols and the previous head of the LaHDAK team and current head of the computer science department at the IUT of Orsay Prof. Chantal Reynaud for their trust and support.

I would also like to thank my current colleagues at the KR&R group at the Vrije Universiteit in Amsterdam. Thank you for instantly making me feel as part of your team. In particular, my warm gratitude goes for Prof. Frank van Harmelen and Dr. Wouter Beek for their trust and assistance. I am grateful for all your contributions to this thesis, and I am very lucky to be part of your amazing group.

I would like to thank the organizers and the members of the LIONES project, supported by the Center for Data Science, funded by the IDEX Paris-Saclay, ANR-11-IDEX-0003-02. In particular, I would like to thank Dr. Caroline Pénicaud and Dr. Elisabeth Guichard for their valuable expertise and contributions in this project.

My warm gratitude goes to my fellow lab mates and friends who have made my journey fun and enjoyable. In particular, I thank Stéphane Dervaux for always being there for me, for teaching me the French culture, and for supporting my awful French on an every-day basis. Being surrounded by very good friends made my life in Paris even more beautiful. I thank Jieying, Luis, Rana, Yann, Pierre-Alexandre, Sema, Mélanie, Irène, Martina, Marie, and Luca for the great moments we spent in these three years.

Special thanks to my amazing friends in Lebanon, Roni, Roy, Karl, Alain, and Alfred for all the special moments. Knowing that I will see you when I come back home was always a great motivation.

My heartfelt gratefulness also goes to my family in France, Nada, Georges, and Joanna. Since day one, you have made me feel at home when "home" was so far away. 'Khalto', your endless love and care means a lot to me.

I also thank, with great affection, my love Annarosa who has been by my side since the day I met her. You have been there during my ups and downs, and an amazing support in all times. Words cannot describe how lucky I am for having you by my side and in my life. Ti amo.

My warmest gratitude goes to my beloved parents and sister, Maha, Georges, Maya and Elio. Your unconditional love and endless sacrifices have made all this possible and got me where I am today. Mother, knowing that you're by my side during all my best and worst moments, made this journey incredibly easy. Thanks for your never ending support. A special thanks to Celia and Mark for the joy they have added to our lives. Being around your innocence makes my world a happier place. My warm gratitude also goes to my grandmothers Wafa and Souad, my aunts Hoda, Hali, Ghada and Nada. Thank you for your prayers, and for always being by my side.

This list would not be complete if I leave out my grandfathers Joseph and Elias to whom I dedicate this PhD. Until we meet again 'Jeddos'.

Last but not least, all praise, honour and glory to my Lord Jesus Christ for His richest grace and mercy for the accomplishment of this thesis.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

Since its adoption by Google in 2012, the term Knowledge Graph has rapidly evolved. Previously referring to a single project for semantically enhancing Google's search results [Singhal, 2012], this term currently refers to a wide range of graphs surging from academic research, community-driven efforts, and industrial projects, such as DBpedia[1], Wikidata[2], and the Facebook Social Graph[3]. Although Google have reaped the credits for its ever increasing popularity, the term knowledge graph has been around for years, making an appearance in Bakker's PhD dissertation [Bakker, 1987] as part of a Dutch project aiming at integrating and structuring scientific knowledge [Nurdiati and Hoede, 2008]. From a broad perspective, any graph-based representation of some knowledge in a machine-readable format, can be described as knowledge graph. However, many argue that knowledge graphs should fulfil certain requirements, necessary for enabling and enhancing various knowledge-based applications, such as semantic searches, intelligent chatbots, fraud detections, and recommendation systems. For instance, [Huang et al., 2017] mention size as a characteristic of knowledge graphs, while [Paulheim, 2017] requires the coverage of a major portion of domains, and [Färber et al., 2016] have restricted the use of this term to RDF[4] graphs. Adopting some of these proposed, more restrictive, definitions will affect the status of several existing knowledge graphs, since not all knowledge graphs are RDF graphs, or domain independent.

With the lack of a formal and standardized definition, a number of guiding principles have emerged for helping data publishers create high quality data and knowledge graphs. While some of the proposed principles, such as FAIR [Wilkinson et al., 2016], have provided a set of goals to ensure that published data are findable, accessible, interoperable, and reusable, independently of the technology used, other principles have acted as a set of methods and steps for publishing open and reusable data on the Web. The most known set of principles were laid out by Tim Berners-Lee in 2006, with the goal of encouraging people to use HTTP[5] IRIs[6] for naming things, and using W3C[7] standards for describing these IRIs (e.g. RDF(S) and OWL[8]), and linking them to other IRIs for providing context. This set of widely adopted principles, known as the Linked Data principles, refers to a set of best practices for publishing structured data on the Web so it can be easily interlinked and managed using semantic queries. The

[1] `https://wiki.dbpedia.org`
[2] `https://www.wikidata.org`
[3] `https://developers.facebook.com/docs/graph-api`
[4] Resource Description Framework
[5] Hypertext Transfer Protocol
[6] Internationalized Resource Identifiers
[7] World Wide Web Consortium
[8] Web Ontology Language

idea is by providing simple principles, for creating and publishing structured data, publishers can also enrich, access, and benefit from a larger decentralized knowledge graph, known as the Web of Data.

Despite the adoption of the Linked Data principles, achieving the FAIR goals still poses a number of significant practical and research challenges, particularly in terms of the interoperability and re-usability of the published data. Firstly, adopting standard knowledge representation languages for expressing, explicit and implicit, domain knowledge still poses particular challenges. Specifically, when dealing with complex domains such as medical and life sciences data, there is a need to express certain types of axioms and relations, that can not be intuitively expressed in even some of the most expressive standardized languages, such as OWL 2 DL. These limitations in the language prompt various research questions discussed in [Krisnadhi et al., 2015], and pose several challenges for modellers to express the necessary knowledge using current standards and best practices. In addition, and while adopting such standardized knowledge representation languages guarantees interoperability at a syntactic level, one of the important challenges consists in achieving interoperability at the semantic level [d'Aquin and Noy, 2012]. Semantic interoperability is the ability to meaningfully and accurately exchange and interpret information produced by different sources. Creating semantically interoperable knowledge graphs requires considerable efforts, and poses several practical challenges for modellers in finding, evaluating and reusing existing well-established models to describe their data. Finally, achieving semantically interoperable knowledge graphs requires making links to other people's data. Such semantic interlinking is typically performed by asserting that two names (IRIs) denote the same real world entity. For this purpose, the Web Ontology Language OWL have introduced the `owl:sameAs` identity predicate. For instance, the triple $\langle \textit{President\_Barack\_Obama}, \texttt{owl:sameAs}, 44\textit{th\_US\_president} \rangle$ asserts that both names actually refer to the same person. Such identity statements indicate that every property asserted to one name will be also inferred to the other, allowing both names to be substituted in all contexts. While such inferences can be extremely useful in enabling and enhancing knowledge-based systems, incorrect use of identity can have wide-ranging effects in a global knowledge space like the Web of Data. With studies dating back to the early Linked Data days showing that `owl:sameAs` is indeed misused in the Web [Jaffri et al., 2008, Ding et al., 2010a, Halpin et al., 2010], one can trace back their presence to several factors. Firstly, most `owl:sameAs` links are generated by heuristic entity resolution techniques, that employs practical strategies which are not guaranteed to be accurate. For instance, an algorithm matching books based on the similarity of their titles and authors is not always accurate, as two different editions of the same book can also share both these traits without being the same, since they do not share the same number of pages. In addition, identity does not hold across all contexts, as things can be considered identical

for some people in certain contexts, while being different in other contexts. For instance, drugs sharing the same chemical structure, but produced by different companies, are considered identical in a scientific context, but are different in a commercial one.

Since suitable alternatives to `owl:sameAs` have yet to exist, or are rarely used in practice, a given Linked Data application is forced to make a choice with respect to each `owl:sameAs` assertion it encounters. This problem of incorrect use of identity is not specific to the Web of Data, and is present in all Knowledge Representation systems [Grant and Subrahmanian, 1995, Nguyen, 2007]. However, the problem is specifically alerting in the Web of Data due to its unprecedented size, the heterogeneity of its users and contents, and the lack of a central naming authority. By now, the problem of the identity use in the Semantic Web is widely recognized, and has been referred to as the "Identity Crisis" [Bouquet et al., 2007], and the "sameAs problem" [Halpin et al., 2010]. As such, a proper approach towards the handling of identity links is required in order to make the Web of Data succeed as an integrated knowledge space.

1.1 Objectives & Contributions

Identity management in knowledge graphs is the main objective of this thesis. Despite its ambitious title, this thesis is a modest attempt to address one particular issue of the identity problem: the excessive and incorrect use of identity links in knowledge graphs. It does not cover related but distinct research topics such as entity resolution and ontology alignment, that focus on techniques [Ferrara et al., 2013] and frameworks [Nentwig et al., 2017] for establishing `owl:sameAs` links. In addition, this thesis does not address the historically significant distinction between locating an electronic document with a URL and denoting an RDF resource with an IRI, known as the problem of Sense and Reference [Halpin, 2010]. This thesis investigates the use of `owl:sameAs` links in the Web of Data, and provides different, yet complementary solutions for this identity problem:

- **Identity Management Service** [Beek et al., 2018]. In order to uncover different aspects of the use of identity in the Semantic Web, and to facilitate access to a large number of identity statements, we propose `sameas.cc`: a web service and a dataset containing the largest number of identity statements that has been gathered from the Web of Data to date. This service provides public access (query and download) to over 558 million distinct `owl:sameAs` statements extracted from the Web of Data. It also provides access to these links' equivalence closure, and the resulting identity sets. For this, we propose an efficient approach for computing and storing the equivalence closure, that exploits the `owl:sameAs` transitive semantics.

The extracted identity statements, and their equivalence closure are accessible at our identity management service: `http://sameas.cc`.

- **Approach for detecting erroneous identity links** [Raad et al., 2018a, Raad et al., 2018b]. With many previous studies showing that identity links are incorrectly used in the Web of Data, there is an ever increasing need to detect these links to ensure the quality of knowledge graphs. For this, we propose an approach for automatically detecting potentially erroneous identity links, by making use of the `owl:sameAs` network topology, and more specifically the network's community structure. Based on the detected communities, an error degree is calculated for each identity link which is subsequently used for ranking these links, allowing potentially erroneous ones to be flagged, and potentially correct ones to be validated. Since the here presented approach is specifically developed in order to be applied to real-world data, the evaluation is run on the `sameas.cc` dataset. The implementation of this approach is available at `https://github.com/raadjoe/LOD-Community-Detection`.
- **A contextual identity relation** [Raad et al., 2017a, Raad et al., 2017b]. In many instances the classical interpretation of identity is too strong for particular purposes, and is not always required, as the notion of identity might change depending on the context. For instance, in some applications, the fact that drugs share the same chemical structure is sufficient to consider them as equivalent, while in other applications it is also necessary that these drugs share the same name. Unfortunately, modelling the specific contexts in which an identity relation holds is cumbersome and, due to arbitrary reuse, and the Open World Assumption, it is impossible to anticipate all contexts in which an entity will be used. For this, we define a new contextual identity relation. This relation expresses an identity between two class instances, that is valid in a context defined regarding a domain ontology. For automatically generating these contextual identity assertions, we propose an algorithm named `DECIDE` (DEtecting Contextual IDEntity). This algorithm detects the most specific contexts in which a pair of instances are identical. In addition, and since not all contexts may be relevant (e.g. a context considering a value without its unit of measure), this algorithm can be guided by different sets of semantic constraints provided by experts for enhancing the detected contexts. The implementation of this approach is available at `https://github.com/raadjoe/DECIDE_v2`.
- **Contextually linked knowledge graphs for life sciences** [Ibanescu et al., 2016, Raad et al., 2018c]. Cases in which objects can not be declared the same are quite common in scientific data, where experiments are mostly conducted by several scientists, in various circumstances, using similar but different products. This incapacity of semantically linking slightly different experiments has been a serious barrier for knowledge-based systems to fully exploit scientific

data, as they are either weakly connected with little semantics (e.g. using `skos:closeMatch`), or are incorrectly declared the same (using `owl:sameAs`). In addition, the classic problems of the heterogeneity of the formats in which scientific data are published, and the terminological variations encountered across the multiple scientific datasets also remain serious barriers in fully exploiting the large amount of data produced everyday. As a way for limiting these syntactic, semantic and identity problems, we introduce a new knowledge graph for life sciences. This graph is constructed in a mutual effort with domain experts from the French National Institute of Agricultural Research (INRA), describing two different domains: the mechanisms leading to the release of flavour compounds during food consumption, and the process of stabilisation of micro-organisms. As a way for semantically linking the different conducted experiments and their participants, we apply our approach for detecting contextual identity links. In addition, we exploit the millions of detected contextual identity links in this graph for discovering certain rules. These rules, when validated by the experts, can be used to predict with a certain degree of confidence, unobserved measures in the experiments, and consequently deployed for completing the constructed knowledge graph. This knowledge graph can be queried and downloaded at `http://sonorus.agroparistech.fr:7200`.

1.2 Thesis Outline

This classic identity problem, recently amplified in the context of the Web of Data, has led to several analysis, discussions, and proposals for limiting its effects. Chapter 2 gives an overview on the proposed solutions, and reflects on the current state of this "sameAs problem". Chapter 3 presents our first contribution for limiting this problem, by introducing the `sameas.cc` dataset and web service, which we deploy for performing several analyses on the use of identity in the Web of Data. Chapter 4 presents our approach of detecting erroneous identity links using network metrics, and the experiments conducted on a large subset of the Web of the Data. Chapter 5 introduces our new contextual identity relation, and presents our approach for automatically detecting these links in an RDF knowledge graph. Chapter 6 presents a new knowledge graph for life sciences, and presents a first use case of exploiting these detected contextual identity links for discovering certain rules, that can help completing the knowledge graph. Chapter 7 summarizes the results of the research presented in this thesis, and discusses its limitations, and some lines for future work.

CHAPTER 2
STATE OF THE ART

Identity is an old and thorny topic. Classically speaking, resources that are identical are considered to share the same properties. With Ψ denoting the set of all properties, this 'Indiscernibility of Identicals' ($a = b \rightarrow (\forall_{\psi \in \Psi})(\psi(a) = \psi(b))$) is attributed to Leibniz [Forrest, 2008] and its converse, the 'Identity of Indiscernibles' ($(\forall_{\psi \in \Psi})(\psi(a) = \psi(b)) \rightarrow a = b$), states that resources that share the same properties are identical. Identity statements play an important role in deduction. Firstly, objects that are known to not share some property, in a closed world assumption setting, are also known to not be identical. Secondly, from the premises $\psi(a)$ and $a = b$ it follows that $\psi(b)$ is also the case. In fact, this latter deduction is central to the Semantic Web notion of Linked Data. Specifically, it allows complementary descriptions of the same resource to be maintained locally, yet interchanged globally, merely by interlinking the names that are used in those respective descriptions. Hence, it becomes clear why the classical notion of identity is used to establish the Linked Data paradigm, and is standardized/formalized as part of the Web Ontology Language (OWL). However, there are also problems with it, and – consequently – criticisms have been levelled against it. We briefly present some of the well-known issues.

Firstly, although this classical notion provides necessary and sufficient conditions for identity, it does not provide an effective procedure for enumerating the extension of the identity relation. In fact, no finite number of facts about a and b can lead us to conclude that they denote the same resource, except for the identity assertion ($a = b$) itself. As such, identity statements can by definition not be deduced from other facts. Secondly, identity over time can pose problems, as a ship[1] may still be considered the same ship, even though some, or even all, of its original components have been replaced by new ones [Lewis, 1986]. In addition, identity does not hold across modal context, allowing Lois Lane to believe that Superman saved her without requiring her to believe that Clark Kent saved her. Finally, identity is context-dependent [Geach, 1967], allowing two medicines, having the same chemical structure, to be considered the same in a medical context, but to be considered different in other contexts (e.g. because they are produced by different companies). These issues in the classical identity definition have led to various philosophical theories, such as the distinction between accidental properties (traits that could be taken away from an object without making it a different thing), and essential properties (core elements needed for a thing to be the thing that it is) [Kripke, 1972]. However, it can be difficult to find an object's essential properties, since a tree can lose all its leaves and still be considered a tree, but a tree cut down and made into a notebook is not considered a tree. Hence, finding out at which point did a tree loses its identity (i.e. lost its essential properties) depends on each context.

[1] Reference to the ship of Theseus or Theseus's paradox

Given that this highly problematic notion of identity is also standardized as part of the Web Ontology Language (OWL), it is normal to encounter the same issues in Semantic Web applications. In fact, and due to the Open World Assumption and the continuous increase of Ψ, identity assertions in the Semantic Web are even more controversial. Firstly, unless two things are explicitly said to be different (e.g. using `owl:differentFrom`), the absence of an identity statement between them does not mean that they are not identical. Compared to the 558M `owl:sameAs` assertions in the 2015's copy of the LOD Cloud [Fernández et al., 2017], this type of assertions is barely present in the Web of Data, with only 3.6K `owl:differentFrom` assertions existing at that time in this same dataset. Secondly, stating that two IRIs are `owl:sameAs`, implies that both these IRIs unambiguously refer to the same real world entity (e.g. the 44th US president Barack Obama). However, some existing identity links do not carefully consider the difference between the IRI referring to a non-information resource (in that case the person Barack Obama), and its corresponding information resource (which is the URL referring to his Web page), leading to the long discussion of "Sense and Reference" [Halpin and Presutti, 2009, Halpin, 2010] which is beyond the scope of this thesis. Finally, studies have shown that modellers have different opinions about whether two objects are the same or not. For instance, in a 2010 analysis [Halpin et al., 2010], three semantic web experts were asked to judge 250 `owl:sameAs` links collected from the Web. This evaluation shows high disagreements, with one judge confirming the correctness of only 73 `owl:sameAs` statements, whilst the two other experts judging up to 132 and 181 links as true `owl:sameAs` assertions. A follow up study in 2015 [Halpin et al., 2015], shows that even more disagreements were encountered when authors evaluate `owl:sameAs` links resulted from inference. While in some cases this may be due to differences in modelling competence, there is also the problem that two modellers may consider different parts of the same knowledge graph within different contexts.

This classic identity problem, recently amplified in the context of the Web of Data, has led to several analyses, discussions, and proposals for limiting its effects. This chapter presents an overview on existing empirical analyses of the `owl:sameAs` use (section 2.1), services designed for managing identity in the Semantic Web (section 2.2), solutions for detecting erroneous identity assertions (section 2.3), and possible alternatives for `owl:sameAs` (section 2.4). Finally, this chapter reflects on the current state of the "sameAs problem" (section 2.5).

2.1 Identity Analysis

The special status of `owl:sameAs` links has motivated several studies into investigating the use of these links in the Web of Data, with each study focusing on specific aspects of identity.

Some studies have focused on the use of identity at the aggregated level of datasets, in order to better understand the common interests between different Linked Data publishers. In such studies, graph nodes represent the datasets, and the weighted edges represent the number of `owl:sameAs` linking the dataset resources. For grouping the retrieved resources into datasets, these studies assume that all data originating from one pay-level domain (PLD) belongs to a single dataset. In an early study, the authors of [Ding et al., 2010b] extracted 8.7M `owl:sameAs` triples from the 2010 Billion Triple Challenge dataset[2]. By visualizing the largest connected component, this study shows that densely connected clusters usually represent datasets that cover similar topics (e.g. a cluster of datasets that publish data related to scientific publications, and a cluster of bioinformatics datasets). A later analysis [Schmachtenberg et al., 2014] crawled 1,014 datasets containing 8M terms. The entire graph of datasets was found to consist of 9 weakly connected components with the largest one containing 297 datasets. This study shows that `dbpedia.org` has the largest indegree (89 datasets asserting `owl:sameAs` links to DBpedia entities), and that `bibsonomy.org` has the largest out-degree (Bibsonomy entities are linked to 91 different datasets). The authors have also analysed the use of other linking predicates, within different categories (e.g. life sciences, geography, publications). This study shows that `owl:sameAs` is the most used predicate for linking within most categories, followed by `rdfs:seeAlso` for life sciences datasets and `foaf:knows` for social networking datasets.

Other studies have focused on analysing the graph structure of the `owl:sameAs` network. In such networks, nodes represent the RDF terms occurring in a certain `owl:sameAs` triple, and edges represent the `owl:sameAs` triples. In an early analysis [Ding et al., 2010b], the transitive closure of 8.7M `owl:sameAs` triples have resulted in a graph of 2.9M connected components (i.e. equivalence classes). Most of these classes are small (average size of 2.4 terms), with only 41 classes with hundreds of terms, and only two classes with thousands of terms. This study shows that `owl:sameAs` networks are not as large and complex as `foaf:knows` networks, with the vast majority having a star-like structure consisting of single central resource connected to a number of peripheral resources. In a later analysis, [Hogan et al., 2011] extracted 3.7M distinct `owl:sameAs` from a corpus of 947M distinct RDF triples, crawled from 3.9M RDF/XML web-documents in 2010. After transitive closure, the data formed 2.16M equivalence classes (average size of 2.65 terms). The largest equivalence class contains 8,481 terms, with 74% of the equivalence classes containing only two terms. Finally, in a 2014 analysis based on the 2011 Billion Triple Challenge dataset, [Wang et al., 2014] observed that the number of `owl:sameAs` statements per term approximates a power-law distribution with coefficient -2.528.

[2]Dataset crawled during March/April 2010 based on datasets provided by Falcon-S, Sindice, Swoogle, SWSE, and Watson using the MultiCrawler/SWSE framework

Finally, other type of analyses have focused on the quality of existing owl:sameAs links in the Web of Data. In such evaluations, Semantic Web experts were asked to manually judge if two IRIs, linked by an owl:sameAs link, actually refer to the same real-world entity, whilst carefully considering the difference between non-information resources and information resources. This type of study was firstly conducted by [Jaffri et al., 2008], in which the authors assessed the quality of authors linkage with DBpedia in the 2006 DBLP dataset. By looking at the 49 most common author names, the results shows that 92% of these authors have incorrect publications affiliated to them, due to erroneous owl:sameAs assertions. In 2010, the authors of [Halpin et al., 2010] manually evaluated a sample of 250 owl:sameAs statements from a collection of 58.6M owl:sameAs links. This study shows that around 21% of the owl:sameAs assertions are incorrect, and should be replaced by a similarity or 'related to' relationships. In a follow up study [Halpin et al., 2015], the authors have showed that owl:sameAs assertions resulting from inference are more likely to be erroneous than randomly selected ones without inference. In another owl:sameAs quality analysis, the authors of [Hogan et al., 2012] manually evaluated 1K pairs occurring in the same equivalence classes, following the transitive closure of 3.7M distinct owl:sameAs triples. This evaluation shows that 2.8% of the pairs are different, and should not belong to the same equivalence class.

Discussion

These different and complementary studies have investigated several aspects of the identity use in the Web of Data. Firstly, they show that not all datasets are transitively linked by owl:sameAs assertions [Schmachtenberg et al., 2014], with each connected component consisting of clusters of densely connected datasets that cover similar topics [Ding et al., 2010b]. In addition, these studies show that owl:sameAs networks have a particular structure, often consisting of central IRIs connected to other peripheral ones [Ding et al., 2010b]. Studies that computed the owl:sameAs transitive closure shows that, on average, each real-world entity is represented by less than three IRIs in the Web of Data [Ding et al., 2010b, Hogan et al., 2011]. Finally, and in terms of the quality of these interlinks, these studies have confirmed the presence of a number of incorrect identity links in the Web of Data, with [Hogan et al., 2012] estimating the number of erroneous links to 2.8%, whilst [Halpin et al., 2010]'s evaluation suggests that around one out of five owl:sameAs links in the Web of Data is erroneous. However, and in comparison to the size of the Web of Data which contains dozens of billions of triples and hundreds of millions of owl:sameAs links, these studies are not representative enough. This absence of large scale and representative analyses is possibly due to the difficulty in finding and accessing identity links in the Web of Data. This issue has motivated several approaches to harvest the Web, and provide efficient access to these identity links

and/or their transitive closure. In the next section, we present these approaches, and investigate their importance in limiting the presence of incorrect identity links, and facilitating access to existing ones.

2.2 Identity Management Services

Identity management services share the common goal of helping users or applications to identify IRIs referring to the same real world entity, and distinguish similar labels referring to different real world entities. For instance, in order to avoid using a resource referring to the river of Niger, while intending in using one referring to the country Niger, one could benefit from such services for re-using an existing universal identifier that unambiguously refers to a certain real-world entity (e.g. the river of Niger). Such type of services have a more centralized vision for identity management in the Web of Data, in which each real-world entity is referenced by a single centralized IRI. On the other hand, one can make use of other types of identity management services to find all identifiers referring to the river of Niger, and discover additional descriptions. Such services can play an important role in enabling large scale identity analysis in the Web, implementing and optimising linked data queries in the presence of co-reference [Schlegel et al., 2014], and detecting erroneous identity assertions [de Melo, 2013, Cuzzola et al., 2015, Valdestilhas et al., 2017].

In the early days of the Web, it was originally conceived that resource identifiers would fall into two classes: locators (URLs) to identify resources by their locations in the context of a particular access protocol such as HTTP or FTP, and names (URNs). URNs [Mealling and Daniel, 1999], were supposed to be the standard for assigning location-independent, globally unique, and persistent identifiers to arbitrary subjects. Each identifier has a defined namespace that is registered with the Internet Assigned Numbers Authority (IANA). For instance, 'ISBN' is a registered namespace that unambiguously identifies any edition of a text-based monographic publication that is available to the public. For instance, *urn:isbn:0451450523* is a URN that identifies the book "The Last Unicorn", using the ISBN namespace. Because of the lack of a well-defined resolution mechanism, and the organizational hurdle of requiring registration with IANA, URNs are hardly used (a total of 47K URNs in the 2015 copy of the LOD, with only 73 registered[3] URN namespaces with IANA at the time of writing). Since 2005, the use of the terms URNs and URLs has been deprecated in technical standards in favour of the term Uniform Resource Identifier (URI), which encompasses both, and the term Internationalized Resource Identifier (IRI) which extends the URI character set that only supports ASCI encoding.

[3]https://www.iana.org/assignments/urn-namespaces/urn-namespaces.xhtml

A more recent proposal for a centrally managed naming service was proposed by [Bouquet et al., 2007]. This public entity name service (ENS), named Okkam[4], intends to establish a global digital space for publishing and managing information about entities. Every entity is uniquely identified with an unambiguous universal URI known as an OKKAM ID, with the idea of encouraging people to reuse these identifiers instead of creating new ones. Each OKKAM ID is matched to a set of existing identifiers (e.g. DBpedia and Wikidata IRIs), using several data linking algorithms that are available in the public entity name service hosted at `http://okkam.org`. For instance, the company 'Apple' has a profile with an Okkam ID[5], which is linked to other non-centrally managed IDs (e.g. `dbpedia/resource/Apple_Inc`). For each OKKAM entity, a set of attributes are collected and stored in the service for the purpose of finding and distinguishing entities from another. However, the public entity name service is no longer maintained, with no information on the number of existing entities, links, and the covered datasets.

Finally, [Glaser et al., 2009] introduced the Consistent Reference Service (CRS), that finds for a given IRI, the list of identifiers that belong to the same identity bundle. These identity bundles are the result of the transitive closure of a mix of identity and similarity relationships (such as `owl:sameAs`, `umbel:isLike`, `skos:closeMatch`, and `vocab:similarTo`). This service is based on 346M triples harvested from multiple RDF dumps and SPARQL endpoints, and hosted at `http://sameas.org`. This large collection of triples linking over 203M IRIs, and resulting in 62.6M identity bundles, has been the basis for many subsequent approaches that aim to detect erroneous identity links (e.g. [de Melo, 2013, Cuzzola et al., 2015, Valdestilhas et al., 2017]).

Discussion

Identity management services play an important role in facilitating the understanding and re-use of IRIs. However we believe that centralized naming authorities such as OKKAM, although they might be adopted within some dedicated domains and applications, they will be of limited use in the context of the Web. As acknowledged by its authors [Bouquet et al., 2007], encouraging people to adopt and accept such Entity Naming Systems would be challenging, as the idea of having to go through an authority in order to use a new name somewhat goes against the philosophy of the ad-hoc, and scale-free nature of the Web, where "anybody is able to say anything about anything". In addition, such systems can only be truly successful once sufficient added value over the use of non-centrally managed identifiers is provided, specifically in providing efficient and high-quality search results, and offering high coverage of

[4] As a variation of Occam's razor: "entities are not to be multiplied without necessity"
[5] eid-9bc2b9fd-cb41-4401-8204-6c8933010acf

real-world entities. Finally, centralizing all names into one system would raise many privacy and security concerns, in a time where the paradigm is shifting towards more decentralization of the Web [Verborgh et al., 2017].

The Consistent Reference Service proposed by [Glaser et al., 2009], is more adopted in Linked Data applications [de Melo, 2013, Cuzzola et al., 2015, Valdestilhas et al., 2017]. However, in its current architecture and status, it faces some limitations. Firstly, identity bundles in the `sameAs.org` service are the result of the transitive closure of a mix of identity and similarity relationships (such as `umbel:isLike` and `skos:exactMatch`). The system does not keep the original predicates, meaning that a user cannot identify if two terms in the same bundle are actually the same, similar or just closely related (e.g. `skos:closeMatch`). The presence of several identity and similarity relations, with different semantics, means that the overall closure is not semantically interpretable (e.g. can not be used by a DL reasoner for inferring new facts). In addition, since no service can guarantee the coverage of all the triples in the Web of Data, one way of ensuring better transparency would be by listing the exploited data sources. This would allow users to evaluate the pertinence of this data in their applications and contexts. The Consistent Reference Service does not provide such information.

2.3 Detection of Erroneous Identity Links

An important aspect of managing identity in the Web of Data is the detection of incorrectly asserted identity links. In order to detect such erroneous links, different kinds of information may be exploited: RDF triples related to the linked resources, domain knowledge that is described in the ontology or that is obtained from experts, or `owl:sameAs` network metrics. In this section, we present existing approaches that detect erroneous identity links, based on three –eventually overlapping– categories of approaches: inconsistency-based (2.3.2), content-based (2.3.3), and network-based approaches (2.3.4). Table 2.1 provides a summary of these approaches, stating their characteristics, requirements, and the data in which the experiments were conducted.

2.3.1 Evaluation Measures

An approach of erroneous link detection can be evaluated using the classic evaluation measures of precision, recall, and accuracy. In Table 2.1 we present these measures as reported in each paper, when available. These evaluation measures can be defined for the problem of detection of erroneous links as follows:

Precision. Represents the number of links classified by the approach as incorrect, and are indeed incorrect identity links (True Positives), over the total number of links classified as incorrect by the approach (True Positives + False Positives).

Recall. Represents the number of links classified by the approach as incorrect, and are indeed incorrect identity links (True Positives), over the total number of incorrect identity links existing in the dataset (True Positives + False Negatives).

Accuracy. Represents the number of links classified by the approach as incorrect, and are indeed incorrect identity links (True Positives), and the number of validated and actually correct identity links (True Negatives), over the total number of identity links classified as incorrect by the approach (True Positives + False Positives), and the total number of identity links validated as correct by the approach (True Negatives + False Negatives).

$$precision = \frac{TP}{TP + FP} \qquad recall = \frac{TP}{TP + FN}$$

$$accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

2.3.2 Inconsistency-based Detection Approaches

These approaches hypothesize that `owl:sameAs` links that lead to logical inconsistencies have higher chances of erroneousness than logically consistent `owl:sameAs`.

Conflicting `owl:sameAs` and `owl:differentFrom`

The first approach for detecting erroneous identity assertions in the Web of Data was introduced by [CudreMauroux et al., 2009], who presented idMesh: a probabilistic and decentralized framework for entity disambiguation. This approach hypothesizes that `owl:sameAs` and `owl:differentFrom` links published by trusted sources, are more likely to be correct than links published by untrustworthy ones. For initialising the sources' trust values, the approach relies on a reputation-based trust mechanisms from P2P networks, on online communities trust metrics, or on the used domains (e.g. closed domains such as `http://www.agroparistech.fr` get higher trust values). In case no information is available, a default 0.5 value is initialized for the source. The approach detects conflicting `owl:sameAs` and `owl:differentFrom` statements based on a graph-based constraint satisfaction problem that exploits the `owl:sameAs`

symmetry and transitivity. They resolve the detected conflicts based on the iteratively refined trustworthiness of the sources declaring the statements (i.e. creating an autocatalytic process where constraint-satisfaction helps discovering untrustworthy sources, and where trust management delivers in return more reasonable prior values for the links). The approach shows high accuracy (75 to 90%) in discovering the equivalence and non-equivalence relations between entities even when 90% of the sources are actually spammers feeding erroneous information. However, this type of approach requires the presence of a large number of `owl:differentFrom` statements, which is not the case in the Web of Data. In addition, scalability evaluation, only conducted on synthetic data, demonstrate a maximum scale involving 8,000 entities and 24,000 links, over 400 machines, focusing solely on network traffic and message exchange as opposed to time. The precision and recall are not reported.

Ontology Axioms Violation

[Hogan et al., 2012] introduced a scalable entity disambiguation approach based on detecting inconsistencies in the equality sets that result from the `owl:sameAs` equivalence closure. This approach detects inconsistent equality sets, by exploiting ten OWL 2 RL/RDF rules expressing the semantics of axioms such as *differentFrom, AsymmetricProperty, complementOf*. When resources causing inconsistencies are detected, they are separated into different seed equivalence classes, in which the approach assigns the remaining resources into one of the seed equivalence classes based on their minimum distance in the non-transitive equivalence class, or using in a case of tie, a concurrence score that is based on the pairs' shared inter- and intra- links. The authors have evaluated their approach on a set of 3.7M unique `owl:sameAs` triples derived from a corpus of 947M unique triples, crawled from 3.9M RDF/XML web-documents in 2010. From the resulting 2.8M equivalence classes, the approach detects only three types of inconsistencies in a total of 280 classes: 185 inconsistencies through disjoint classes, 94 through distinct literal values for inverse-functional properties, and one through *owl:differentFrom* assertions. On average, repairing an equivalence class requires its partition into 3.23 consistent partitions. After manually evaluating 503 pairs randomly chosen from the 280 inconsistent classes, the results show that 85% of the pairs that were separated from the same equivalence class are indeed different (i.e. precision), leading to the separation of 40% of the pairs evaluated as wrong by the judges (i.e. recall). This result shows that consistency does not imply correctness, with 60% of the pairs evaluated as different still belong to the same (now consistent) equivalence classes. Hence suggesting that the recall could be much lower than 40%, as the approach is not capable of detecting different pairs from the other 2.8M consistent equivalence classes. The total runtime of this approach is 2.35 hours.

Table 2.1: Overview of erroneous identity links detection approaches. The approaches are presented in chronological order, stating their type, their requirements, the dataset on which the experiments were conducted, and the reported results.

Approach	Type of Approach	Requirements	Evaluated Data	Results
[CudreMauroux et al., 2009]	Inconsistency-based	- Source Trustworthiness - Presence of owl:differentFrom statements	Synthetic graph with 8K entities, and 24K links, from 400 peers	- 75 to 90% accuracy (depending on the number of spammer sources)
[Hogan et al., 2012]	Inconsistency-based	Ontology Axioms (OWL 2 RL/RDF rules)	3.77M unique `owl:sameAs` from a 2010 crawl of 3.9M Web documents	- 85% precision - 40% recall (only 280 inconsistent classes out of 2.8M)
[Guéret et al., 2012]	Network Metrics	-	Silk sample of: 50 correct `owl:sameAs` 50 erroneous `owl:sameAs`	- 50% precision - 68% recall
[de Melo, 2013]	Inconsistency-based	UNA	- BTC2011: 3.4M `owl:sameAs` - sameAs.org: 22.4M `owl:sameAs` - BTC2011 + sameAs.org	No precision or recall evaluation
[Acosta et al., 2013]	Content-based (crowdsourcing)	Necessary descriptions for each resource	DBpedia-Freebase: 95 `owl:sameAs`	- 94% accuracy - 0% recall (higher recall for other interlinks)
[Papaleo et al., 2014]	Inconsistency-based and Content-based	- Ontology Axioms (disjoint-class, (inverse)functional, and/or local complete properties) - Ontology Mappings	344 `owl:sameAs` produced by 3 different linking tools (OAEI 2010)	- 37 to 88% precision - 75 to 100% recall (depending on the dataset)
[Paulheim, 2014]	Content-based (outlier detection)	-	Peel-DBpedia: 2,087 `owl:sameAs` DBTropes-DBpedia: 4,229 `owl:sameAs`	- 58 to 80% AUC - 50% F1-measure (no precision or recall evaluation)
[Cuzzola et al., 2015]	Content-based (natural language analysis)	Textual descriptions for each resource	sameas.org: 411 from 7,690 collected `owl:sameAs`	- 93% precision - 75% recall
[Valdestilhas et al., 2017]	Inconsistency-based	UNA	LinkLion: 19.2M `owl:sameAs`	No precision or recall evaluation
[Sarasua et al., 2017]	Network Metrics	-	65K `owl:sameAs` from the 2014 LOD crawl	No precision or recall evaluation

[Papaleo et al., 2014] introduced another inconsistency-based approach to invalidate identity statements. This approach firstly builds a contextual graph of a specified depth that describes each of the involved resources in a certain identity link. This contextual graph considers only the subpart of RDF descriptions that can be involved in conflicting statements: class disjointness, (inverse) functional properties and local complete properties. When the two concerned resources belong to heterogeneous sources, the approach requires the mapping of their properties. After building the contextual graphs, the Unit-resolution inference rule is applied until saturation to detect inconsistencies within these graphs. The evaluation of the approach was not based on a sample of existing `owl:sameAs` links in the LOD. The authors opted for three `owl:sameAs` datasets produced by three different linking tools in the context of the 2010 Ontology Alignment Evaluation Initiative (OAEI)[6], with a total of 344 links. The results show low precision in two datasets (37 and 42.3%) and high precision in the third one (88%), with a recall varying between 75 and 100%, depending on the dataset. Finally, the authors show that when applied after a linking tool, this invalidation approach can increase the tool's precision (from 3 to 25 percentage points). However, this approach requires expert knowledge, ontology axioms, ontology alignments and its scalability has not been evaluated.

Unique Name Assumption Violation

These approaches hypothesize that individual datasets preserve the Unique Name Assumption (UNA), and that violations of the UNA are indicative of erroneous identity links [de Melo, 2013, Valdestilhas et al., 2017]. The UNA indicates that two terms, with distinct IRIs in the same dataset, do not refer to the same real world entity.

[de Melo, 2013] creates undirected graphs from existing `owl:sameAs` links, then applies a linear program relaxation algorithm, that aims at deleting the minimal number of edges in order to ensure that the unique name constraint is no longer violated. This algorithm is applied separately on each connected component. For the evaluation of the approach, they have firstly considered the 2011 Billion Triple Challenge dataset containing 3.4M `owl:sameAs` links, that resulted into 1.3M equivalence classes (i.e. connected components). Then a 2011 dump of the `sameas.org` dataset that contains 22.4M `owl:sameAs`, resulting in 11.8M equivalence classes. Finally, a third graph consisting of the combination of both data collections, containing 34.4M `owl:sameAs`, that have resulted in 12.7M equivalence classes. On the latter graph, the approach have detected 519K distinct pairs that occur in the same equivalence class, and at the same time belong to the same dataset (UNA violation). For satisfying the UNA constraint, the approach removed 280K links, that represent in that con-

[6] `http://oaei.ontologymatching.org/2010/`

text the erroneous owl:sameAs statements. Meaning that on average each deleted link have caused 1.85 violations in this graph, while every deleted link in the *BTC2011* and sameas.org dataset have caused 4.24 and 1.53 violations on average, respectively. The total runtime of the approach is not stated.

[Valdestilhas et al., 2017] generate the equivalence classes based on an algorithm called *Union Find*. After generating the equivalence classes, and akin to [de Melo, 2013], this approach detects the IRIs which share the same equivalence class and at the same time share the same dataset. However, instead of deleting triples to ensure the non-violation of the unique name constraint, this approach ranks the erroneous candidates based on the number of detected resources with errors. It was applied to check which link discovery framework from the *LinkLion* linkset repository, containing 19.2M owl:sameAs links, has a better score. The results show that at least 13% of the owl:sameAs links are "erroneous", with *sameas.org* having the worst consistency, if we consider that the UNA is respected in the LOD. The approach is scalable, with a total runtime of 4.6 minutes.

The precision, recall and accuracy of both approaches have not been evaluated. Interestingly, [de Melo, 2013] claims that most of the unique name assumption violations stem from incorrect identity links, not from inadvertent duplicates (e.g. very few DBpedia IRIs with different names exist that describe exactly the same real world entity). Whilst in [Valdestilhas et al., 2017]'s manual analysis of a random sample of 100 errors, they show that 90% of the errors stem from duplications within the dataset, instead of referring to two different real world entities. These contradicting results leave many uncertainties on the effectiveness of the UNA assumption, within each dataset, for the task of detecting erroneous links.

2.3.3 Content-based Approaches

These approaches exploit the resources descriptions to identify incorrect owl:sameAs links, relying on the resources' type (i.e. rdf:type) and/or the presence of some properties (i.e. the list of instantiated properties) [Paulheim, 2014], or the property values [Acosta et al., 2013, Papaleo et al., 2014, Cuzzola et al., 2015].

[Acosta et al., 2013] looked into the use of crowdsourcing as a mean to handle data quality problems in DBpedia. The paper focuses on three categories of quality issues: (i) objects incorrectly or incompletely extracted, (ii) data types incorrectly extracted, and most importantly for this topic (iii) interlinking (e.g. owl:sameAs for linking to external data sources and dbr:wikiPageExternalLinks for linking to external Web sites). The

adopted methodology consists of firstly involving domain experts for finding and classifying incorrect triples, and verifying these classifications using the Amazon Mechanical Turk (MTurk). The experts have evaluated 24K triples, describing 521 distinct DBpedia resources. They flagged as incorrect a total of 1.5K triples, whilst stating each type of detected error. These triples were also evaluated by the paper's authors as a way to create a gold standard, and were sent to the MTurk crowd for verification. Surprisingly, and according to the gold standard, Linked Data experts showed a 15% precision in evaluating interlinks. More specifically, the experts have incorrectly invalidated all `owl:sameAs` statements (95 `owl:sameAs` in total, indicating a 0% precision). Checking the types of error signalled by the experts in this evaluation[7], one can see that all these `owl:sameAs` links were signalled by the same expert, stating the same error type as "Links to Freebase". The MTurk workers have correctly judged 62% of the interlinking statements using a 'first answer' approach, and 94% of them using a 'majority voting' approach. These results show that MTurk workers are more efficient in evaluting interlinks, in particularily using a 'majority voting' approach. In addition, these results show that finding and classifying incorrect interlinks is more complex than other types of errors (71% and 82% precision for object and datatypes values extraction errors, respectively). However, with the whole process taking around 25 days[8], this adapted crowdsourcing methodology shows little feasibility in the Web of Data.

[Paulheim, 2014] presented a multi-dimensional and scalable outlier detection approach for finding erroneous identity links. This work hypothesizes that identity links follow certain patterns, hence links that violate those patterns are erroneous. This approach represents each identity link as a feature vector using direct types, using all ingoing and outgoing properties, or a combination of both. For detecting outliers, 6 different methods were tested (e.g. k-NN global anomaly score, one-class support vector machines), using different parameters (10 different runs in total). Each method assign a score to each `owl:sameAs` indicating the likeliness of being an outlier. These methods were tested on two link sets: Peel Session-DBpedia (2,087 links) and DBTropes-DBpedia (4,229 links). The experiments show much better results on the first dataset in terms of AUC[9], and show that using only the type features works best. The maximum F1-measure obtained is 54%, which the authors state that it is mainly due to flagging up to 3/4 of all links as outliers (high recall value). The precision and recall are not reported. The approach is fast in most cases, depending on which outlier detection method is applied, with a runtime varying between seconds to 15 minutes.

[7] `https://docs.google.com/spreadsheets/d/15u3NjomX3nYF6OuMNU3w76yd5IWAcRcsTlHbCBLw6l8/edit#gid=0`

[8] three predefined weeks for the contest and 4 days for the MTurk workers

[9] area under the ROC curve: the probability of wrong links to get lower scores than correct ones

[Cuzzola et al., 2015] proposed the SCID approach, that hypothesizes that an `owl:sameAs` link between two resources that do not have similar textual descriptions is erroneous. This approach firstly calculates a similarity score between the IRIs involved in a given `owl:sameAs` link using the textual description associated to them (e.g., through the `rdfs:comment` property). For calculating the similarity score, the approach relies on the position and the relevance of each resource with respect to the associated DBpedia categories and then it employs this score to determine whether the identity link is valid or needs to be flagged for removal. The approach was tested on 411 `owl:sameAs` links, resulting from a data cleansing of an original 7,690 link dataset extracted from *sameas.org*. The experimental results show that this approach can correctly flag questionable identity assertions, attaining precisions as high as 100% with a 56% recall when the threshold is set at 0.2. For a reasonable precision versus recall trade-off, the authors suggest a 0.5 or 0.6 threshold where the precision is between 86 and 93% and the recall between 75 and 79%. However, this approach requires the presence of textual description in both resources, which explains the high number of discarded links from the original dataset. The evaluation was restricted on the qualitative part, without any mention on the method's scalability or the total runtime of the experiments.

2.3.4 Network-based Approaches

Some approaches have looked into the use of network metrics for evaluating the quality of `owl:sameAs` links.

[Guéret et al., 2012] introduced LINK-QA: an extensible framework for performing quality assessment on the Web of Data. This approach, hypothesizes that the quality of a `owl:sameAs` link can be determined by its impact on the network structure. This impact is measured using three classic network metrics (clustering coefficient, betweenness centrality, and degree) and two Linked Data-specific ones (`owl:sameAs` chains, and description richness). For instance, the measure of betweenness centrality is based on the idea that networks dominated by highly central nodes are more prone to critical failure in case those central nodes cease to operate or are renamed. Hence, a link's quality is calculated with respect to its impact in reducing the overall discrepancy among the centrality values of the nodes. The two Linked Data-specific measures hypothesize that the quality of an `owl:sameAs` statement is measured based on its impact in closing an open `owl:sameAs` chain, and its contribution in adding complementary descriptions to the identity statement subject from the target resource. The experiments were conducted on 100 known good and bad quality links created using the Silk mapping tool. These experiments demonstrated that the classic network metrics are insufficient for detecting the quality of a

link, while the two Linked Data specific ones proved more successful in distinguishing between correct and incorrect links. According to the authors, the demonstrated result of 50% precision and 68% recall is mainly due the small network sample that was chosen for the experiments. The authors claim that the approach is scalable and can be distributed, without stating the runtime of the experiments.

Finally, for evaluating an identity link's quality, [Sarasua et al., 2017] have extended the notion of description enrichment proposed by the previous approach. The approach hypothesizes that an inter-dataset link that extends the description of the entities is of higher quality. The authors propose a set of measures for analysing a link based on the resulted extension in classification, description, entity connectivity, data set connectivity and the increase in the number of vocabularies. The experiments were conducted on around 1 million links connecting 35 datasets from the 2014 LOD crawl. These links include 65K `owl:sameAs` statements, with the rest corresponding to classification and relationship links such as `rdf:type` and `rdfs:seeAlso`, respectively. The experiments solely show which types of links add the highest gain to the source entity, without evaluating the precision, recall, and accuracy of this approach in detecting incorrect links, neither stating the total runtime.

Discussion

It has now been broadly acknowledged that erroneous identity links are present in the Linked Open Data, and that additional efforts are needed in order to detect them. In this section we discuss the advantages and drawbacks of the presented approaches, according to the three following criteria:

Efficiency. An efficient approach is able to detect a large number of erroneous identity statements (i.e. high recall), without incorrectly classifying correct identity ones as erroneous (i.e. high precision).

Transparency. It is necessary to have approaches offering transparency to the community, by making their tools, experimental data, and their results publicly accessible. This will allow users to directly benefit from such approaches by discarding the links that were evaluated as incorrect during this approach, or only consider the ones that were validated as correct. In addition, and since probably no approach would single handedly resolve the identity links problem in the LOD, it is important to provide transparency for allowing other approaches to compare, and hopefully improve, their results. Table 2.2 presents the resources that were made available by each approach.

Feasibility on the LOD. According to the 4th Linked Data principle, the main importance of identity links is to link resources in the context of the Web of

Data, and allow applications to use these links and discover new things[10]. Hence, an important criteria is the feasibility of an approach in the context of the Linked Open Data, where approaches are expected to scale to hundreds of millions of triples, and where certain assumptions on the data can not be presumed.

Half of the here presented approaches have looked into inconsistency detection as a mean to detect erroneous identity links. Some of these approaches are based on axioms that can be declared in the ontology, mappings that can be detected between schemas, or conflicting statements (i.e. `owl:sameAs` with `owl:differentFrom`). However, [Hogan et al., 2012]'s evaluation suggests that consistency does not necessarily indicate correctness, showing that a large number of incorrect identity statements occur in consistent equivalence classes. In addition, these experiments show that such inconsistencies are not frequent in the LOD Cloud, with only 280 inconsistent classes detected from 2.8M equivalence classes (0.01%). This fact might have prompted other inconsistency-based approaches such as [CudreMauroux et al., 2009] and [Papaleo et al., 2014] to conduct their experiments on synthetic data and linksets, respectively. Nevertheless, and despite the low feasibility on the LOD, these approaches have showed promising results on the respective datasets in terms of accuracy and precision, with [CudreMauroux et al., 2009] reporting accuracy as high as 90%, [Hogan et al., 2012] reporting an 85% precision, and [Papaleo et al., 2014] reporting an 88% precision in one linkset. However, and as presented in Table 2.2, these approaches offer very little transparency, as we are solely able to access the public linkset used in [Papaleo et al., 2014]'s experiments.

Other types of approaches have looked into detecting inconsistencies by presuming the unique name assumption (UNA) [de Melo, 2013, Valdestilhas et al., 2017]. The experiments show contradicting results on whether the UNA is presumed in each dataset or not (with [de Melo, 2013] claiming that most UNA violations stem from incorrect identity links, whilst [Valdestilhas et al., 2017]'s analysis showing that 90% of UNA violations stem from duplications). With no evaluation of the precision, recall and accuracy of both approaches, these experiments leave many uncertainties on the effectiveness of the UNA for detecting erroneous identity links in the LOD.

Content-based approaches such as [Acosta et al., 2013] have looked into the use of crowdsourcing for handling data quality problems in the Web, including wrong interlinks. This approach shows good efficiency in terms of precision, and offers full transparency by testing their methodology on a public dataset, and providing access to their tool, results, and gold standard. However, and as expected, crowdsourcing approaches are not scalable, requiring around 25 days for inspecting a total of 521 distinct DBpedia resources. On the other hand, au-

[10]https://www.w3.org/DesignIssues/LinkedData.html

tomated content-based approaches such as [Cuzzola et al., 2015] have showed promising results by associating resources' textual descriptions with DBpedia categories for understanding the linked resources' meaning. Despite reporting recall numbers as high as 90%, the experiments suggest that recall is much lower in the context of the Web, as they were able to evaluate only 411 out of 7,690 `owl:sameAs` (due to a preliminary data cleansing that primarily discards resources with no textual descriptions). In addition, and since there is no mention of the total runtime of this approach, the feasibility of this approach on billions of RDF triples (since they also require additional triples than `owl:sameAs` links) has not been demonstrated. Other content-based approaches such as [Paulheim, 2014] have showed that resources' types can be exploited for detecting outlier identity links, with AUC as high as 80%, and an F1-measure of 50%. However, the experiments suggest low precisions, with the reported results showing that in certain cases, up to 3/4 of all links are flagged as outliers. In addition, the experiments show large differences between the reported results in each dataset (AUC dropping from 80% to 58% in the DPTropes dataset), indicating that such methods are highly dependant on how data are modelled. Finally, with the approach being tested on around 6K links, its feasibility in the LOD has not been demonstrated.

Finally, [Guéret et al., 2012] and [Sarasua et al., 2017] have looked into the use of network metrics for evaluating the quality of `owl:sameAs` links, without requiring any assumptions on the data. [Guéret et al., 2012]'s experiments on a sample of 100 links, show that classic network metrics are not efficient for evaluating the quality of an `owl:sameAs` link. The Linked Data-specific network metrics that are based on closing `owl:sameAs` chains, and enriching the target entity's descriptions have been proven to be slightly more effective. However, we believe that the latter measure, also adapted by [Sarasua et al., 2017], hypothesizing that `owl:sameAs` links which add more information to an entity are more useful, can not be successfully adapted to detect incorrect identity links in the LOD. For instance, an erroneous `owl:sameAs` linking an IRI referring to the river Niger to an IRI referring to the country Niger, will massively enrich the description of the former, whilst a true `owl:sameAs` assertion might barely enrich the object's description. With [Guéret et al., 2012]'s experiments conducted on 100 `owl:sameAs` links, and [Sarasua et al., 2017]'s precision, recall and accuracy not been evaluated, the feasibility of these measures in the LOD remain untested. However, by making their codes publicly available on the Web, these approaches enable further testing of these measures.

[11] `http://swse.deri.org/entity/`
[12] `https://github.com/cgueret/LinkedData-QA`
[13] `http://bit.ly/Linked-QA`
[14] `https://github.com/cgueret/LinkedData-QA/tree/master/reports`
[15] `https://km.aifb.kit.edu/projects/btc-2011/`
[16] `http://nl.dbpedia.org:8080/TripleCheckMate/`

Table 2.2: Transparency overview of each erroneous identity link detection approach.

Approach	Dataset	Tool	Results	Gold Standard
[CudreMauroux et al., 2009]	-	-	-	-
[Hogan et al., 2012]	-	-	-	Link not Working[11]
[Guéret et al., 2012]	File Dumps[12]	Source Code[13]	HTML Reports[14]	-
[de Melo, 2013]	BTC 2011[15]	-	-	-
[Acosta et al., 2013]	DBpedia[16]	Source Code[17]	- Campaign Results[18] - MTurk Results[19]	Authors Evaluation[20]
[Papaleo et al., 2014]	PR OAEI 2010[21]	-	-	-
[Paulheim, 2014]	- Peel Sessions[22] - DBTropes[23]	Workflow[24]	-	-
[Cuzzola et al., 2015]	-	One Function but Link not Working[25]	-	-
[Valdestilhas et al., 2017]	Link not Working[26]	Source Code[27]	-	100 Output Samples[28]
[Sarasua et al., 2017]	LOD crawl[29]	Source Code[30]	Box Plots[31]	-

2.4 Alternative Identity Links

Some approaches have proposed to represent and/or find alternative identity relations. In this section we present existing alternatives, which either come in the form of simple predicates representing weaker types of identity or similarity, or approaches introducing techniques for representing and detecting contextual identity.

[17] https://github.com/AKSW/TripleCheckMate/releases/tag/DBpediaCampaign
[18] http://nl.dbpedia.org:8080/TripleCheckMate/
[19] http://people.aifb.kit.edu/mac/DBpediaQualityAssessment/experiments.html
[20] http://people.aifb.kit.edu/mac/DBpediaQualityAssessment/experiments.html
[21] http://oaei.ontologymatching.org/2010/im/
[22] http://dbtune.org/bbc/peel/
[23] http://skipforward.opendfki.de/wiki/DBTropes
[24] https://dws.informatik.uni-mannheim.de/en/research/rapidminerlodextension/
[25] http://ls3.rnet.ryerson.ca/predicatefinder/category/
[26] https://www.dropbox.com/s/m24xoxzm0h60ywl/correct.tar.gz?dl=1
[27] https://github.com/firmao/CEDAL
[28] https://github.com/dice-group/CEDAL/blob/master/100Sample.tsv
[29] https://drive.google.com/file/d/0B3W6K8QxmfLnc3FwV1lZd1hVemM/view
[30] https://github.com/criscod/SeaStar
[31] https://github.com/criscod/SeaStar/tree/master/data/plots

2.4.1 Weak-Identity and Similarity Predicates

Some vocabularies acknowledged the abusive use of `owl:sameAs` and provided alternative similarity and identity links. We present in the following some alternative interlinking predicates:

`rdfs:seeAlso`: this property is not used to denote any identity relation, but is used to indicate a resource that might provide additional information about the subject resource. This relationship was heavily used in linking Friend of a Friend (FOAF) data alongside the property `foaf:knows`, prior to the rise of `owl:sameAs` [Ding et al., 2010a]. Despite not having well-defined semantics, this property could still be useful in linking closely related entities and datasets.

SKOS predicates: The Simple Knowledge Organization System (SKOS) [Miles and Bechhofer, 2009] is a common data model for sharing and linking knowledge organization systems via the Semantic Web. SKOS introduces three mapping properties that correspond to different types of `owl:sameAs` usage. Firstly, `skos:relatedMatch` is used to state an associative mapping link between two concepts. `skos:closeMatch` indicates that "two concepts are sufficiently similar that they can be used interchangeably in some applications". Finally `skos:exactMatch` indicates "a high degree of confidence that the concepts can be used interchangeably across a wide range of applications". Whilst the misuse of these mapping properties can have much less implications than the misuse of `owl:sameAs`, their use for linking concepts is limited due to their lack for well-defined contexts of use. For instance, `skos:relatedMatch` is highly ambiguous and could probably relate most the concepts of the Semantic Web (since everything is related to everything in some way). In addition, the applications (i.e. the contexts) where the concepts related by `skos:closeMatch` or `skos:exactMatch` can interchange are not defined, and are eventually subjective. However, their main limitation relies in the fact that these predicates can only be used for IRIs of type SKOS concept.

In addition, the UMBEL[32] vocabulary introduced predicates such as the symmetrical property `umbel:isLike` which is used "to assert an associative link between similar individuals who may or may not be identical, but are believed to be so". Vocab.org[33] introduced the property `vocab:similarTo` to be used when having two things that are not the `owl:sameAs` but are similar to a certain extent. [de Melo, 2013] introduced `lvont:nearlySameAs` and `lvont:somewhatSameAs`, two predicates for expressing near-identity in the Lexvo.org[34] vocabulary, with definitions explicitly left vague, "simply because similarity is a very vague notion". He also introduced `lvont:strictlySameAs`, a predicate which is declared formally

[32] `http://umbel.org`
[33] `http://vocab.org`
[34] `http://lexvo.org`

equivalent to owl:sameAs, but just introduced for the purpose of distinguishing strict identity use from the erroneous use of the latter. Finally, the schema.org vocabulary[35] includes the schema:sameAs property. However, the semantics of this property is substantially different from that of owl:sameAs. It states that two terms "are two pages with the same primary topic" and does not express equality.

Finally, [Halpin et al., 2010] proposed the Similarity Ontology (SO) in which they hierarchically represent 13 different similarity and identity predicates. This ontology includes owl:sameAs, rdfs:seeAlso, and the three previously described SKOS predicates. For formally defining their semantics, the authors have characterized the eight newly introduced predicates by reflexivity, transitivity and symmetry properties. The most specific predicate in this ontology is owl:sameAs, and the most general ones are so:claimsRelated and so:claimsSimilar. The predicates prefixed with the word claims express a subjective identity or similarity relation in which their validity depends on the (contextual) interpretation of the user. The most specific newly-introduced predicate is so:identical. This predicate follows the same definition as owl:sameAs in the sense that two IRIs linked by this predicate do refer to the same real world entity. However, and contrary to owl:sameAs, this predicate is referentially opaque and does not follow Leibniz's law. Meaning that properties ascribed to one IRI are not necessarily appropriate for the other, and can not be substituted. As an example of referential opacity, the authors state the case of social inappropriateness in using certain names, referring to the same real world entity, in certain contexts. However, and despite proposing several alternative semantics for the strict identity relationship, this approach does not tackle the problem on how the contexts, in which an identity link is valid, can be explicitly represented. Hence, no indications on which properties ascribed to one IRI, will be also inferred to its identical (or similar) IRI.

2.4.2 Contextual Identity

The standardized semantics of owl:sameAs can be thought of as instigating an implicit context that is characterized by all (possible) properties to have the same values for the linked resources. Weaker kinds of identity can be expressed by considering a subset of properties with respect to which two resources can be considered to be the same. At the moment, the way of encoding contexts on the Web is largely ad hoc, as contexts are often embedded in application programs, or implied by community agreement. The issue of deploying contexts in KR systems has been extensively studied in AI. For the introduction of contexts as formal objects, see [Loyola, 2007] for a survey. In the Semantic Web, explicit rep-

[35] https://schema.org

resentation of context has been a topic of discussion since its early days, where the variety and volume of the web poses a new set of challenges than the ones encountered in previous AI systems [Bouquet et al., 2003].

The earliest standardized approach for explicitly encoding contexts in RDF is called *reification*[36]. This standardized data structure allows assertions to be made about RDF triples. Such assertions are encoded as resources of type `rdf:Statement`, to which metadata (i.e. a context) can be annotated, but eventually requiring 4 triples to represent an RDF statement. Another technique to represent a context in the Semantic Web is the use of N-ary relations[37]. This model which was proposed to represent statements between more than two individuals, can also be used to annotate the statements themselves, hence adding contexts to relationships. In addition, named graphs [Carroll et al., 2005] which are mostly used for representing provenance, can also be used to assert the context in which a triple or a set of triples hold. [Nguyen et al., 2014] propose the creation of a special instance for every triple predicate for which we want to provide the context. This instance will be related to its more generic property using the `singletonPropertyOf` predicate. For instance, the singleton property `MarriedTo#1` for which you can specify the context (e.g. provenance, date, etc.) is `rdf:singletonPropertyOf` of the generic property `MarriedTo`. Finally, [Giménez-García et al., 2017] proposed NdFluents, a multi-dimension annotation ontology that provides temporal parts to the subject and object of the triple, that can be used for representing a context.

With several approaches focusing on representing contexts in the Semantic Web, a recent approach have focused on the specific issue of detecting and representing contextual identity. [Beek et al., 2016] propose an approach that allows the characterization of the context in which a `owl:sameAs` link is valid. A context is represented by a subset of properties for which two individuals must have the same values, with all the possible subsets of properties organized in a lattice using the set inclusion relation. For instance, two drugs having the same chemical structure, but produced by different companies, are identical in the context where the commercial supplier of the drugs is discarded (i.e. the context considers solely the property *chemicalStructure*).

Discussion

In this section, we have presented several alternative predicates that may replace the use of `owl:sameAs` in some situations. A big downside of most of these approaches is the lack of formal semantics. For example, `skos:exactMatch` indicates a high degree of confidence that the concepts can

[36] `https://www.w3.org/TR/rdf11-mt/`
[37] `https://www.w3.org/TR/swbp-n-aryRelations`

Table 2.3: Overview of the usage of alternative identity links in the LOD Cloud.

Property	**Triples**
`owl:sameAs`	558,943,116
`rdfs:seeAlso`	169,172,965
`skos:exactMatch`	566,137
`skos:closeMatch`	371,011
`umbel:isLike`	461,054
`vocab:similarTo`	283
`lvont:nearlySameAs`	3,067
`lvont:somewhatSameAs`	1
`lvont:strictlySameAs`	0

be used interchangeably across a wide range of information retrieval applications. Whether a degree of confidence is high (enough) is subjective, and the meaning of this relation even changes over time, because information is always evolving over time. Also, some proposed alternative properties do not denote equivalence relations, which means that they are of limited use in linking and reasoning. In addition, most of these approaches require data publishers to change their modelling practice, needing a lot of momentum in order to create new datasets, or to change existing ones in order to make use of these alternative properties. As a result, and as presented in Table 2.3, most of these proposals lack uptake and are only used in a handful of datasets.

The approach proposed by [Beek et al., 2016], that come up with a new context-dependent semantics for the `owl:sameAs` property have the benefit that it does not require existing modelling practices to be changed. However, this approach only considers properties describing an instance locally in the RDF graph (i.e. a path of length 1). Moreover, this representation of the contexts does not consider the classes of the ontology, and consequently does not allow to consider properties differently, according to each class of the ontology. In addition, given the large number of possible contexts in which two entities can be identical, this approach does not provide means for users to set certain constraints on the contexts for filtering irrelevant contexts. An example of such constraints can be indicating the necessary properties that should be present in a context, and indicating irrelevant properties that can be discarded in such identity contexts. This filtering process can massively reduce the complexity of calculating the identity contexts, and can facilitate the finding and use of the relevant ones. Finally, no practical approach was proposed for representing the

identity contexts using Semantic Web standards.

2.5 Conclusion

In this section, we have presented several efforts that aim at solving, or at least limiting, the "sameAs problem" at hand. We will now give a generalized overview of the current situation.

Identity management services play an important role in facilitating the understanding and re-use of IRIs, and enabling large-scale analysis of the identity usage in the Web. We believe that identity management services such as `sameas.org` will see more uptake over time, as they make it possible to use some of the benefits of linking to other datasets, while at the same time giving the user some control as to which datasets to link to (and which datasets not to link to). However, in their current status, these services are not able to provide a definite reliable solution in terms of resource coverage, and up-to-date support for acting as true enablers for identity analysis and query answering services. Given the importance of such identity management services, and the drawbacks of existing ones, we propose in Chapter 3 a new identity management service that considers the identified issues. This proposed identity service has enabled us to conduct several types of identity analysis, which are an order of magnitude larger than the ones presented in Section 2.1.

In complementary of facilitating access to the identity links asserted in the Web, there is an important need to evaluate their correctness. By validating correct identity links, and detecting erroneous ones, linked data applications can make use of the `owl:sameAs` semantics for inferring new facts and making more connections, with higher levels of certitude. This has led to the emergence of several approaches for detecting erroneous identity links, with a rate of almost one approach per year since the emergence of Linked Data. While there exist approaches that have high recall, ones that have high accuracy, ones that are scalable, ones with no assumptions on the data, ones that are applied to real-world datasets, and ones that could be efficiently used as complementary to linking tools, there is currently no approach that exhibits all these features. The discussion in Section 2.3.4 shows that an approach of detecting erroneous identity links that can be efficiently applied on the whole LOD Cloud has yet to emerge, with many of the existing ones either lacking scalability, or requiring assumptions that are not valid in the context of the Web. In addition to the feasibility issue, we believe that the lack of transparency by most approaches is another important drawback in this area. As a result, we find ourselves with many interesting techniques, with very little materialized results for other approaches to build on, or for users to deploy in real world applications. Given the necessity of such approaches, and considering their current drawbacks, we

propose in Chapter 4 a novel approach for detecting erroneous identity links in the Web, based solely on the `owl:sameAs` network's community structure.

Finally, and given the highly problematic notion of identity standardized in `owl:sameAs`, and the necessity in expressing weaker notions of identity in certain cases, many approaches have proposed alternative identity predicates. However, with the contexts in which two entities are identical being not explicitly defined, these proposed predicates have limited semantics. In addition, and as discussed in Section 2.4.2, [Beek et al., 2016]'s proposition for a contextualised semantics for `owl:sameAs` have several limits, mainly in terms of the contexts' expressiveness and relevance. Hence, given the current presented limitations, we propose in Chapter 5 a new contextual identity relation. This approach extends the notion of contexts proposed by [Beek et al., 2016], by defining contexts as sub-ontologies and not uniquely as a set of properties. This allows contexts in which the identity of two class instances holds to be globally represented (i.e. not only in terms of properties of path 1), and to be parametrized according to the different ontology classes. In addition, we propose an algorithm that automatically detects the contexts in which two class instances are identical, and can be guided by a set of semantic constraints provided by experts, for filtering irrelevant identity contexts.

CHAPTER 3
IDENTITY ANALYSIS AND MANAGEMENT SERVICE

This chapter is based on the following publication:

- Wouter Beek, Joe Raad, Jan Wielemaker, and Frank van Harmelen. "sameAs.cc: The Closure of 500M owl:sameAs statements". In *Extended Semantic Web Conference*, pages 65–80, 2018 (best resource paper award).

Identity management services represent an important aspect in solving the "sameAs problem", as they can facilitate the re-use and understanding of IRIs. For instance, one can use such services to clarify the meaning of a resource and prevent unwanted inferences by verifying its identical resources. Although such services can become big factors in limiting the problem at hand, in their current status, no service is able to provide a definite reliable solution in terms of semantic interpretability, data coverage, and up-to-date support. Even though applications of a LOD Cloud-wide identity service are beyond the scope of this chapter, there are many use-cases for such services:

Findability of backlinks. Since the Semantic Web does not allow backlinks to be followed (an architectural property it shares with the World Wide Web), it is only possible to follow outgoing `owl:sameAs` links but not incoming ones. An identity service retrieves all IRIs that are linked through `owl:sameAs` links, and thereby allows the full set of assertions about a given resource to be retrieved from across the LOD Cloud.

Query answering. A special case of the findability of links arises in distributed query answering over the LOD Cloud, which requires an overview of existing alignments between concepts and individuals [Joshi et al., 2012].

Query answering under entailment. When a SPARQL query is evaluated under OWL entailment, the query engine must follow a large number of `owl:sameAs` links in order to retrieve the full result set. With an identity service, a query engine can translate the terms in the query to an IRI that represents the set of identical terms under entailment, which allows a SPARQL query to be executed using solely a single identifier.

Ontology alignment. Some algorithms rely on the identity of the class individuals in order to automatically compute alignments at the conceptual level (i.e. class and properties equivalence and subsumption relationships). For instance, if two classes share the same set of individuals, or a set of individuals that are declared `owl:sameAs`, then there can be a strong presumption that these classes are equivalent [Euzenat et al., 2007]. The availability

of a large dataset of real-world identity links can help quantify the utility of existing alignment algorithms such as [Correndo et al., 2012].

This chapter introduces a new identity management service, and makes the following three contributions:

1. It presents the largest downloadable dataset of identity statements that have been gathered from the LOD Cloud to date, and its equivalence closure. The dataset and its closure are also exposed through a web service. Even though the dataset and closure are quite large, they can be stored on a USB stick and queried from a regular laptop.

2. It gives an in-depth analysis of this dataset, its closure, and its aggregation into datasets.

3. It presents an efficient approach for extracting and storing the identity statements, and calculating their equivalence closure.

The rest of this chapter is structured as follows. Section 3.1 describes the approach for calculating and storing the explicit and implicit identity relations, and the requirements it must satisfy. Section 3.2 presents the implementation and the experiments. Section 3.3 gives an analysis of some of the key properties of our dataset, and the use of identity links in the LOD Cloud. Section 3.4 describes the `sameas.cc` dataset and web service, and Section 3.5 concludes.

3.1 Approach

In this section we describe our approach for extracting, calculating, and storing the identity relations and their transitive closure. Our approach is composed of three main steps: (1) extracting the explicit `owl:sameAs` statements, (2) removing the unnecessary `owl:sameAs` statements for calculating the closure, and finally (3) calculating the closure by partitioning the `owl:sameAs` network into several identity sets. In this chapter, we refer to the calculation of the closure as the partitioning into identity sets, since the materialization of the closure will not be stored. The problem of calculating the equivalence closure can be defined as follows:

Let N denote the set of RDF nodes: the RDF terms (IRIs, literals, and blank nodes) that appear in the subject or object position of at least one, non-reflexive, `owl:sameAs` triple. A *partitioning* of N is a collection of non-empty and mutually disjoint subsets $N_k \subseteq N$ (called partition members) that together cover N.

In a network solely composed of N with their `owl:sameAs` statements, these partition members are called *equality sets*, and the terms belonging to the same equality set are called *identity sets*. According to the `owl:sameAs` semantics, all RDF terms belonging to the same identity set denotes the same real world entity: $\forall x, y \text{ with } x \in N_k, y \in N_k \rightarrow x = y$. In this work, we do not consider singleton identity sets, which are the result of terms that solely appear in reflexive `owl:sameAs` statements, and the result of terms which do not appear in any `owl:sameAs` statement.

In order to calculate the closure, each identity set should be closed under equivalence, while taking in consideration multiple dimensions of complexity:

The closure can be too large to store. In Section 3.3, we will see that the LOD Cloud contains identity sets with cardinality well over 100K. It is not feasible to store the materialization of each identity set since the space consumption of that approach is quadratic in the size of the identity set (e.g., the closure of an identity set of 100K terms contains 10B identity statements).

For this, we do not store the materialization of the closure, but store the identity sets themselves, which is only linear in terms of the size of the universe of discourse (i.e. the set N of RDF nodes).

$|N_k|$ can be too large to store. Even the number of elements within one identity set can be too large to store in memory. Since our calculation of the closure must have a low hardware footprint and must be future proof, we do not assume that every individual identity set is always small enough to fit in memory.

Datasets changes over time. We calculate the identity closure for a large snapshot of the LOD Cloud. Since datasets in the LOD cloud are constantly changing, and datasets are constantly added, our approach supports incremental updates of the closure, allowing for both additions and deletions, without having to recompute the entire closure.

3.1.1 Explicit Identity Network: Extraction

Given as input a data graph consisting of different directed relations between entities, the first step of our approach consists of extracting all the identity links existing in this graph.

Definition 1 (Data Graph) A data graph is a directed and labelled graph $G = (V, E, \Sigma_E, l_E)$. V is the set of nodes[1]. E is the set of node pairs or edges. Σ_E is the

[1]In RDF, nodes are terms that appear in the subject and/or object position of at least one triple (IRIs, literals, and blank nodes).

set of edge labels. $l_E : E \rightarrow 2^{\Sigma_E}$ is a function that assigns to each edge $\in E$ a set of labels belonging to Σ_E (with $l_E(e)$ representing the labels denoted to e).

From a given data graph G, we can extract the explicit identity network G_{ex} (definition 2), which is a directed labelled graph that only includes those edges whose labels include `owl:sameAs`.

Definition 2 (Explicit Identity Network) Given a graph $G = (V, E, \Sigma_E, l_E)$, the related explicit identity network $G_{ex} = (N, E_{ex})$ is the edge-induced subgraph $G[\{e \in E \,|\, \{\texttt{owl:sameAs}\} \subseteq l_E(e)\}]$. N is the set of terms that appear in the subject and/or object position of at least one `owl:sameAs` statement ($N \subseteq V$). E_{ex} is the set of node pairs or edges for which a statement $\langle x, \texttt{owl:sameAs}, y \rangle$ has been asserted in G ($E_{ex} \subseteq E$).

3.1.2 Explicit Identity Network: Compaction

Since `owl:sameAs` is reflexive, symmetric and transitive, the size of the input data can be significantly reduced prior to calculating the identity closure. We call this preparation step *compaction*. Assuming an alphabetic order $<$ on RDF terms, we can reduce the input for the closure algorithm to a more concise set of pairs: $\{(x, y) \,|\, e_{x,y} \wedge x < y\}$. In this step, reflexive and duplicate symmetric edges in the explicit identity network G_{ex} are discarded.

3.1.3 Implicit Identity Network: Closure

In this step, we partition the remaining terms N' after compaction into different identity sets. We will not store the materialization of the closure G_{im} (Definition 3), but only the identity sets themselves.

Definition 3 (Implicit Identity Network) Given the set of sorted pairs of the explicit identity network G_{ex}, the implicit identity network $G_{im} = (N', E_{im})$ is the closure under equivalence (reflexivity, symmetry and transitivity) of each equality set. N' denotes the set of RDF terms that appear in the subject and/or object position of at least one non-reflexive `owl:sameAs` statement.

The partition of N' into different identity sets consists of a map[2] from nodes to identity sets ($N' \mapsto \mathcal{P}(N)$). We present in the following our desired mapping design, and the proposed algorithm for partitioning N' into identity sets.

[2]Note that each term in N does indeed belong to a unique non-singleton identity set.

Mapping Design

In order to optimize for space, we do not want to store the same identity set multiple times. We illustrate this for the identity set $\{x_1, x_2, \ldots, x_n\}$, where $\mapsto$ denotes a functional mapping from keys to values:

$$
\begin{aligned}
&x_1 \mapsto \{x_1, x_2, \ldots, x_n\} \\
&x_2 \mapsto \{x_1, x_2, \ldots, x_n\} \\
&\ldots \\
&x_n \mapsto \{x_1, x_2, \ldots, x_n\}
\end{aligned}
$$

According to this design an identity set S is stored $|S|$ times. Instead, we want a design that uses natural numbers ($\mathbb{N}$) as (arbitrary) identifiers denoting identity sets, as follows:

$$
\begin{aligned}
&x_1 \mapsto 1 \\
&x_2 \mapsto 1 \\
&\ldots \\
&x_n \mapsto 1 \\
&1 \mapsto \{x_1, x_2 \ldots, x_n\}
\end{aligned}
$$

For this design we need two key/value indexes:

1. A mapping from each RDF term to the key (ID) of the unique identity set that it belongs to. $val : N \mapsto_v ID$.
2. A mapping from an identity set key (ID) to its corresponding identity set. $key : ID \mapsto_k \mathcal{P}(N)$.

Hence, $val(x)$ gives us the identity set ID of an RDF term x, and the composition $key(val(x))$ gives us the identity set of x.

Algorithm

For partitioning N' into different identity sets, we have designed an incremental algorithm that parses each sorted identity pair (x, y), representing the output of the explicit identity network compaction. The algorithm distinguishes between four cases:

Case 1. Neither x nor y occurs in any identity set. A new identity set identifier id is generated and assigned to both x and y:

$$x \mapsto_v id$$
$$y \mapsto_v id$$
$$id \mapsto_k \{x, y\}$$

Case 2. Only x already occurs in an identity set. In this case, the existing identity set of x is extended to contain y as well:

$$y \mapsto_v val(x)$$
$$val(x) \mapsto_k key(val(x)) \cup \{y\}$$

Case 3. Only y already occurs in an identity set. Similar to the previous case.

Case 4. x and y already occur, but in different identity sets. In this case one of the two keys is chosen and assigned to represent the union of the two identity sets:

$$val(x) \mapsto_k key(val(x)) \cup key(val(y))$$
$$(\forall y' \in key(val(y)))(y' \mapsto_v val(x))$$

This is the most costly step, especially when both identity sets are large, but it is also relatively rare, since the input pairs are sorted during the compacting stage. A further speedup is obtained by choosing to merge the smaller of the two sets into the larger one.

3.2 Implementation & Experiments

In this section, we describe the implementation and experiments of our approach on a large copy of the LOD Cloud. We firstly describe the dataset in which our experiments are based on (section 3.2.1). Then, we present the implementation and experiments of extracting (section 3.2.2), and compacting (section 3.2.3) the explicit identity network. Finally, we present the computation of the transitive closure of this large collection of extracted identity links (section 3.2.4). The overall workflow of the identity network extraction, compaction and closure is given in Figure 3.1.

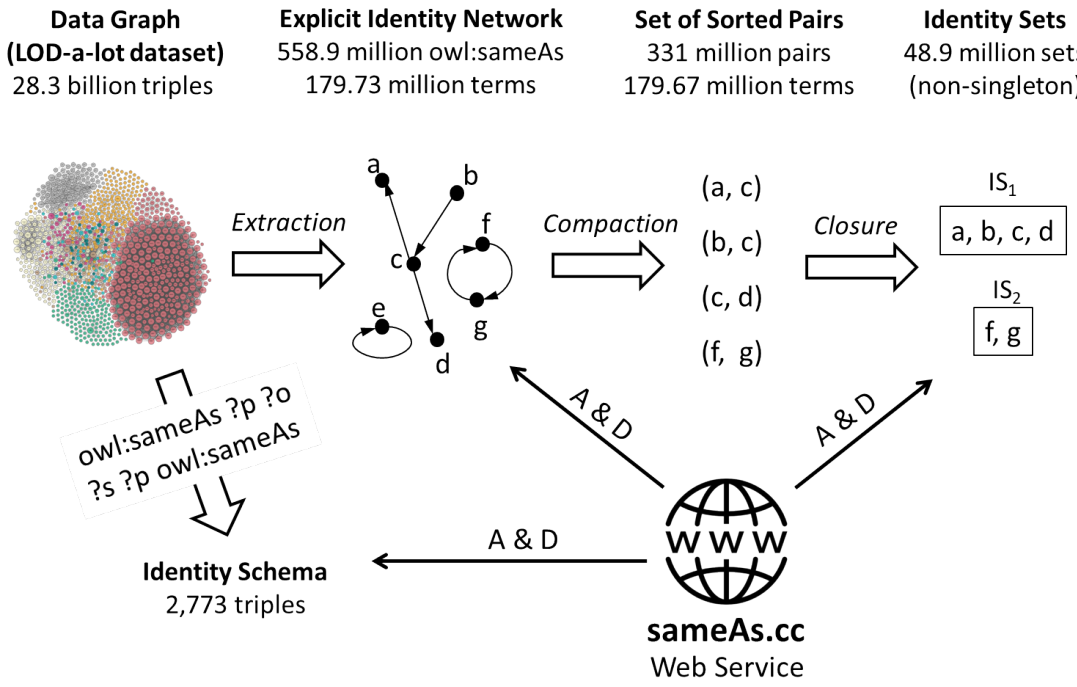


Figure 3.1: Workflow of the identity network extraction, compaction and closure. A& D indicates that the resource is freely accessible and downloadable at the sameAs.cc web service hosted at `http://sameas.cc`.

3.2.1 Data Graph

Our datasets and web service are based on the *LOD-a-lot*[3] dataset [Fernández et al., 2017]. *LOD-a-lot* proposes an effective way of packaging a standards compliant subset of the LOD Cloud into a ready-to-use file comprising data from the LOD Laundromat[4]. This dataset is exposed in a single HDT file that is 524 GB in size, and is publicly accessible (via an LDF interface) and downloadable (as HDT Dump). We briefly present its main components.

The LOD Laundromat [Beek et al., 2014] is a service that (i) crawls LOD datasets from Datahub[5] and other manually collected seeds; (ii) cleans the data by recovering syntax errors, removing duplicates, and replacing blank nodes with well-known IRIs[6]; and finally (iii) converts and republishes the datasets in the form of Gzipped N-Triples/N-Quads files. The current version (May 2015) is composed of 657,902 datasets and contains more than 38 billion triples (including between-dataset duplicates). Each

[3] `http://lod-a-lot.lod.labs.vu.nl`
[4] `http://lodlaundromat.org`
[5] `https://datahub.io`
[6] `https://www.w3.org/TR/rdf11-concepts/#section-skolemization`

dataset is serialized in Header-Dictionary Triples (HDT)[7] for download, and is also published as an Linked Data Fragment (LDF)[8] endpoint.

Header-Dictionary-Triples (HDT) [Fernández et al., 2013] is a binary compression format of RDF data. HDT keeps big datasets compressed for RDF preservation and sharing, and –at the same time– provides basic query functionality without prior decompression. An HDT-encoded dataset is composed by three logical components: (i) the header, which holds the datasets' metadata using plain RDF, allowing consumers to have an initial idea of key properties of the content before retrieving the whole dataset; (ii) the dictionary, which represents a catalog that assigns a mapping between resources and unique IDs; and finally (iii) the triples, which represents the RDF triples of the dataset as a set of tuples of three IDs.

Linked Data Fragments (LDF) [Verborgh et al., 2016] is a conceptual framework that provides a uniform view on all possible interfaces to RDF, by observing that each interface partitions a dataset into its own specific kind of fragments. It is aimed at improving the scalability and availability of SPARQL endpoints by minimizing server resource usage, and moving intelligence to the client. This allows the querying of simple triple patterns, in which its results are retrieved incrementally through pagination. As such, server load is minimized and large data collections can be exposed with high availability. Given that HDT provides fast, low-cost triple pattern resolution, LDF has been traditionally used in combination with HDT.

The resultant *LOD-a-lot* dataset, which represents our data graph (definition 1), contains more than 28.3 billion unique triples that represent a large copy of the LOD Cloud. This dataset contains more than 5 billion unique terms, related by more than 1.1 billion predicates.

3.2.2 Explicit Identity Network: Extraction

We use the *LOD-a-lot* HDT Dump to extract the explicit identity network (G_{ex}), and the HDT C++ library[9] to stream the result set of the following SPARQL query to a file. This process takes ~27 minutes:

[7] `http://rdfhdt.org/`
[8] `http://linkeddatafragments.org/`
[9] `https://github.com/rdfhdt/hdt-cpp`

```
select distinct ?s ?p ?o {
bind (owl:sameAs ?p)
?s ?p ?o }
```

The results of this query are unique (keyword `distinct`) and the projection (`?s ?p ?o`) returns triples instead of pairs, so that regular RDF tools for storage and querying can be used. The explicit identity assertions are stored in the order in which they are asserted by the original data publishers.

558.9 million triples that connect 179.73 million terms, are the result of this SPARQL query. These `owl:sameAs` triples are written to an N-Triples file, which is subsequently converted to an HDT file. The HDT creation process takes almost four hours using a single CPU core. The resulting HDT file is 4.5 GB in size, plus an additional 2.2 GB for the index file that is automatically generated upon first use.

3.2.3 Explicit Identity Network: Compaction

Since `owl:sameAs` is reflexive, symmetric and transitive, the size of the input data can be significantly reduced prior to calculating the identity closure, by discarding reflexive and duplicate symmetric edges in the explicit identity network G_{ex}. For this we use GNU sort unique.

GNU sort is faster when it is assigned multiple threads (`--parallel=4`), but this is not required. It also uses less memory when assigned a directory where it can create temporary files containing intermediate results (`-T $(tmp-dir)`). Since the exact order in which we sort is not required to follow natural language conventions, we explicitly disable lexicographic sorting of Unicode characters (setting environment variable `LC_ALL` to `C`, where sorting is done according to the byte values). We use process substitution to read from (`<(...)`) and write to (`>...`) a compressed GNU zip stream.

Figure 3.2 shows the significant impact of the compaction step, where the top node represents the full set of identity statements (G_{ex}), and the three bottom nodes represent the partition of G_{ex} into the following sub-relations: the reflexive pairs, the duplicate symmetric pairs, and the compacted explicit identity network that discards the two previous ones. The explicit identity network (G_{ex}) containing 558.9M edges and 179.73M nodes is reduced to a set of 331M sorted pairs and 179.67M nodes. As a result, we leave out ~2.8M reflexive edges and ~225M *duplicate* symmetric edges. We also leave out 67,261 nodes that only appear in such removed edges. The input size for the identity closure algorithm has been reduced by over 40%, taking 35 minutes on an SSD disk.

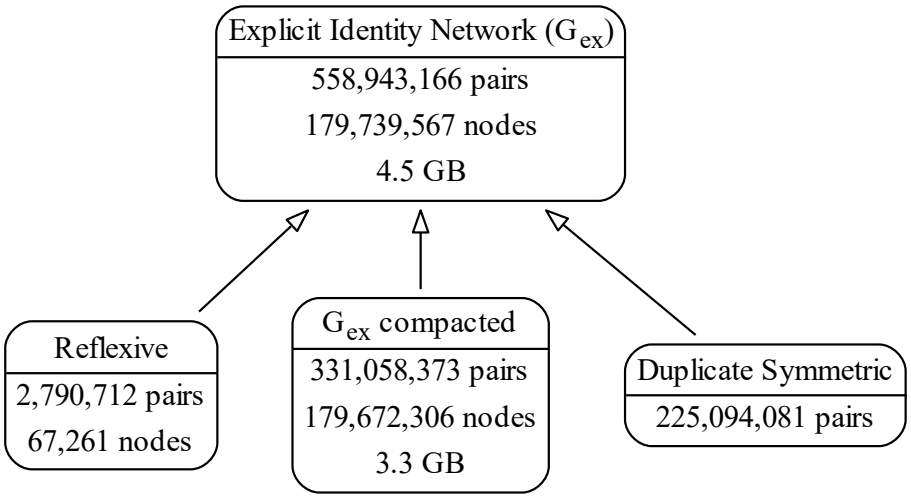


Figure 3.2: Overview of the explicit identity network compaction.

3.2.4 Implicit Identity Network: Closure

Now that we have a compacted version of G_{ex}, we calculate the identity closure that consists of a map from nodes to identity sets. In order to build an efficient implementation of this key-value scheme, we need a solution that (i) uses almost no memory and scales over an (SSD) disk, (ii) is able to store billions of key-value pairs, and (iii) allows such pairs to be added/removed dynamically over time. For this we use the RocksDB[10] persistent key-value store through a SWI Prolog API[11] that was designed for this purpose, allowing to simultaneously read from and write to the database. Since changes to the identity relation can be applied incrementally, the initial creation step only needs to be performed once.

The calculation of the identity closure takes just under 5 hours using 2 CPU cores on a regular laptop. The result is a 9.3GB on-disk RocksDB database: 2.7GB for mapping each term to an identity set ID ($N \mapsto_v ID$), and 6.6GB for mapping each identity set ID to its corresponding identity set ($ID \mapsto_k P(N)$).

[10] `https://rocksdb.org`
[11] `https://github.com/JanWielemaker/rocksdb`.

3.3 Data analytics

In this section we perform several analyses over the dataset created in the experiments described in the previous section. In what follows, we will use the following RDF prefixes for brevity:

owl: `http://www.w3.org/2002/07/owl#`
rdf: `http://www.w3.org/1999/02/22-rdf-syntax-ns#`
rdfs: `http://www.w3.org/2000/01/rdf-schema#`
xsd: `http://www.w3.org/2001/XMLSchema#`
dbr: `http://dbpedia.org/resource/`

3.3.1 Explicit Identity Network Analysis

Firstly, we provide some analysis over the size of the explicit identity network, and specifically over the number and type of terms that occur in `owl:sameAs` statements. Then, we analyse the number of outgoing and incoming `owl:sameAs` statements that occur by term. Finally, we analyse how the number of `owl:sameAs` statements are distributed over datasets, giving a high level impression on datasets that act as domain-specific naming authorities.

Terms in the Explicit Identity Network (G_{ex})

The explicit identity network contains 179,739,567 unique terms, representing the total number of terms that occur in `owl:sameAs` assertions in the *LOD-a-lot* dataset. As to be expected, the vast majority of these are IRIs (175,078,015 or 97.41%). Only a few literals are involved in the identity relation (3,583,673 or 1.99%), and even fewer blank nodes (1,077,847 or 0.60%). The majority of IRIs contain the HTTP(S) scheme (174,995,686 or 97.36.). Figure 3.3 gives an overview of the terms involved in the explicit identity network.

Statements in the Explicit Identity Network (G_{ex})

The LOD Laundromat corpus contains a total of 558,943,116 `owl:sameAs` statements. Based on the 2011 Billion Triple Challenge dataset, the authors of [Wang et al., 2014] observed that the number of `owl:sameAs` statements per

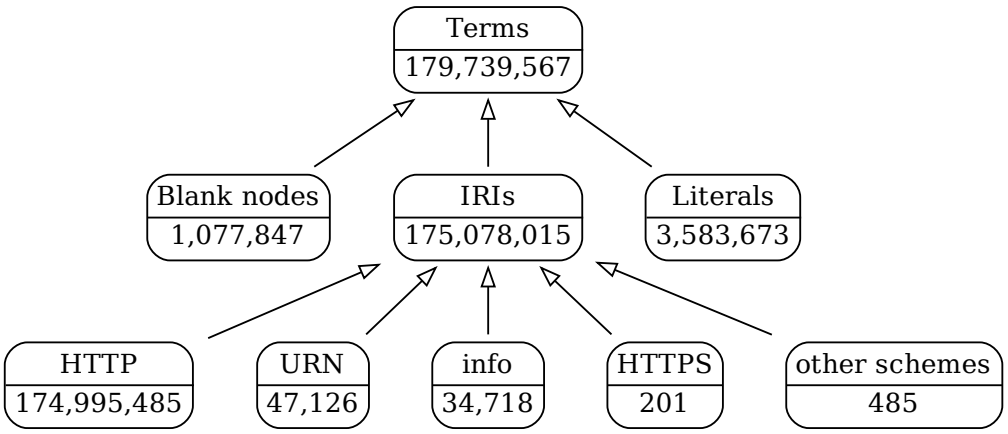


Figure 3.3: Overview of the terms involved in the explicit identity network. Blank nodes, IRIs and literals do not sum to the number of terms exactly, because there are 32 terms that are neither (they are syntactically malformed IRIs).

term approximated a power-law distribution[12] with coefficient -2.528. In contrast to this, we find that in the 2015 LOD Laundromat corpus, although most terms do appear in a small number of statements, this distribution does not display a power-law distribution. The patterns for the distribution of **incoming arcs** (identity statements where the term appears in the object position) and the distribution of **outgoing arcs**, (identity statement where the term appears in the subject position) all follow a similar distribution pattern (Figure 3.4).

Dataset Relations in the Explicit Identity Network (G_{ex})

Because `owl:sameAs` is the most frequently used predicate to link between datasets [Schmachtenberg et al., 2014], we also analysed G_{ex} at the aggregation level of links[13] between datasets. Unfortunately, there is no formal definition of what a dataset is. Since most of the terms involved in `owl:sameAs` assertions are HTTP(S) IRIs (Section 3.3.1), the notion of a *namespace* is a good proxy. According to the RDF 1.1 standard, IRIs belong to the same namespace if they have "a common substring". Obviously not every common substring counts as a namespace, otherwise all IRIs would be in the same namespace. A good pragmatic choice for a namespace-denoting substring is to take the prefix of HTTP(S) IRIs that ends with the *host name*. The host name is part of every syntactically valid HTTP(S) IRI, and denotes a physical machine that is located on the Internet.

Using this interpretation, Figure 3.5 shows that the number of terms occur-

[12] $p(x) = \alpha x^{-\beta}$ where $\beta = 2.528$

[13] In this section, a *link* is an `owl:sameAs` statement between terms that belong to different datasets

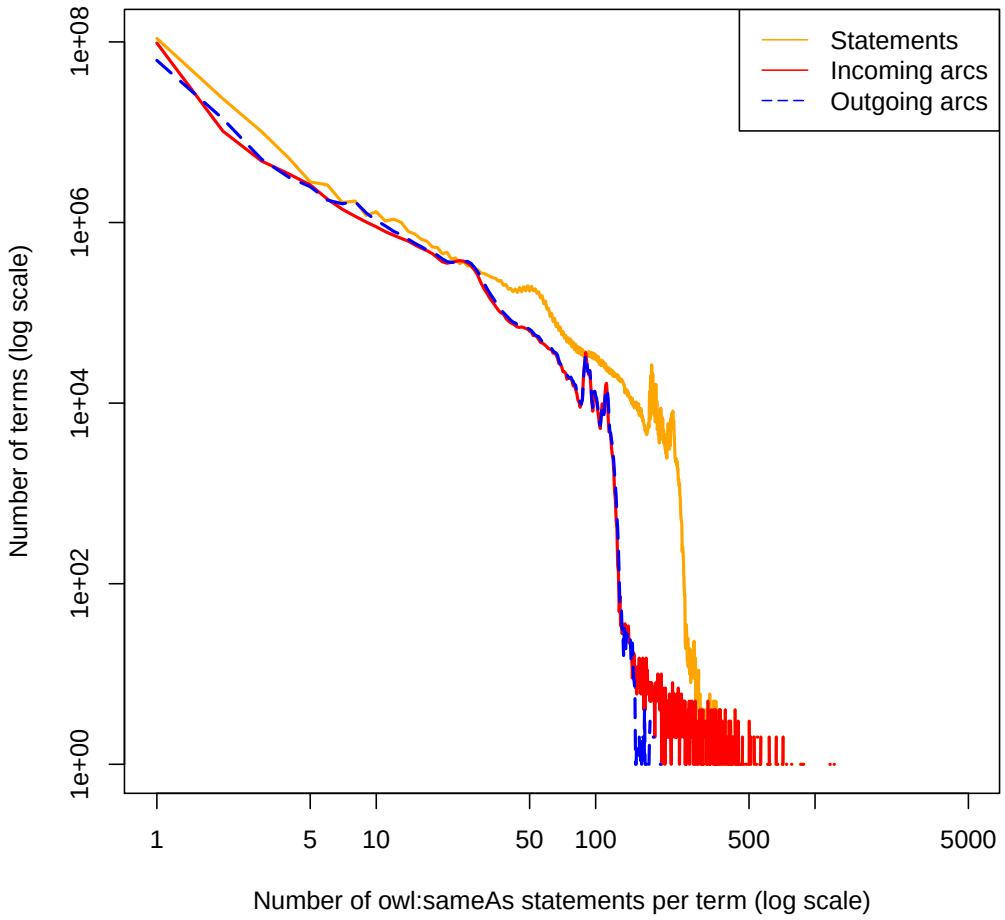


Figure 3.4: The distribution of `owl:sameAs` statements per term.

ing in `owl:sameAs` links is very unevenly distributed over namespaces (which we use as proxies of datasets).

For each namespace we calculated the number of *incoming* and *outgoing* links (statements whose subject, respectively object, term is in a different namespace.) The remaining statements are *internal edges* (they either have two HTTP(S) IRIs that belong to the same namespace, or they have at least one node that is not an HTTP(S) IRI (i.e., either a blank node or a literal). Figure 3.6 shows the distribution of internal edges, incoming links, and outgoing links over namespaces. While the majority of namespaces have incoming links, far fewer namespaces have outgoing links. This means that a relatively small number of namespaces is linking to a relatively large number of them. These namespaces are responsible for interlinking in the LOD Cloud. Finally, an even smaller number of namespaces have internal `owl:sameAs` edges. This means that most namespaces only use identity statements for linking to other datasets, but not for equating dataset-internal resources, suggesting that most datasets enforce the Unique Name Assumption internally.

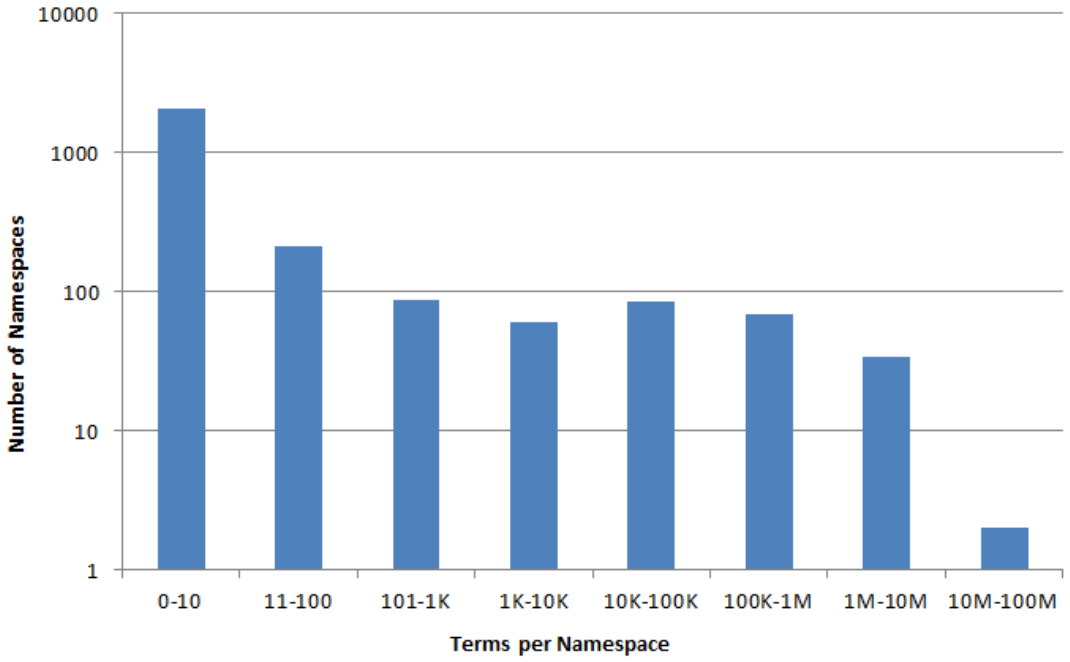


Figure 3.5: The number of terms in identity links by namespace.

To give a high level impression, we have visualised the entire identity-graph at namespace level in Figure 3.7. This graph contains 2,618 host-based namespaces/datasets, that are connected through 10,791 edges, and consists of 142 components. The large black cluster at the bottom of the figure is the densely interconnected set of multilingual variants of `dbpedia.org`, with the two high centrality nodes for `dbpedia.org` and `freebase.com` clearly visible just above the black cluster. The figure shows that there exist high-centrality nodes that act as domain-specific naming authorities/hubs. For example, the central node in the large top cluster is `www.bibsonomy.org`, which links to a large number of bibliographic datasets. A similar role is fulfilled by `geonames.org`, for interlinking geographic datasets; `bio2rdf.org`, for interlinking biochemistry datasets; and `revyu.com` (appearing at the right hand-side of the figure), for interlinking datasets that contain online reviews. A high-resolution version of this figure, together with textual namespace labels, is available at `https://sameas.cc/explicit/img`.

3.3.2 Implicit Identity Network Analysis

We provide some analysis over the size of the implicit identity network. Specifically, we analyse the terms that occurs in non-reflexive `owl:sameAs` statements and the resulting identity sets. Then we calculate the number of necessary

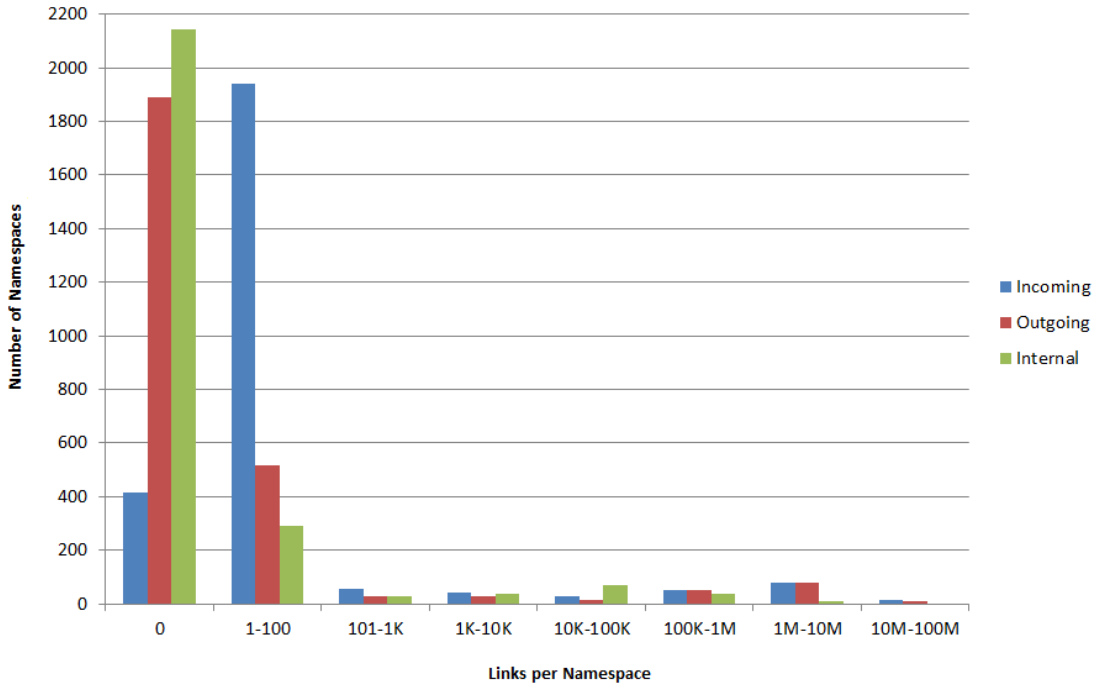


Figure 3.6: The distribution of internal edges, incoming links, and outgoing links by namespace.

`owl:sameAs` that would be needed in order to express the full materialization of G_{im}, and the minimal number of identity statements that would result in the same closure.

Terms in the Implicit Identity Network (G_{im})

The number of unique terms in G_{im} is 179,672,306. This is less than the number of unique terms in G_{ex} (179,739,567), because 67,261 terms (or 0.037%) *only* appear in reflexive `owl:sameAs` assertions.

Identity sets of the Implicit Identity Network (G_{im})

The number of identity sets is 48,999,148. Since reflexive statements were discarded during the compaction phase, all these identity sets are non-singleton. The *LOD-a-lot* file, from which we extract G_{ex}, contains 5,093,948,017 unique terms. This means that there are 5,044,948,869 singleton identity sets in the LOD. Figure 3.8 shows that the distribution of identity set size is very uneven and fits a power law with exponent 3.3 ±0.04. The majority of non-singleton identity sets (31,337,556 sets; 63.96%) contain only two terms. There are relatively few large identity sets, with the largest one having a cardinality of 177,794.

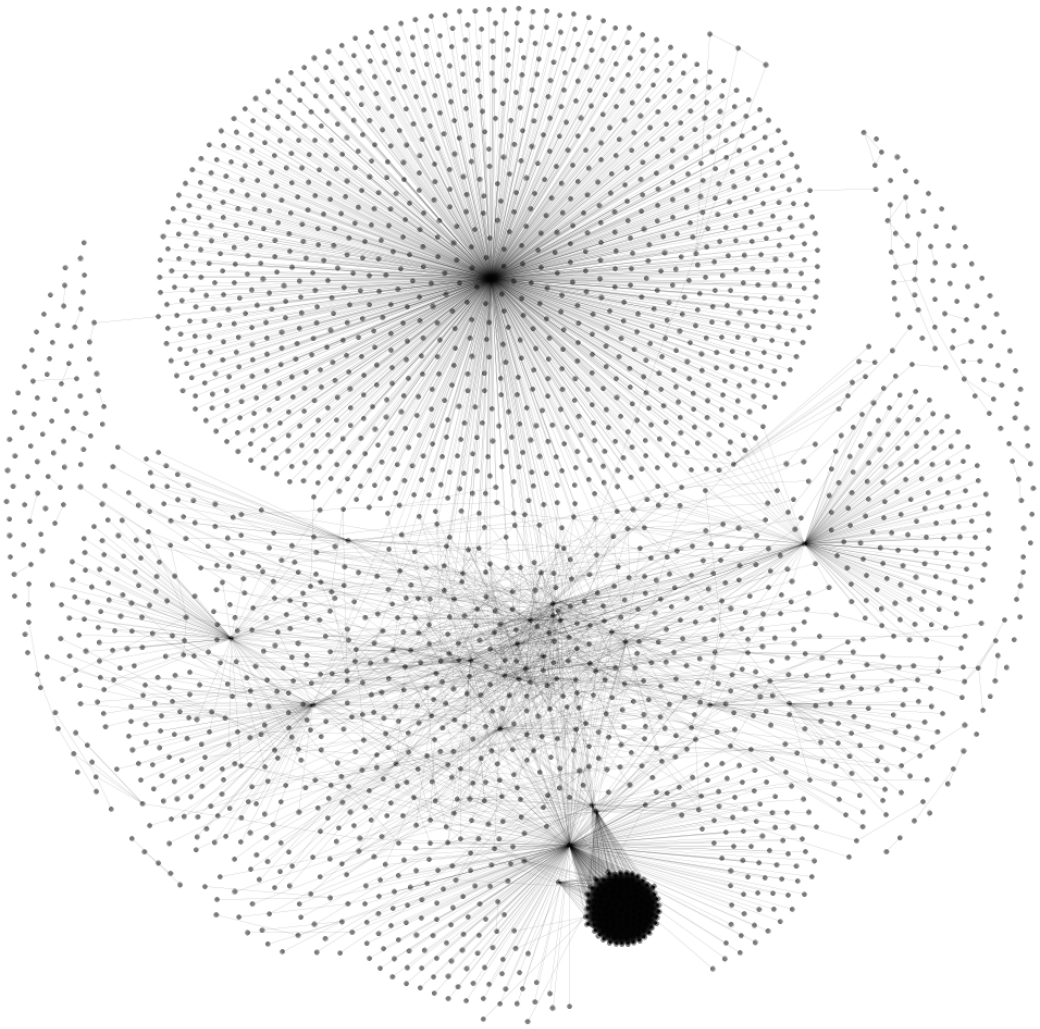


Figure 3.7: All inter-dataset links in the LOD Cloud. Thicker edges represent more identity links. The full diagram is available at `https://sameas.cc/explicit/img`.

Edges in the Implicit Identity Network (G_{im})

We want to calculate the number of necessary `owl:sameAs` that would be needed in order to express the full materialization of G_{im}. This calculation requires us to query and stream through the full RocksDB closure index, and therefore gives a good indication of the processing time required for running large-scale jobs over the `sameas.cc` dataset. The calculation (i) retrieves all identity sets, (ii) calculates their cardinality, and (iii) sums the squares of the cardinalities. This operation takes only 55.6 seconds and shows that the materialization consists of 35,201,120,188 `owl:sameAs` statements. Meaning that in

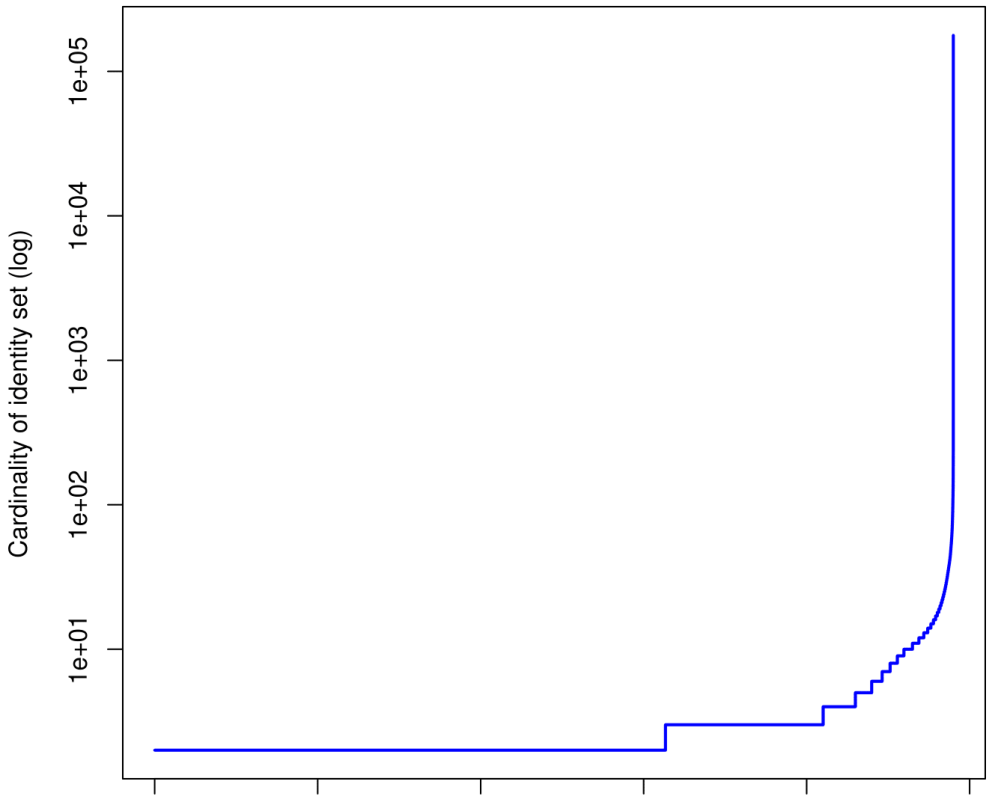


Figure 3.8: The distribution of identity set cardinality in G_{im}. The x-axis lists all 48,999,148 non-singleton identity sets.

case a full materialization of G_{im} is required, this would at least double the number of triples of the LOD-a-lot dataset. Notice that almost 90% (or 31,610,706,436 statements) of the materialization is contributed by the single largest identity set (i.e. with a cardinality of 177,794).

For further analysis, we want to calculate the minimal number of identity statements that would result in the same closure. We call such a minimal identity relation *a kernel*, and calculate it as the number of terms whose equivalence set is not a singleton set, minus the number of non-singleton identity sets. The kernel identity relation for G_{im} consists of 130,673,158 statements (or 0.37% of G_{im}). This also means that 76.6% of the explicit identity statements (G_{ex}) can be removed from the LOD-a-lot dataset, without any implication on the closure.

3.3.3 Schema Assertions About Identity

In this section we observe assertions in which the IRI `owl:sameAs` is in the subject or object position. There are 2,773 assertions about `owl:sameAs` that extend the schema as defined in the OWL vocabulary in interesting ways. The dataset is available at `https://sameas.cc/schema`. We observe the following kinds of schema extensions:

Super-properties of `owl:sameAs` As indicated in [Halpin et al., 2010], there is a need for properties that are weaker than `owl:sameAs` that express different shades of similarity and relatedness:

```
s: owl:sameAs
p: rdfs:subPropertyOf
o: <http://lexvo.org/ontology#nearlySameAs> .
```

However, some super-property assertions introduce semantic incoherences. For instance, since identity is the strongest equivalence relation, it does not make sense to assert new and specific *identity* relations that are super-properties of it. The following statement introduces the semantic incoherence that everything is an individual:

```
s: owl:sameAs
p: rdfs:subPropertyOf
o: owl:sameIndividualAs .
```

Sub-properties of `owl:sameAs` Several datasets introduce sub-properties of `owl:sameAs`, i.e., strengthening of the identity relation, without a clear use case. Our hypothesis is that these datasets intend to *weaken* the `owl:sameAs` property instead, since there are many use cases for weaker forms of similarity, relatedness, and context-dependent identity. For example:

```
s: <http://www.bbc.co.uk/ontologies/coreconcepts/sameAs>
p: rdfs:subPropertyOf
o: owl:sameAs .
```

Domain/range declarations As observed earlier by [Hogan et al., 2010], the intersection-based semantics of `rdfs:domain` and `rdfs:range` is often not followed. The following classes are asserted as the domain of `owl:sameAs`, effectively stating that all resources are both legal entities, anniversaries, strings, etc.

```
s: owl:sameAs
   p: rdfs:domain
         o: <http://govwild.org/0.6/GWOntology.rdf#LegalEntity> ,
         o: <http://s.opencalais.com/1/type/em/e/Anniversary> ;
```

```
        p: rdfs:range
             o: xsd:string .
```

Properties identical to `owl:sameAs` Several datasets mint alternative names for `owl:sameAs`, e.g.:

```
s: <http://rhm.cdepot.net/xml/#is>
p: owl:sameAs
o: owl:sameAs .

s: <http://sw.opencyc.org/concept/Mx4robv6phbFQdiM86Z2jmH52g>
p: owl:sameAs
o: owl:sameAs .
```

3.4 Dataset & Web Service

In this section, we present both the `sameas.cc` dataset and Web service.

3.4.1 Dataset

The `sameas.cc` dataset is available at `https://sameas.cc` and consists of the following components:

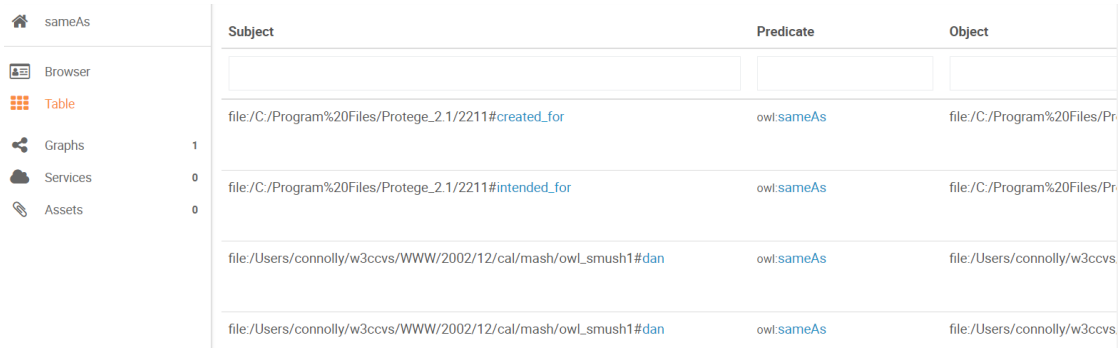


Figure 3.9: Screenshot of the *`sameas.cc` Triple Pattern API*. The screenshot shows 4 out of the 558,943,116 `owl:sameAs` statements existing in the dataset.

The Explicit Identity Dataset (G_{ex}) can be browsed online, queried for Triple Patterns, and downloaded as N-Triples and HDT.

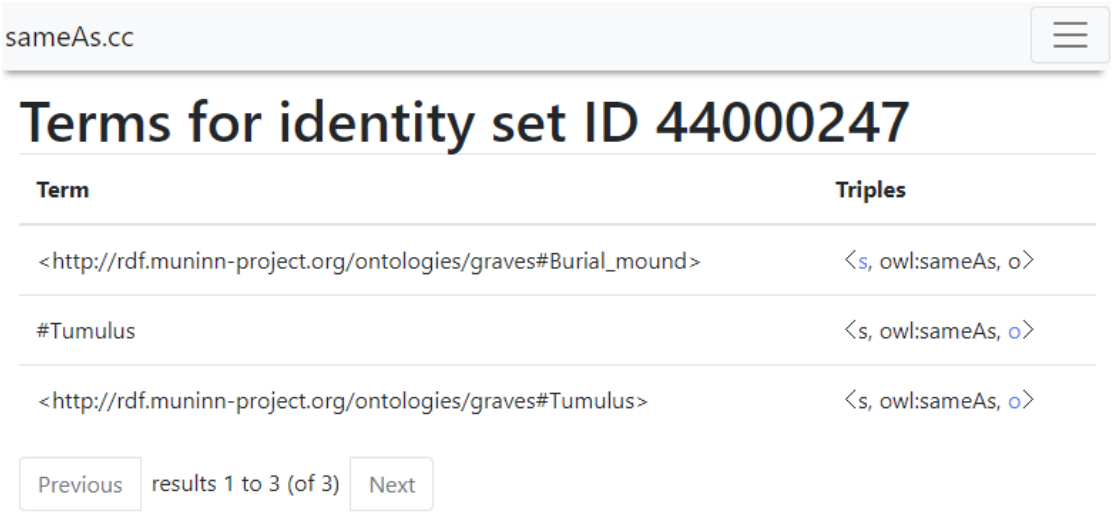


Figure 3.10: Screenshot of the `sameas.cc` *Identity Sets API*. The screenshot shows the little known fact that tumulus is a synonym for burial mound.

The Implicit Identity Dataset (G_{im}) is published as a downloadable snapshot of the RocksDB index (instead of a materialized RDF file). When RocksDB is installed, this snapshot can be queried locally.

The Identity Schema can be browsed online, queried for Triple Patterns, and downloaded in N-Triples, and HDT.

3.4.2 Web Service

The `sameas.cc` web service[14] consists of the following components:

Triple Pattern API. The explicit identity relation web service (`https://sameas.cc/explicit/tp`) allows all `owl:sameAs` assertions to be queried with Triple Patterns. Queries are expressed through (combinations of) the HTTP query parameters `subject`, `predicate`, and `object`. Figure 3.9 presents 4 out of the 558,943,116 `owl:sameAs` statements existing in the dataset.

Closure API. The implicit identity relation can be queried through the following URI paths:

`https://sameas.cc/id` Enumerates all identity set IDs. Each member of the identity closure is assigned such a unique ID.

[14]code is available at `https://github.com/wouterbeek/SameAs-Server`.

https://sameas.cc/id?term=dbr:Albert_Einstein Returns the ID of the identity set to which the given RDF term belongs.

https://sameas.cc/term Enumerates all RDF terms that appear in the identity relation.

https://sameas.cc/term?id=44000247 Enumerates only the RDF terms that appear in the identity set with ID 44000247 as key. Figure 3.10 presents the results of this request.

We deliberately expose the internal key-value mechanism explained in Section 3.2.4 to the users of the `sameas.cc` Closure API. The typical use case that we envision is one in which (i) terms are replaced by identity set identifiers, (ii) efficient computation is performed with the much more compact identifiers, and (iii) only when computation is done and end results need to be displayed are identifiers translated back to the potentially many terms that make up the respective identity sets.

3.5 Conclusion

In this chapter we have presented `sameas.cc`, the largest and most versatile dataset and web service of semantic identity links to date. The resource that we provide includes the largest collection of `owl:sameAs` assertions and the closure calculated over it. Even though the datasets are large, the algorithms and data-structures we deployed ensure that the resources can be stored on and queried from a regular laptop. In addition to the dataset and web services themselves, we have also presented several analytics over the data, including calculations of the size of the identity relation, its closure and its kernel, and various distributions. The analyses we presented in this chapter is an order of magnitude larger than previous conducted identity analyses. Finally, these presented resources can be freely downloaded and queried from our identity management service hosted at `http://sameas.cc`, and can be used by other researchers in order to uncover aspects of identity that have not been studied before.

In contrary to this work's main predecessor [Glaser et al., 2009], by solely considering `owl:sameAs` statements, we have provided –in theory– semantically interpretable identity sets that can be used for instance by a DL reasoner in order to infer new facts. In addition, and since the explicit identity statements are extracted from the LOD-a-lot dataset, we can provide users with provenance information on which dataset is covered in `sameas.cc`, through the LOD Laundromat service. Table 3.1 shows an overview of the two datasets.

Looking at some of our resulting identity sets and their IRIs descriptions, it is clear that some of these sets contain IRIs that do not refer to the same real

Table 3.1: Overview of sameas.org and sameas.cc.

	sameas.org	**sameas.cc**
#Terms	203,953,936	179,739,567
#Statements	346,425,685	558,943,116
#owl:sameAs	Unknown	558,943,116
#Partitions	62,591,808	48,999,148
#Identity Sets	Unknown	48,999,148

world entity. Since these identity sets are solely based on the transitive closure of owl:sameAs links, this interpretation indicates that identity is indeed misused in the Web [Jaffri et al., 2008, Ding et al., 2010a, Halpin et al., 2010]. In the next chapter, we analyse some of the resulted identity sets, and present an approach for detecting the erroneous owl:sameAs assertions causing this equivalence mash-up.

CHAPTER 4
ERRONEOUS IDENTITY LINK DETECTION

This chapter is based on the following publications:

- Joe Raad, Wouter Beek, Nathalie Pernelle, Fatiha Saïs and Frank van Harmelen. "Détection de liens d'identité erronés en utilisant la détection de communautés dans les graphes d'identité". In *Revue des Sciences et Technologies de l'Information-Série ISI: Ingénierie des Systèmes d'Information*, 23(3-4):95–118, 2018.
- Joe Raad, Wouter Beek, Frank van Harmelen, Nathalie Pernelle, and Fatiha Saïs. "Detecting Erroneous Identity Links on the Web using Network Metrics". In *International Semantic Web Conference*, pages 391–407, 2018.

It has now been broadly acknowledged that erroneous identity links are present in the Semantic Web. The presence of such links poses an important threat on the quality of the data on the web, specifically when reasoning is intended. This issue has led to the emergence of several approaches over the recent years that aim at detecting these links that violate the strict logical semantics of `owl:sameAs`. As presented in Section 2.3.4, while there are approaches that have high recall, ones that have high accuracy, ones that are scalable, ones that are applied to real-world datasets, and ones that do not presume any assumptions on the data, there is currently no approach that exhibits all these features.

This chapter presents a novel approach for the automatic detection of potentially erroneous `owl:sameAs` statements. The approach consists of applying an existing community detection algorithm to an RDF graph that contains solely `owl:sameAs` statements. Based on the communities that are detected, an error degree is calculated for each identity link in the graph. The error degree of an `owl:sameAs` link depends on the density of the community(ies) in which the two terms exist, and whether the identity link is symmetrical or not. It is subsequently used to rank identity links, allowing potentially erroneous links to be identified, and potentially true `owl:sameAs` to be validated. Since the here presented approach is specifically developed in order to be applied to real-world data, the experiment is run on the largest collection of identity links to date.

This chapter makes the following contributions:

1. It presents an approach that detects potential erroneous `owl:sameAs` links, and validates potential correct ones based on the topology of the identity network itself. Not requiring access to resource descriptions, property mappings, vocabulary alignments, or additional assumptions like the UNA, constitute the main strong points of this approach with comparison to the state of the art.

2. It calculates and publishes the error degree of over 558 million `owl:sameAs` statements in the LOD Cloud with a total runtime of 11 hours. Showing that an error degree of every identity link can be calculated in practice.

3. It reveals that the network structure of the `owl:sameAs` links, and eventually our proposed error degree, can indeed be used to distinguish between correct and incorrect `owl:sameAs` statements in many cases.

4. It presents an analysis on some of the incorrect identity links' sources and types, and the network effect that some of these links can cause.

The rest of this chapter is structured as follows. Section 4.1 introduces the notion of a community structure in a network and some of the community detection algorithms. Section 4.2 describes our approach for detecting erroneous identity links. Section 4.3 describes the experiments and the implementation. Section 4.4 gives an analysis and an evaluation of the efficiency of the presented approach. Section 4.5 concludes.

4.1 Community Structure

This chapter presents an approach for detecting erroneous identity links on the Web, by introducing a measure that is based on the community structure of the identity network. We believe community detection to be a particularly good fit for identity error detection, since it can be applied to the network structure of the `owl:sameAs` graph itself. In fact, we suppose that the quality of an identity link can be evaluated based on the density of the community(ies) in which this link belongs. Before presenting our approach, we introduce in this section what is a community structure in a network, and some of the most effective approaches in detecting such structure.

4.1.1 Overview

The modern science of networks has brought significant advances to our understanding of complex systems. One of the most relevant features of graphs representing real systems is community structure. Community detection is a form of data analysis that seeks to automatically determine the community structure of a complex network. Importantly, it only requires information that is already encoded in the network topology. Despite the absence of a universally agreed upon definition, communities are typically thought of as groups that have dense connections among their members, but sparse connections with the rest of the network. Figure 4.1 illustrates an example of a community structure, with three groups of nodes with dense internal connections and sparser connections between the groups. The three communities are non-overlapping, as there does not exist a node which belongs to multiple communities.

Detecting a network's community structure is of great importance in many concrete applications and disciplines such as computer science, biology, and sociology, disciplines where systems are often represented as graphs. This has led to the emergence of several community detection algorithms, mostly making use of techniques from physics (e.g. spin model, optimization, random walks), as well as making use of computer science concepts and methods (e.g. nonlinear dynamics, discrete mathematics) [Fortunato, 2010]. All such techniques aims at identifying group of nodes which are connected "more densely" to each other than to nodes in other groups. Hence, the differences between such methods ultimately come down to the precise definition of "more densely" and the algorithmic heuristic followed to identify such groups [Porter et al., 2009]. According to [Plantié and Crampes, 2013], community detection algorithms have three types of outputs:

Graph Partition. Most community detection algorithms returns a graph partition, where each node is associated with solely one group of nodes, without any overlap between these groups (e.g. Figure 4.1).

Hypergraph. The hypergraph model, where communities can overlap, is known to be specially relevant in social networks, where persons have connections to several social groups like family, friends, and colleagues. See [Xie et al., 2013] for a survey on such type of algorithms.

Concept graphs or Galois lattices. The first use of Galois lattices for representing network data is owed to [Freeman and White, 1993]. In Galois lattices, a community (called concept) is defined as individuals (called objects) who share a subset of properties. The result of a Galois lattice based algorithm is a unique and complete lattice of overlapping concepts (i.e. objects can appear in multiple, and even all, concepts).

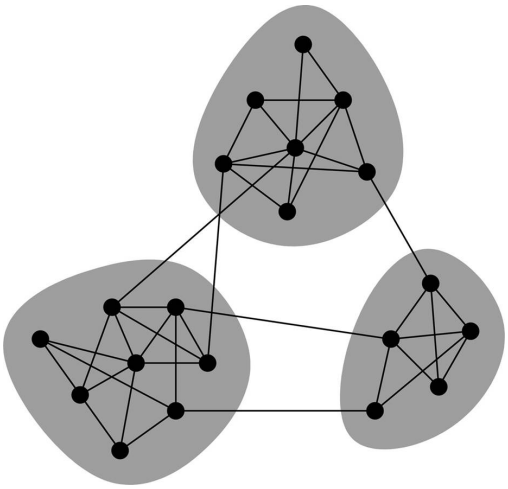


Figure 4.1: A simple graph with three non-overlapping communities.

Many methods have been proposed to extract non-overlapping communities. This availability of methods is certainly due to the ease of describing this type of problem and drawing a partition in comparison with hypergraphs or Galois lattices. Overlapping communities have gained popularity since [Palla et al., 2005]. However, it is still unclear how to characterize vertices who are shared by multiple communities, and particularly the shared vertices who lies in central positions of the communities (as opposed to expecting communities to share vertices lying at their borders). In addition, the membership of vertices in different communities enormously increases the number of possible covers with respect to standard partitions, resulting in much more computationally demanding algorithms [Fortunato, 2010]. Similarly to hypergraphs, computing a Galois hierarchy from a graph is much more computationally demanding with respect to standard partitions, and requires an input graph with a set of different properties for returning multiple lattice concepts.

As we aim to detect the community structure of the `owl:sameAs` network, we find that graph partitioning algorithms are more suitable in comparison with hypergraphs and Galois lattices for the following reasons:

Scalability constraint. As discussed in the previous chapter, the `owl:sameAs` network is a graph containing hundreds of millions of `owl:sameAs` statements. Hence a low computationally demanding algorithms for calculating its community structure is a necessary requirement.

Identity network properties. The `owl:sameAs` network is a graph uniquely composed of `owl:sameAs` links. Since the Galois hierarchy requires a number of different properties, it is not suitable for detecting the community structure in this case.

Identity constraint. Communities that can overlap are interesting in social net-

works, where the network properties have weak semantics (e.g. knows, related to, has friend). Since identity is binary and transitive, overlapping communities are more difficult to interpret.

In the next section, we present some of the graph partitioning algorithms.

4.1.2 Graph Partitioning Algorithms

In this section, we focus on graph partitioning algorithms that are more suitable in detecting the community structure of the `owl:sameAs` network. Even by restricting our choice to such type of algorithms, there still exist a great number of algorithms that partitions the graph into a set of densely related group of nodes. For instance, in one of the most exhaustive surveys with respect to the number of tackled methods, [Fortunato, 2010] classifies the graph partitioning algorithms into seven families:

Traditional methods representing the traditional clustering algorithms, such as the popular *k-means* algorithm [MacQueen et al., 1967] that partitions the graph into a k number of clusters given as input.

Divisive algorithms which rely on calculating the *betweenness centrality* of the graph vertices, such as [Newman and Girvan, 2004].

Spectral algorithms which rely on the use of *spectral properties* of graph matrices for finding partitions, such as [Donetti and Munoz, 2004].

Dynamic algorithms which consist of methods employing processes running on the graph, such as *spin-spin* interactions [Reichardt and Bornholdt, 2004], *random walks* [Zhou and Lipowsky, 2004], and *synchronization* [Boccaletti et al., 2007].

Statistical inference-based methods which aim at deducing properties of graphs starting from a set of observations and hypotheses on how vertices are connected to each other, such as methods adopting *Bayesian inference* [Newman and Leicht, 2007].

Multi-resolution and hierarchical methods which aim at detecting communities at different scales, resulting in more than one graph partition, such as [Arenas et al., 2008].

Modularity-based algorithms which aim on optimising the *modularity* quality function. Modularity is a measure firstly introduced by [Newman and Girvan, 2004] to measure the quality of community detection algorithms, and since then, it has rapidly become the most used and best known quality function [Fortunato, 2010].

Having a great number of clustering techniques, we have relied on existing surveys for choosing the best performing community detection algorithm for our task. In their 2009 survey, [Lancichinetti and Fortunato, 2009b] carried out a comparative analysis of the performances of 12 community detection algorithms[1], that exploit some of the most interesting ideas and techniques that have been developed over the last years. The tests were performed against a class of benchmark graphs, with heterogeneous distributions of degree and community size, including the GN benchmark [Girvan and Newman, 2002], the LFR benchmark [Lancichinetti et al., 2008, Lancichinetti and Fortunato, 2009a], and some random graphs. This study concludes that the modularity-based method by [Blondel et al., 2008], the statistical inference-based method by [Rosvall and Bergstrom, 2008], and the multiresolution method by [Ronhovde and Nussinov, 2009] all have an excellent performance, with the additional advantage of low computational complexity.

In a more recent study, [Yang et al., 2016] compare the results of 8 state-of-the-art community detection algorithms in terms of accuracy and computing time. Interestingly, only half of these algorithms were considered in the previous survey, with the tests also being conducted on the LFR benchmark. This study concludes that by taking both accuracy and computing time into account, the modularity-based method by [Blondel et al., 2008] outperforms all the other algorithms.

Given that the method proposed by [Blondel et al., 2008] outperforms the other 15 algorithms in two different studies, with an additional advantage of low computational complexity, we will deploy this algorithm for detecting the community structure in the `owl:sameAs` network. Next section presents an overview of this algorithm.

4.1.3 Louvain Algorithm

The Louvain algorithm is a method for detecting communities in large networks, created by [Blondel et al., 2008] from the University Catholique de Louvain (the affiliation of authors has given the method its name). It is a greedy non-deterministic method, introduced for the general case of weighted graphs, for the purpose of optimising the modularity of the partitions. The modularity of a partition is a scalar value between -1 and 1 that measures the density of links inside communities as compared to links between communities. In the case of weighted networks, modularity is defined as follows:

[1] Admitting that it is impossible to consider all existing algorithms, as their number is huge.

$$Q = \frac{1}{2m} \sum_{i,j} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta(c_i, c_j) \tag{4.1}$$

where:

$\boldsymbol{A_{ij}}$ represents the weight of the edge between the nodes i and j

$\boldsymbol{k_i}$ **and** $\boldsymbol{k_j}$ represent the sum of the weights of the edges attached to the nodes i and j, respectively

$\boldsymbol{c_i}$ **and** $\boldsymbol{c_j}$ represent the community to which to the nodes i and j are assigned, respectively

$\boldsymbol{2m}$ $= \frac{1}{2} \sum_{i,j} A_{ij}$ and representing the sum of all of the edge weights in the graph

$\boldsymbol{\delta(u, v)}$ is 1 if $u = v$ and 0 otherwise

Modularity has been used to compare the quality of the partitions obtained by different methods, but also as an objective function to optimize [Newman and Girvan, 2004]. Networks with high modularity have dense connections between the nodes within communities but sparse connections between nodes in different communities. This is the intuition of the Louvain algorithm, which is divided in two phases:

Firstly, it starts out by assigning a different community to each node of a given network. Hence, in this initial partition, there as many communities as there are nodes. Then, given a node u, the algorithm computes the gain in weighted modularity resulting from putting u in the community of its neighbour v. The node u is then placed in the community of the neighbour that yields the highest gain to the modularity score, but only if this gain is positive. If no positive gain is possible, u stays in its original community. This process is applied repeatedly and sequentially for all nodes until no further improvement can be achieved (i.e. when modularity cannot be improved by any node move). At the end of the first phase, one obtains the first level partition.

In the second phase, each community from the previous phase is regarded as a single node. To do so, the weights of the links between the new nodes are given by the sum of the weight of the links between nodes in the corresponding two communities. Links between nodes of the same community lead to self-loops for this community in the new network. Once this second phase is completed, the same procedure is repeated until the modularity (which is always computed with respect to the original graph) no longer increases.

Although the exact computational complexity of the Louvain algorithm is not known, the method seems to run in time $O(N \log N)$, with N representing the number of nodes in the graph [Blondel et al., 2008]. The exact modularity optimization is known to be NP-hard (non-deterministic polynomial-time hard), with most of the computational effort spent on the optimization at the first level.

4.2 Approach

We believe community detection to be a particularly good fit for identity error detection, since it can be applied to the network structure of the `owl:sameAs` graph itself. In fact, we suppose that the quality of an identity link can be evaluated based on the density of the community(ies) in which this link belongs. Since the Louvain algorithm has already been successfully used in other domains, we believe that it can also perform well on the task of detecting `owl:sameAs`-based communities. The approach that we suggest does not require access to resource descriptions, property mappings, or vocabulary alignments. Also, it does not rely on additional assumptions like the UNA that could be false for some dataset (e.g., datasets that are constructed over a longer period of time and/or by a large group of contributors). Finally, current approaches for identity error detection have not always been applied to real-world `owl:sameAs` links, and no current approach has been evaluated at Web scale (i.e. applied to hundreds of millions of links) due to multiple dimensions of complexity. In the following, we identify the desired requirements for our algorithm:

Low memory footprint. The calculation of erroneous identity links must not have a large memory footprint, since it must be able to scale to very large identity networks, and preferably to all identity statements that appear in the LOD Cloud.

Parallel computing. It must be possible to perform computation in parallel, to allow errors to be detected relatively quickly, preferably directly after the publication of the potential error into the LOD Cloud.

Dynamic. Calculation must be resilient against incremental updates. Since triples are added to and removed from the LOD Cloud constantly, adding or removing an `owl:sameAs` link must only require a re-ranking of the concerned links.

This section presents our approach for detecting erroneous identity links by exploiting the community structure of the identity network itself. This section describes the two main steps that our approach is composed of: the construction of the identity network (section 4.2.1), and the ranking of each identity link

based on the community structure (section 4.2.2). This chapter uses the terminology and symbolism introduced in the previous chapter.

4.2.1 Identity Network Construction

Constructing the identity network consists of two phases: extracting the explicit identity network (definition 2), and transforming it into a more compacted and weighted identity network.

Explicit Identity Network Extraction

The first phase consists of extracting the explicit identity network G_{ex} from a data graph G (definition 1). This phase is described in the previous chapter (section 3.1.1).

Identity Network Construction

In this second phase, we can reduce the size of the explicit identity network G_{ex} into a more concisely represented undirected and weighted identity network I (definition 4), without losing any significant information. Since reflexive `owl:sameAs` statements are implied by the semantics of identity, there is no need to represent them explicitly. In addition, since the symmetric statements e_{ij} and e_{ji} make the same assertion: that v_i and v_j refer to the same thing, we can represent this more efficiently, by including only one undirected edge with a weight of 2. A weight of 1 is assigned for edges which either e_{ij} or e_{ji}, but not both, are present in N.

Definition 4 (Identity Network) Given an explicit identity network $G_{ex} = (N, E_{ex})$, the identity network is an undirected labeled graph $I = (V_I, E_I, \{1, 2\}, w)$, where V_I is the set of nodes ($V_I \in N$), and E_I is the set of edges. $\{1, 2\}$ are the edges labels, and $w : E_I \rightarrow \{1, 2\}$ is the labeling function that assigns a weight w_{ij} to each edge e_{ij}. For an explicit identity network $G_{ex} = (N, E_{ex})$, the corresponding identity network I is derived as follows:

- $E_I := \{e_{ij} \in E_{ex} \mid i < j\}$
- $V_I := N[E_I]$, i.e., the vertex-induced subgraph
- $w(e_{ij}) := \begin{cases} 2, & \text{if } e_{ij} \in E_{ex} \text{ and } e_{ji} \in E_{ex} \\ 1, & \text{if not} \end{cases}$

4.2.2 Links Ranking

Our approach of detecting erroneous identity links consists of ranking each `owl:sameAs` link in the data graph. For ranking the identity links, we partition the identity network into several connected components. After partitioning, we aim to detect, in a separate manner in each of these networks, the `owl:sameAs` links that are incorrect by assigning an error degree for each link. Partitioning the graph is more logically sound, since there is no identity links between two connected components. In addition, partitioning the graph is beneficial for implementing an algorithm that achieves the requirements cited in the beginning of the chapter (low memory footprint, parallel computing, and dynamic).

Graph Partitioning

Given $I = (V_I, E_I, \Sigma_{E_I}, w)$, a partitioning of V_I is a collection of non-empty and mutually disjoint subsets $V_k \subseteq V_I$ that together cover V_I. Since the closure of E_I forms an equivalence set (the semantics of the `owl:sameAs` property states that it is reflexive, symmetric, and transitive), it also induces a unique partitioning. We call members of this partition *identity sets*. These partition members correspond to the connected components of I that we call *equality sets* (definition 5). For partitioning the graph, we apply the technique that we used for the identity links closure in the previous chapter (section 3.2.4).

Definition 5 (Equality Set) Given an identity network $I = (V_I, E_I, \{1, 2\}, w)$, an equality set Q_k is a connected component of I. The identity set V_k represents the set of members of this equality set.

Links Ranking

After partitioning the identity network into several equality sets, we detect a set of non-overlapping communities by applying the *Louvain* algorithm (section 4.1.3) for each equality set. Given an equality set Q_k, the *Louvain* algorithm returns a set of non-overlapping communities $C(Q_k) = \{C_1, C_2, \ldots, C_n\}$ where:

- a community C of size $|C|$ (i.e. the number of nodes) is a subgraph of Q_k such that the nodes of C are densely connected (i.e. the modularity of the Q_k is maximized).
- $\bigcup_{1 \leq i \leq n} C_i = Q_k$ and $\forall C_i, C_j \in C(Q_k)\ s.t.\ i \neq j, C_i \cap C_j = \emptyset$.

We then evaluate each identity link by relying on its weight and the structure of the community(ies) it occurs in. We hypothesise that an identity link which is reciprocally asserted has higher chances of correctness that a non-symmetrically

asserted identity link. In addition, we hypothesise that not all detected communities have similar qualities. For this, and by relying on the community's density, we assign higher chances of correctness for `owl:sameAs` links connecting two IRIs in a densely connected community, or connecting two IRIs in two heavily interlinked communities. More precisely, to compute an erroneous degree of each `owl:sameAs`, we distinguish between two types of possible links: the *intra-community links* and the *inter-community links*.

Definition 6 (Intra-Community Link) Given a community C, an intra-community link in C noted by e_C is a weighted edge e_{ij} where v_i and $v_j \in C$. We denote by E_C the set of intra-community links in C.

Definition 7 (Inter-Community Link) Given two non overlapping communities C_i and C_j, an inter-community link between C_i and C_j noted by $e_{C_{ij}}$ is an edge e_{ij} where $v_i \in C_i$ and $v_j \in C_j$. We denote by $E_{C_{ij}}$ the set of inter-community links between C_i and C_j.

For evaluating an *intra-community link*, we rely both on the density of the community containing the edge, and the weight of this edge. The lower the density of this community and the weight of an edge are, the higher the *error degree* will be.

Definition 8 (Intra-Community Link Error Degree) . Let e_C be an intra-community link of the community C, the intra-community *error degree* of e_c denoted by $err(e_C)$ is defined as follows:

$$err(e_C) = \frac{1}{w(e_C)} \times (1 - \frac{W_C}{|C| \times (|C| - 1)}) \tag{4.2}$$

where $W_C = \sum_{e_C \in E_C} w(e)$

For evaluating an *inter-community link*, we rely both on the density of the inter-community connections, and the weight of this edge. The less the two communities are connected to each other and the lower the weight of an edge is, the higher the *error degree* will be.

Definition 9 (Inter-Community Link Error Degree) . Let $e_{C_{ij}}$ be an inter-community link of the communities C_i and C_j, the inter-community *error degree* of $e_{C_{ij}}$ denoted by $err(e_{C_{ij}})$ is defined as follows:

$$err(e_{C_{ij}}) = \frac{1}{w(e_{C_{ij}})} \times (1 - \frac{W_{C_{ij}}}{2 \times |C_i| \times |C_j|}) \tag{4.3}$$

where $W_{C_{ij}} = \sum_{e_{C_{ij}} \in E_{C_{ij}}} w(e)$

Algorithm 1 provides a summary of the necessary steps for ranking identity links, taking a data graph as input, and returning an error degree for each `owl:sameAs` link in the identity network.

Algorithm 1: Identity Links Ranking

Input: G: a Data graph
Output: E^{err}: a set of pairs in the from $\{(e_1, err(e_1)), \ldots, (e_m, err(e_m))\}$ with m is the number of edges in the identity network extracted from G

```
1  I_ex ← ExtractSameAsEdges(G); // the explicit identity network
2  I ← empty_graph; // the identity network
3  foreach (e(v1, v2) ∈ I_ex and v1 ≠ v2) do
4      if (I.containsEdge(e(v2, v1, 1))) then
5          I.updateWeight(e(v2, v1, 2); // set the weight of this edge to 2
6      else
7          I.addEdge(e(v1, v2, 1)); // add this edge to I with a weight = 1
8  P ← I.partition(); // partitioning the graph into equality sets
9  foreach (Q ∈ P) do
10     C_set ← LouvainCommunityDetectionAlgorithm(Q);
11     foreach (e ∈ C_set) do
12         if (e is intra-community-edge(c_i)) then
13             err(e) ← intraCommunityErroneousness(c_i);
14         else
15             // e is an inter-community edge, c_j is the other community to
                   which e is belonging to;
16             err(e) ← interCommunityErroneousness(c_i, c_j);
17         E^err.add(e, err(e));
18 return E^err;
```

4.3 Implementation & Experiments

In this section we describe our implementation and experiments of the previously presented approach on a large copy of the LOD Cloud.

4.3.1 Data Graph

We use the same data graph described in the previous chapter (section 3.2.1). The *LOD-a-lot* dataset [Fernández et al., 2017], which represents our data graph (definition 1), contains more than 28.3 billion unique triples that represent a large copy of the LOD Cloud. This dataset contains more than 5 billion unique

terms, related by more than 1.1 billion predicates. This data is exposed in a single HDT file that is 524 GB in size, and publicly accessible (via an LDF interface) and downloadable (as HDT Dump).

4.3.2 Explicit Identity Network Extraction

In order to extract the explicit identity network we use the method described in the previous chapter (section 3.2.2). It consists in performing a Triple Pattern query of the form $\langle ?, \texttt{owl:sameAs}, ? \rangle$ with the HDT C++ library[2]. This extraction process takes around four hours using 1 CPU core, resulting in an explicit identity network of 558.9M edges and 179.73M nodes. The explicit identity network is publicly available at `https://sameas.cc/triple`.

4.3.3 Identity Network Construction

From the explicit identity network described above, we build the identity network (definition 4) containing ~331M weighted edges and 179.67M terms. We leave out ~2.8M reflexive edges and ~225M *duplicate* symmetric edges. As a result, we also leave out 67,261 nodes that only appear in such removed edges. This indicates that 68% of the identity network edges are redundantly asserted, with a weight = 2.

4.3.4 Graph Partitioning

The next step consists of partitioning the identity network into several equality sets (definition 5). We have deployed the algorithm described in the previous chapter (section 3.2.4) that partitions the identity network into ~49M identity sets, in just under 5 hours using 2 CPU cores. The equality sets were easily constructed using the explicit identity network and the resulted identity sets which are publicly available at `http://sameas.cc/id`.

4.3.5 Links Ranking

Once the identity network has been partitioned, we apply the *Louvain* algorithm to detect communities in each equality set. As discussed in section

[2] `https://github.com/rdfhdt/hdt-cpp`

4.1.3, the *Louvain* method is a greedy and non-deterministic algorithm. Meaning that in different runs, the algorithm might produce different communities, with no insurance that the global maximum of modularity will be attained. For this, we have run *Louvain* 10 times on each equality set, and finally considered the community structure with the highest modularity. After detecting the communities, we assign an error degree to all edges of each equality set. This process takes 80 minutes[3], resulting an error degree to each irreflexive[4] `owl:sameAs` statement (~556M statements) in the explicit identity network. The error degree distribution of these statements is presented in Figure 4.2. This figure shows that around 73% of the statements have an error degree below 0.4, whilst around 5% of the `owl:sameAs` statements have an error degree higher than 0.8. Whilst this distribution is mainly caused by the high number of symmetrical identity statements in the LOD, it also indicates that most equality sets have a rather dense structure. The 179.67M terms of the identity network were assigned into a total of 55.6M communities, with the communities size varying between 2 and 4,934 terms (averaging ~3 terms per community). The Java implementation of the link ranking process is available at `http://github.com/raadjoe/LOD-Community-Detection`. The erroneous degree of all the `owl:sameAs` statements are available in our identity Web service (`https://sameAs.cc`).

4.4 Analysis & Evaluation

4.4.1 Community Structure Analysis

In this section we provide a first analysis of the community structure obtained from two equality sets (the largest equality set and the one about Barack Obama) based on the IRIs contained in the communities. In a 2016 study conducted on the same data collection, [de Rooij et al., 2016] have shown that the social meaning encoded in IRI names significantly coincides with the formal meaning of IRI-denoted resources. Hence, indicating that IRIs can give an idea on the quality of the detected communities.

Community Structure in the Largest Equality Set

The largest equality set Q_{max} contains 177,794 terms connected by 2,849,650 undirected and weighted edges. This equality set is the result of the compaction

[3] on an 8GB RAM Windows 10 machine, using 2 CPU cores

[4] reflexive statements were discarded in I, and symmetrical ones have the same error degree

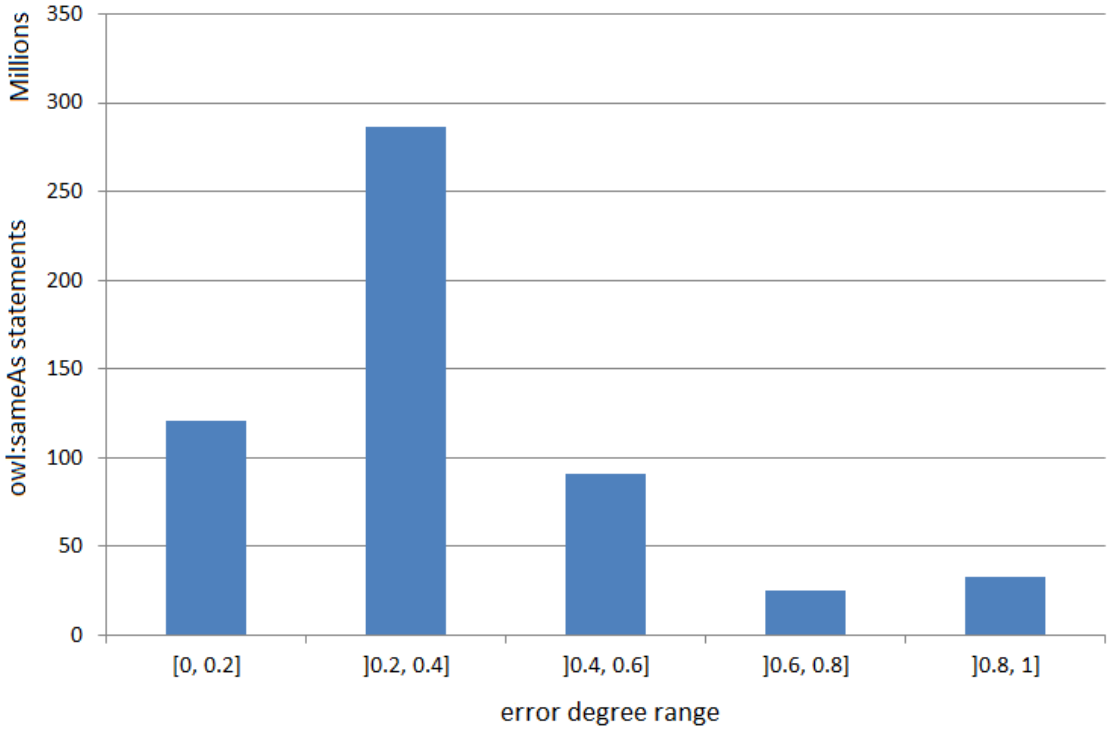


Figure 4.2: Error degree distribution of 556M `owl:sameAs` statements

.

of 5,547,463 distinct `owl:sameAs` statements (~ 1% of the `owl:sameAs` in the *LOD-a-lot* dataset). This identity set is available at `https://sameas.cc/term?id=4073`. By looking at the IRIs of this equality set, we can observe that it contains a large number of terms denoting different countries, cities, things and persons (e.g. Bolivia, Dublin, Coca-Cola, Albert Einstein, an *empty string*, and so on). This observation clearly shows that this equality set contains a large number of erroneous `owl:sameAs` statements .

Applying the *Louvain* algorithm on Q_{max} resulted in 930 non-overlapping communities, with a size varying from 32 to 2,320 terms per community. As a first interpretation on the community structure, we have solely looked at the IRIs. Despite a few exceptions, we can see that this algorithm is able to group related (and possibly identical) terms in the same community, while keeping out unrelated terms in other communities. For instance, the community C_{258}, illustrated in Figure 4.3 contains 242 terms. We can see from this excerpt that most of these terms come from the DBpedia dataset and refer to descriptions of Dublin expressed in different languages and ways: `City of Dublin`, `Capital of Ireland`, `Baile Atha Cliath` (Dublin in Irish), `Dyflin` (the old Norse name for The Kingdom of Dublin), etc. However, we can also see that this community contains terms that do not refer to the city of Dublin, but actually refer to the weather in Dublin or visitor information for Dublin.

```
-- Community 258 --  (size = 242)
<http://af.dbpedia.org/resource/Dublin>
<http://am.dbpedia.org/resource/ደብሊን>
<http://an.dbpedia.org/resource/Dublín>
<http://ar.dbpedia.org/resource/دبلن>
<http://ast.dbpedia.org/resource/Ciudá_de_Dublín>
<http://bat-smg.dbpedia.org/resource/Doblėns>
<http://be-x-old.dbpedia.org/resource/Дублін>
<http://br.dbpedia.org/resource/Dulenn>
<http://ca.dbpedia.org/resource/Dublín>
<http://ce.dbpedia.org/resource/Дублин>
<http://commons.dbpedia.org/resource/Dublin_-_Baile_Átha_Cliath>
<http://cs.dbpedia.org/resource/Dublin>
<http://dbpedia.org/resource/Baile_Atha_Cliath>
<http://dbpedia.org/resource/BÁC>
<http://dbpedia.org/resource/Capital_of_Ireland>
<http://dbpedia.org/resource/Capital_of_Republic_of_Ireland>
<http://dbpedia.org/resource/Central_Dublin>
<http://dbpedia.org/resource/City_Center,_Dublin>
<http://dbpedia.org/resource/City_of_Dublin>
<http://dbpedia.org/resource/Dyflin>
<http://dbpedia.org/resource/Europe/Dublin>
<http://dbpedia.org/resource/The_weather_in_Dublin>
<http://dbpedia.org/resource/UN/LOCODE:IEDUB>
<http://dbpedia.org/resource/Visitor_Information_for_Dublin,_Ireland>
<http://dbpedia.org/resource/West_Dublin>
<http://de.dbpedia.org/resource/Dublin>
<http://demo.openlinksw.com/Northwind/Province/ei/Dublin#this>
<http://sws.geonames.org/2964574/>
<http://wordnet.rkbexplorer.com/id/synset-Dublin-noun-1>
<http://www4.wiwiss.fu-berlin.de/flickrwrappr/photos/Dublin>
```

Figure 4.3: Excerpt of the 242 terms included in the community containing the IRI `http://dbpedia.org/resource/dublin`

.

With this excerpt of the Dublin community, we can see that an `owl:sameAs` statement between two terms in the same community is not necessarily correct, and requires evaluation as well.

Community Structure in the 'Barack Obama' Equality Set

We present here an analysis of the community structure detected on the equality set Q_{obama} which has a reasonable size and thus easier to analyse. The equality set containing the term `http://dbpedia.org/resource/Barack_Obama` is composed of 440 terms connected by 7,615 undirected and weighted edges. This equality set, illustrated in Figure 4.4, is based on 14,917 explicit `owl:sameAs` statements, and its identity set is available at (`https://sameas.cc/term?id=5723`).

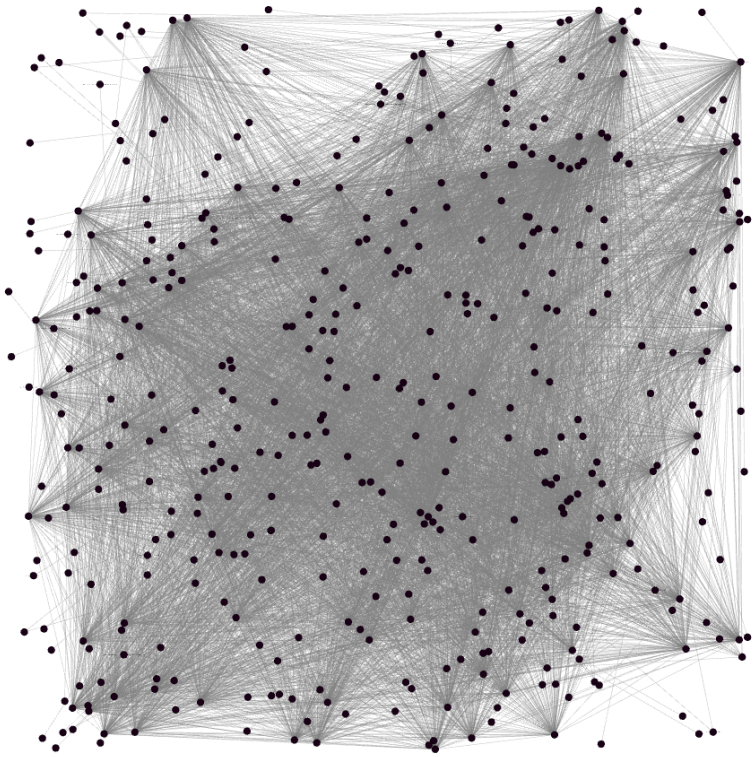


Figure 4.4: The equality set containing the term `http://dbpedia.org/resource/Barack_Obama`. It is composed of 440 terms and 7,615 undirected weighted edges, compacted from 14,917 `owl:sameAs` statements

Applying the *Louvain* algorithm on Q_{obama} resulted in 4 non-overlapping communities, with a size varying from 34 to 166 terms per community. The resulting community structure of Q_{obama} is presented in Figure 4.5, and can be interpreted as follows:

C_0 (purple) includes 166 terms, with 98% of the links of this community representing cross-language symmetrical links between DBpedia IRIs (e.g. `http://fr.dbpedia.org/resource/Barack_Obama`) referring to the person Barack Obama.

C_1 (green) includes 162 terms, mostly DBpedia IRIs of the person Obama in his different roles and political functions (e.g. `http://dbpedia.org/resource/President_barack_obama`, `http://dbpedia.org/resource/senator_obama`).

C_2 (orange) includes 78 terms, mostly referring to the presidency and administration of Barack Obama (e.g. `http://dbpedia.org/resource/Obama_cabinet`, `http://dbpedia.org/resource/Barack_Hussein_Obama_administration`)

C_3 (blue) includes 34 terms from different datasets denoting various entities such as: Barack Obama the person, his senate career,

and a misused literal (|"http://dbpedia.org/resource/United_States_Senate_career_of_Barack_Obama, "http://dbpedia.org/resource/Barack_Obama"^xsd:string).

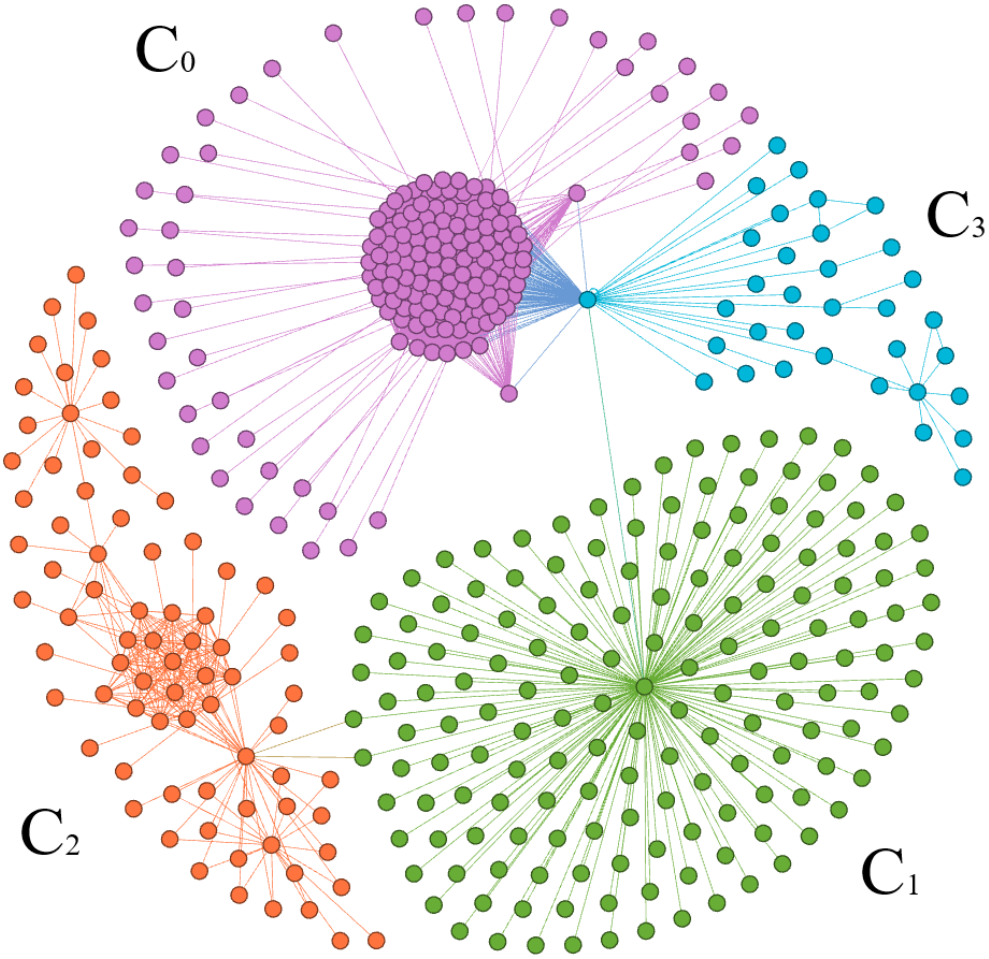


Figure 4.5: The communities detected from the equality set containing the term http://dbpedia.org/resource/Barack_Obama using the *Louvain* algorithm. The 4 detected communities are distinguished by their nodes' color. The full figure is available at https://sameas.cc/img/obama-large.svg.

4.4.2 Links Ranking Evaluation

In order to evaluate the accuracy of our ranking approach, we have conducted several manual evaluations. The judges relied on the descriptions[5] associated to the terms in the *LOD-a-lot* dataset [Fernández et al., 2017], and did not have

[5]judges were asked to not consider the owl:sameAs assertions associated to a term

any prior knowledge about each link's error degree (i.e. whether they are evaluating a well-ranked link or not). In order to avoid any incoherence between the evaluations, the judges were asked to justify all their evaluations, and were given the following instructions: **(a) the same:** if two terms denote the same entity (e.g. Obama and the First Black US President), **(b) related:** not intended to refer to the same entity but closely related (e.g. Obama and the Obama Administration, or Obama and the Wikipedia article of Obama), **(c) unrelated:** not the same nor closely related (e.g. Obama and the Indian Ocean), **(d) can't tell:** in case there are no sufficient descriptions available for determining the meaning of both terms (i.e. non-dereferenced IRIs and IRIs appearing solely as subjects or objects of `owl:sameAs` statements in the LOD).

A. Error degree interpretation in the 'Barack Obama' Equality Set

Firstly, we have relied on the previous observations, made on the community structure presented in Figure 4.5, to interpret the error degree distribution:

- an `owl:sameAs` statement in C_0 has an average error rate of 0.24. A manual evaluation of 30 random `owl:sameAs` statements in this community shows that they are all true identity links.

- the low density of C_1 has led to several correct `owl:sameAs` statements to have a high error degree (0.9). This is due to the fact that there is only one term linking to all the 161 other terms in this community, with most of these edges being non-symmetrical links.

- the only two `owl:sameAs` statements in this equality set with an error value $\simeq 1$ (0.999) are the edges in the graph connecting the IRI `http://rdf.freebase.com/ns/m.05b6w1g` from C_2 to both IRIs `http://dbpedia.org/resource/President_Barack_Obama` and `http://dbpedia.org/resource/President_Obama` from C_1. Relying on their descriptions in the *LOD-a-lot* dataset, we can see that the Freebase IRI refers to the presidency of Obama, while the two other IRIs refer to the person Obama, indicating that both statements are incorrect. These two detected incorrect identity statements have led to the false equivalence of the 78 terms of C_2 with the rest of the network's terms.

B. Accuracy Evaluation on a Subset of the Identity Network

In this evaluation, we aim at defining a threshold x of the error degree, in which `owl:sameAs` links that have an error degree $\leq x$ will have high probability of

correctness, and links which have an error degree $> x$ have high probability of being erroneous. In order to determine this threshold, four semantic web experts were asked to evaluate a subset of the identity network. Based on the judges' evaluations we can deploy the following terms:

True Positives (TP) referring to `owl:sameAs` links which have an error degree $> x$ and were evaluated by the judges as incorrect (related or unrelated) identity links.

False Positives (FP) referring to `owl:sameAs` links which have an error degree $> x$ and were evaluated by the judges as true identity links.

True Negatives (TN) referring to `owl:sameAs` links which have an error degree $\leq x$ and were evaluated by the judges as true identity links.

False Negatives (FN) referring to `owl:sameAs` links which have an error degree $\leq x$ and were evaluated by the judges as incorrect (related or unrelated) identity links.

By definition, accuracy indicates the percentage of links correctly evaluated by our approach:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.4}$$

The judges were asked to evaluate 200 `owl:sameAs` links (50 links each), representing a sample of each bin of the error degree distribution presented in Figure 4.2. We consider that when a human expert is not able to confirm the correctness of a certain identity link due to the absence of necessary descriptions for one of the two involved IRIs, no automated approach can. With this assumption, we will not consider links judged by the experts as "can't tell" in the accuracy evaluation.

From the results presented in Table 4.1, we can observe that:

- the higher an error degree is, the more likely that the link is erroneous.
- 100% of the evaluated links with an error degree ≤ 0.4. are correct.
- when the error degree is between 0.4 and 0.8, 83.3% of the `owl:sameAs` links are correct. However, in 13.3% of the cases, such links might have been used to refer to two different, but related terms.
- an `owl:sameAs` with an error degree > 0.8 is a less reliable identity statement, referring in 31.8% of the cases to two different, and most of times unrelated terms.

Table 4.1: Evaluation of 200 `owl:sameAs` links, with each 40 links randomly chosen from a certain range of error degree. The percentages between parentheses are calculated without considering the links evaluated as "can't tell".

error degree range	0-0.2	0.2-0.4	0.4-0.6	0.6-0.8	0.8-1	**total**
same	35 (100%)	22 (100%)	18 (85.7%)	7 (77.8%)	15 (68.2%)	**97 (89%)**
related	0	0	2	2	2	**6**
unrelated	0	0	1	0	5	**6**
related + unrelated	0 (0%)	0 (0%)	3 (14.3%)	2 (22.2%)	7 (31.8%)	**12 (11%)**
can't tell	5	18	19	31	18	**91**
total	**40**	**40**	**40**	**40**	**40**	**200**

We have further investigated the 22 evaluated identity links with an error degree over 0.8. Two features were observed from the 7 incorrect identity statements: (i) their error degree is most of the times higher than the true `owl:sameAs` links, and (ii) they all belong to equality sets with a higher number of terms than the true ones. To further investigate these observations, we have evaluated 60 additional links with an error degree > 0.9. The first set of links (S1) represents 20 random identity links from the largest equality set. The second set of links (S2) represents 20 random identity links with an error degree ≃1 (> 0.99). The third set of links (S3) represents 20 random links from the largest equality set with an error degree ≃1.

The results presented in Table 4.2, suggest that our approach is accurate in detecting erroneous identity links when the threshold is fixed at 0.99, and when only equality sets with a high number of terms are considered. However, since it is difficult to determine the equality sets' size range in which our approach would maintain such high accuracy, we fix the threshold at 0.99 without considering the equality set size.

In order to calculate an approximative accuracy of our approach based on this threshold, we rely on the set of links evaluated during these experiments. More specifically we consider the links evaluated in Table 4.1, Table 4.2, and links previously evaluated in the 'Obama' Equality Set. Table 4.3 presents the True Negatives (TN) which are the correct `owl:sameAs` with an error degree ≤ 0.99, the True Positives (TP) which are the erroneous ones with an error degree > 0.99, the False Positives (FP), and the False Negatives (FN).

Links with err ≤ 0.99: Table 4.1 includes 109 `owl:sameAs` links with an error

Table 4.2: Evaluation of 60 owl:sameAs links with an error degree > 0.9, with the first set of 20 owl:sameAs links (S1) randomly chosen from the largest equality set, (S2) randomly chosen from all links with an error degree ≃1, (S3) randomly chosen from the largest equality set with an error degree ≃1

	Largest equality set(S1)	*err* ≃1 (S2)	Largest & *err* ≃1 (S3)
same	6 (50%)	6 (60%)	2 (11.7%)
related	1	1	2
unrelated	5	3	13
related+unrelated	6 (50%)	4 (40%)	15 (88.2%)
can't tell	8	10	3
Total	**20**	**20**	**20**

degree ≤ 0.99 (i.e. no evaluated link from the [0.8-1] bin have an error degree > 0.99), with 97 out of these 109 judged as correct identity links. Table 4.2 includes 11 owl:sameAs with an error degree ≤ 0.99 (i.e. 1 out of the 12 links in the (S1) set has an error degree > 0.99), with 6 out of these 11 links evaluated as correct links. We have manually evaluated 30 owl:sameAs from the C_0 in the 'Obama' Equality Set with an error degree ≤ 0.99, with all of these links being judged as true owl:sameAs, representing cross-language identity links. Hence, 133 links (TN) out of the 150 links (TN+FN) with an error degree ≤ 0.99 are correct identity links, suggesting a precision of 88.6% in validating owl:sameAs links.

Links with err > 0.99: Table 4.2 includes 28 owl:sameAs with an error degree > 0.99, with 20 out of these 28 links evaluated as erroneous links. We have also manually evaluated the only 2 owl:sameAs links in the 'Obama' equality set with an error degree > 0.99, connecting the Freebase resource from C_2 to DBpedia resources in C_1, with both of these links judged as erroneous. Hence, 22 (TP) out of the 30 evaluated links with an error degree > 0.99 (TP+FP) are erroneous, suggesting a precision of 73.3% in detecting erroneous identity links. However, we admit that in practice, in random equality sets, the precision might be closer to 40% as the (S2) evaluation suggests.

Our manual evaluation of 180 owl:sameAs statements[6] suggests that our approach is able to correctly classify an owl:sameAs link (as correct or erro-

[6]Discarding the statements judged by the experts as "can't tell"

Table 4.3: Correctness of the manually evaluated links, based on a threshold of 0.99. Specifically it presents the True Negatives (TN) which are the correct `owl:sameAs` with an error degree ≤ 0.99, the True Positives (TP) which are the erroneous ones with an error degree > 0.99, the False Negatives (FN), and the False Positives (FP) from the links evaluated in Table 4.1, Table 4.2, and the 'Obama' Equality Set.

	TN	**TP**	**FN**	**FP**	**Total**
Table 4.1	97	0	12	0	**109**
Table 4.2	6	20	5	8	**39**
'Obama' EqSet	30	2	0	0	**32**
Total	**133**	**22**	**17**	**8**	**180**

neous) with an 86% accuracy. If we discard the non-randomly chosen links (i.e. discard the links manually evaluated from the 'Obama' equality set and the largest equality set), the accuracy of our approach would almost remain the same (85%), due to the high number of true negatives.

C. Accuracy Evaluation according to a State-of-the-Art Gold Standard

We have tested the accuracy of our approach on the only state-of-the-art approach [Acosta et al., 2013] that publishes[7] its manually evaluated links. This content-based approach, presented in Section 2.3, uses crowdsourcing for evaluating the quality of the links in the LOD. During their evaluation, the authors have manually evaluated 95 `owl:sameAs` links, corresponding all to correct DBpedia-Freebase identity interlinks. Out of these 95 gold standard links, we found 78 in our dataset (82%). Verifying their error degrees[8], we found that only 1 out of these 78 links was assigned an error degree higher than 0.99 (FP), with the rest having an error degree between 0.52 and 0.94 (TN), suggesting an accuracy of 98.7% according to this gold standard.

D. Recall Evaluation

In order to evaluate the recall of our approach, we have verified how our approach can rank newly introduced erroneous `owl:sameAs` statements. Firstly, we have chosen 40 random terms[9] in the explicit identity network, making

[7] `http://people.aifb.kit.edu/mac/DBpediaQualityAssessment/experiments.html`

[8] `https://github.com/raadjoe/LOD-Community-Detection/blob/master/resources/interlinking_GS_err.csv`

[9] we also made sure to include 5 terms that belong to the same equality set

sure that all these terms are different and not explicitly owl:sameAs (e.g. dbr:Paris, dbr:Strawberry, dbr:Facebook). From the 40 selected terms, we have generated all the possible 780 undirected edges between them. We added separately, each edge e_{ij} to the identity network with $w(e_{ij})$=1, calculated its error degree, and removed it from the identity network before adding the next one. The resulted error degrees of the newly introduced erroneous identity links range from 0.87 to 0.9999. When the threshold is fixed at 0.99, the recall of detecting erroneous identity links is 93%, with 725 (TP) out of the 780 added links (TP+FN) having an error degree > 0.99.

E. Evaluation of the Symmetry Impact in the Erroneous Degree

In this final evaluation, we want to verify the hypothesis we consider in our error degree measure, that a symmetrical identity link have a higher chance of correctness than a non-symmetrical one. We have evaluated in these experiments, including the 78 gold standard links, a total of 370 owl:sameAs links. The judges were able to classify 258 of these links, in which 39 were judged as erroneous identity statements: 12 links in Table 4.1, 25 links in Table 4.2, and 2 links from the Barack Obama equality set that connect the communities $C1$ and $C2$. As Table 4.4 shows, from the 258 evaluated owl:sameAs links, 94 correspond to symmetrically duplicate links (i.e. they have a weight of 2 in the identity network). Only 2 out of these 94 symmetrical links were judged as *related* by the judges, with the rest being judged as correct identity links (98% chance of correctness). On the other hand, 37 out of the 164 non-symmetrical owl:sameAs links were judged as erroneous (10 related and 27 unrelated), indicating a 77% chance of correctness. These number suggests that a symmetrical identity link has more chances of correctness than a non-symmetrical one.

For further investigation, we have discarded the weight from the error degree measure (i.e. the error degree is now solely dependent on the density of the communities), and ranked all the owl:sameAs links all over again. To evaluate the impact of the weight on the accuracy of detecting erroneous links, we have randomly evaluated 30 owl:sameAs links that have the same characteristics as the links from the (S3) set (i.e. error degree > 0.99 and belong to the largest equality set). Out of the 30 links, the judges have evaluated that 17 owl:sameAs relate two resources referring to the same real world entity, 2 owl:sameAs relates two unrelated resources, and were not able to judge the remaining 11 links due to insufficient descriptions. This evaluation shows that when discarding the weight from the error degree, the precision of the approach in detecting erroneous owl:sameAs links drops from 88% to 11% (in the largest equality set and when the threshold is fixed at 0.99). This is due to the addition of $\sim 20K$ duplicate symmetrical links, with a value > 0.99, in the largest equality set.

Table 4.4: Analysis of the 370 evaluated links according to their symmetrical property

	Symmetrical	**Non-symmetrical**	**Total**
same	92 (98%)	127 (77%)	**219 (85%)**
related	2	10	**12**
unrelated	0	27	**27**
related + unrelated	2 (2%)	37 (23%)	**39 (15%)**
can't tell	36	76	**112**
Total	**130**	**240**	**370**

This result falls in line with Bernard Vatant's suggestion (see [Ding et al., 2010a]) that an `owl:sameAs` is not symmetric, and that `owl:sameAs` assertions should be supported reciprocally by both owners of the resources connected by an `owl:sameAs` link, in order to be considered strongly equivalent.

Results Interpretation.

The experiments conducted in this paper, on a subset of 28 billion unique triples of the LOD Cloud, shows that there exist several incorrect `owl:sameAs` statements in the Web of Data. These erroneous identity statements have led to the false equivalence of many unrelated terms (e.g. Dublin, Coca-Cola, and Albert Einstein), and many related terms (e.g. Barack Obama the person, and his administration). With a total runtime of 11 hours, these experiments show that an error degree of every existing identity link in the LOD Cloud can be computed in practice. Our manual evaluation of these error degrees suggests that:

1. **our error degree can validate a large number of identity links in the LOD Cloud.** Around 555 million `owl:sameAs` (99.7%) have an error degree ≤ 0.99. With a precision of 88.6% in validating `owl:sameAs` links, our results suggest that our approach can correctly validate a large number of `owl:sameAs` links in the LOD. When higher precision is required over the recall, one could consider identity links with an error degree below 0.4 (manual evaluation suggest 100% precision), which refer

to 73% of the owl:sameAs links in the LOD Cloud (~ 405M).

2. **our error degree can detect numerous erroneous identity links in the LOD Cloud.** Around 1.2 million owl:sameAs links have an error degree > 0.99. With a precision varying between 40 and 73.3% depending on the the equality set's size, our results suggest that by discarding links with an error degree > 0.99, our approach can remove between 480K to 880K incorrect identity statements in the LOD.

3. **our approach can give an approximation on the quality of identity links in the LOD Cloud.** Around 450M owl:sameAs in the LOD are symmetrical (225M edges in the identity network with a weight of 2). With a 98% probability of correctness, the results suggest that (i) around 10M owl:sameAs statements are erroneous. From the remaining 106M non-symmetrical statements, there exist around 105M owl:sameAs with an error degree ≤ 0.99, and with a 88.6% probability of correctness, the results suggest that (ii) an additional 12M owl:sameAs are probably erroneous. With an erroneous probability varying between 40 and 88% depending on the equality set size, the results finally suggest that (iii) 480 to 880K additional statements with an error degree > 0.99 are probably erroneous. Therefore, relying on the error degree and the symmetry of the owl:sameAs statements in the LOD, we estimate that there could be around 22.5M erroneous owl:sameAs, representing around 4% of the total owl:sameAs statements in the LOD. This number is quite close to [Hogan et al., 2012]'s estimation[10] that 2.8% of owl:sameAs links are erroneous, and much more optimistic than [Halpin et al., 2010]'s estimation that around 21% of owl:sameAs links on the Web are incorrect, and [Cuzzola et al., 2015]'s estimation of 61% where they found 251 incorrect links out of 411 owl:sameAs.

We are aware that these numbers are just an estimation suggested by the error degree distribution and the symmetry of the existing owl:sameAs links in the LOD Cloud, and the manual evaluation of around 300 owl:sameAs links in total (from a total of 558.9M statements).

4.5 Conclusion

In this chapter, we have presented an approach for detecting erroneous owl:sameAs statements in RDF graphs. Our approach is uniquely based on the

[10] based on the manual evaluation of 1000 pairs from the same equivalence class (i.e. not necessarily explicitly owl:sameAs)

topology of the identity network itself, with no other assumption on the graph. In order to illustrate its ability to scale, we have evaluated our approach on a subset crawled from the LOD containing 28 billion triples, with over 558 million `owl:sameAs` statements. With an accuracy of 86%, the manual evaluation of around 300 `owl:sameAs` links shows that the here introduced error degree can indeed be used for distinguishing between correct and incorrect `owl:sameAs` statements. The experiments also show that an error degree for each identity link in the LOD Cloud can be computed in practice, with a total runtime of 11 hours on an a regular laptop. The error degree of all the `owl:sameAs` statements are available on our identity Web service (`https://sameAs.cc`), which will allow others to replicate, check, and hopefully improve upon the here presented results.

In the following, we describe how the here presented approach can be evaluated in comparison with the approaches presented in Section 2.3, in terms of accuracy, precision, recall, transparency and feasibility in the LOD:

Accuracy. The manual evaluation of around 300 `owl:sameAs` links suggest that our approach can correctly classify an identity link with an 86% accuracy. These results are in line with some of the best presented approaches in terms of accuracy [CudreMauroux et al., 2009, Acosta et al., 2013], with an accuracy of 90%, 94% respectively. However, these approaches were tested on a synthetic graph of 24K links, a set of 95 links, respectively, with all of these approaches also requiring some assumptions on the data (source trustworthiness or some descriptions for each resource).

Precision. Out of the 30 manually evaluated links with an error degree > 0.99, 22 links were judged as erroneous. This evaluation suggests an average precision of 73% in detecting erroneous identity links, ranging from 40% to 88% depending on the equality sets' size. The here reported precision is lower on average compared to [Hogan et al., 2012, Cuzzola et al., 2015, Papaleo et al., 2014], who respectively report precisions of 85%, 93% and 88% (on one out of 3 linksets). However, these approaches respectively require the presence of logical inconsistencies, textual descriptions, or ontology mappings.

Recall. Based on the identified threshold of 0.99, the detection of 725 out of the 780 erroneous links we injected in the LOD shows a recall of (93%). These results suggest some of the highest recalls with regards to existing approaches, with the exception of [Papaleo et al., 2014] who have obtained a recall of 100% on a particular linkset of 112 `owl:sameAs` links, while requiring ontology mappings and the presence of specific types of properties.

Transparency. In addition to the crowdsourcing approach proposed by [Acosta et al., 2013], we are the second approach that allows the replica-

tion of the experiments, by using a public dataset, publishing the links score with our gold standard, and making our tool publicly available.

Feasibility in the LOD. In contrary to existing approaches, the here presented experiments have indeed proven the feasibility of our approach in the LOD. In terms of scalability, we have improved the state of the art by an order of magnitude (compared to [de Melo, 2013] and [Valdestilhas et al., 2017], with datasets of 25M and 19M respectively). In addition, the here present approach relies only on the community structure of the `owl:sameAs` links, and requires no additional assumptions on the data, which makes it highly applicable in the context of the Web.

Now that the replications of misusing `owl:sameAs` are clear and alarming in the here computed equivalence closure, we can see the necessity of having new types of identity relations that can accurately interpret the semantics of identity intended by the user, without suffering from the identified problems discussed in Chapter 2. In the next chapter, we introduce a new contextual identity relation, with an approach for automatically detecting these contextual identity links, allowing to replace in certain cases the erroneous use of `owl:sameAs`.

CHAPTER 5
CONTEXTUAL IDENTITY RELATION

This chapter is based on the following publications:

- Joe Raad, Nathalie Pernelle, and Fatiha Saïs. "Detection of Contextual Identity Links in a Knowledge Base". In *Proceedings of the Knowledge Capture Conference*, p. 8. ACM, 2017.
- Joe Raad, Nathalie Pernelle, and Fatiha Saïs. "Détection de liens d'identité contextuels dans une base de connaissances". In *IC 2017-28es Journées francophones d'Ingénierie des Connaissances*, pages 56–67, 2017 (best paper award).

In the previous chapter, we have seen that there exist several erroneous identity links in the Web, estimating that around 4% of the `owl:sameAs` statements in the LOD Cloud are incorrect. While some of the detected `owl:sameAs` are fundamentally erroneous, linking two completely unrelated terms such as the country *Bolivia* and the scientist *Albert Einstein*, some of the links we investigated relate two different, but closely related terms that are considered the same in some contexts but not in others. Such cases are quite common in datasets that describe scientific experiments, where data are collected by different scientists, and the experiments' circumstances and participants (e.g. products, materials, etc.) tend to change, even slightly, from one experiment to another. Therefore, individuals can rarely be declared the same in all contexts, as the notion of identity might vary depending on the context. For instance, in some applications, the fact that two drugs share the the same chemical structure is sufficient to consider them as equivalent (in a scientific context), while in other commercial applications, it is also necessary that these drugs share the same name. Likewise, two lemonades with different quantity but equal proportions of lemon, water and sugar can be considered the same in a gustatory context, and different in the context of an energetic and nutritional study. The standing practice for linking such terms is the use of weaker notions of relatedness, such as `rdfs:seeAlso` and `skos:exactMatch`, with more than 169M and 566K triples respectively asserted in the LOD Cloud (see section 2.4 for a list of weaker identity predicates). However, these relations have limited semantics, and do not explicit the contexts in which the related terms can be substituted, thereby limiting reasoners in drawing inferences.

Given that the classical notion of identity, standardized in the `owl:sameAs` predicate, is highly problematic (see chapter 2), and given the limit of existing

properties in offering alternative semantics for identity with respect to a given context (see section 2.4), we propose in this chapter a novel approach for representing and detecting contextual identity links. More specifically this chapter makes the following contributions:

1. It introduces a new relation for expressing contextual identity between two class instances. In this alternative notion, the contexts in which the identity holds are defined and explicit to the user with regard to a domain ontology. For defining the contextual identity, this chapter defines the notion of global context, their order relations, and the conditions that should be fulfilled for declaring an identity between two given instances in a certain (global) context.
2. It presents an algorithm for detecting the most specific global contexts in which a pair of instances are identical. This algorithm can also be guided by a set of semantic constraints provided by experts, in order to filter irrelevant identity contexts.

The rest of this chapter is structured as follows. Section 5.1 presents the contextual identity relation and defines the criteria for identity. Section 5.2 presents our approach for automatically detecting the contextual identity links in an RDF knowledge graph, and Section 5.3 concludes.

5.1 Contextual Identity Definition

In this chapter we present a new approach for discovering contextual identity relationships in RDF knowledge graphs. The approach aims at detecting identity links that are valid in certain contexts, defined as sub-ontologies of the domain ontology. In this section, we present the considered RDF knowledge graphs, the problem statement, and introduce the contextual identity relation and the necessary notions for defining it.

5.1.1 RDF Knowledge Graph

In this approach, we consider knowledge graphs where the ontology is represented in RDFS (Resource Description Framework Schema), and the data represented in RDF[1].

Definition 10 (RDF Knowledge Graph) A knowledge graph $\mathcal{B}$ is defined by a couple $(\mathcal{O}, \mathcal{F})$ where:

[1] `https://www.w3.org/RDF/`

- $\mathcal{O} = (\mathcal{C}, \mathcal{P}, \mathcal{A})$ represents the conceptual model of the knowledge graph, defined by a set of classes $\mathcal{C}$, a set of properties $\mathcal{P}$, and a set of axioms $\mathcal{A}$ such as the subsumption relations between classes, and the domains and ranges axioms. We use the following notation for expressing subsumption relations: $c_2 \sqsubseteq c_1$ for expressing that the class c_2 is subsumed by the class c_1 (i.e. c_2 is more specific than c_1).
- $\mathcal{F} = \{(s, p, o)\}$ is a collection of triples consisting of the resource being described (subject s), a relationship (predicate p), and a relationship value (object o). Identifiers for p, s and o are IRIs, except for the object o which can also be a literal[2] (e.g. a string or any other XML-sanctioned datatype). We note $\mathcal{I}^{\downarrow}$ the set of instances i of a class c.

5.1.2 Problem statement

The problem of detecting contextual identity links can be defined as follows: given a knowledge graph $\mathcal{B} = (\mathcal{O}, \mathcal{F})$ and a set $\mathcal{I}^{tc}$ of instances of a target class tc of the ontology $\mathcal{O}$, find for the set of all instance pairs $(i_1, i_2) \in (\mathcal{I}^{tc} \times \mathcal{I}^{tc})$ the most specific contexts in which (i_1, i_2) are identical. A context is defined as a sub-ontology of $\mathcal{O}$, which represents the vocabulary (i.e. a set of classes and properties) in which two instances are considered as identical.

For instance, in the example depicted in Figure 5.1, the two instances $pr3$ and $pr4$ of the target class *Process* can be seen as identical when all the ontology's properties and classes are considered. On the other hand, the two instances $pr1$ and $pr2$ can be considered as identical in two distinct contexts. In a first context, we can consider all the devices composing the drugs and for every device we consider its volume. However, in this context, the description of a volume is reduced to the measure unit (i.e. we do not consider the property *hasValue*). A second context in which these two processes are identical is the context where we take into account the volume of the *Bioreactor* described by its value and its measure unit, but we only consider the presence of the *Pump* in the processes without considering its volume.

We note that the properties taken into account for comparing the instances of the class *Volume* should not vary according to whether we are comparing the volume of the *Bioreactor* or that of the *Pump*. Hence, it is not a task of calculating the most specific graph shared by two instances of the class *Process*, where the classes' descriptions could vary according to the considered instances. In addition, and in order to guarantee a certain semantic uniformity, we want to guarantee that if a property p of a class c appears in a context, then it must be instantiated and has identical values (up to a renaming of the instance's IRI).

[2]We do not consider blank nodes in this work.

In order to improve the efficiency of our approach and the relevance of the contexts, we propose to take into account certain experts' knowledge during the detection of the contexts. Contextual identity links are not necessarily of interest for all classes (e.g. instances of the class *Volume*), but for only one or more target classes whose identity links are of interest to the considered application (e.g. processes involved in an experiment). We thus consider knowledge that a property or a class can be ignored, that two properties must appear together (e.g. *hasValue* with *hasUnit*) or that a property must necessarily appear in a context.

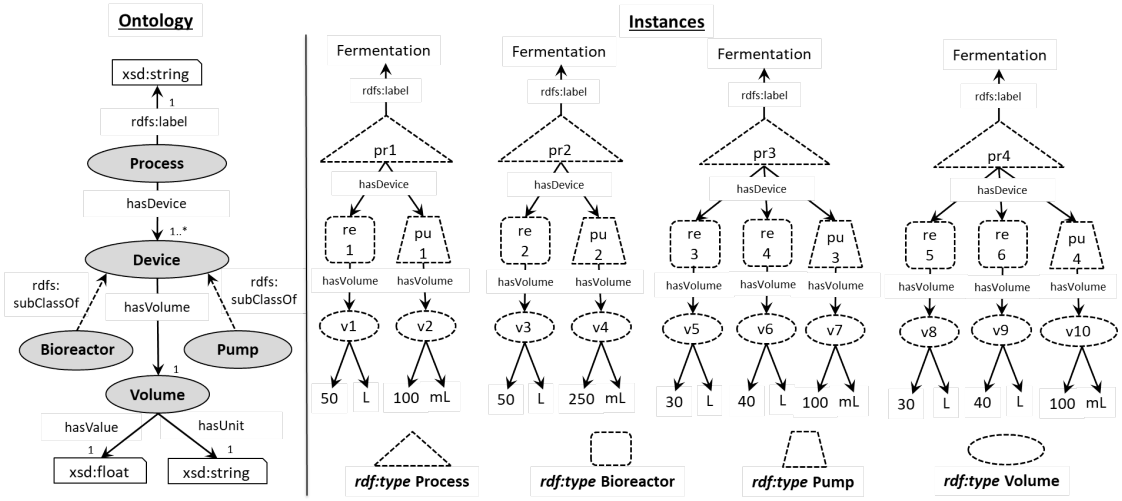


Figure 5.1: An extract of ontology O, with four instances of the target class *Process*.

5.1.3 Identity Contexts

For formally defining the notion of identity contexts, we firstly introduce the set of classes *DepC* that can be involved in the identity contexts. Then, we formally define the notion of global context and the contextual identity relation that expresses that two instances are identical in a given global context.

A. Descriptive Classes

The set of descriptive classes, noted *DepC*, represents the set of classes that may appear in the identity contexts. It is a subset of the ontology classes that are instantiated in the knowledge graph. Specifically, *DepC* is composed of the most general classes (in the sense of the subsumption relationship) of the ontology O among the explicitly instantiated classes in $\mathcal{F}$ (i.e. the *rdf:type* is not inferred). In the following, we note *directType(i,c)* the class c explicitly stated as the *rdf:type* of the instance i in $\mathcal{F}$.

Definition 11 (Descriptive Classes) A subset of instantiated classes c_i of $\mathcal{B}$ such that:

$$DepC = \{c_i \in C \mid \nexists c_j \in C \; s.t. \; \exists x, directType(x, c_j) \; and \; c_i \sqsubseteq c_j\}$$

Example (Descriptive Classes). In Figure 5.1, *DepC* contains all the classes of the graph except of the class *Device* which is not instantiated. Therefore, the instances *re*1 and *pu*1 will be uniquely considered as of type *Bioreactor* and *Pump*, respectively.

B. Global Context

A global context is a connected sub-ontology of O. It is composed of a set of classes and properties of O, and a set of axioms. In a global context, these axioms are limited to a set of constraints on the properties' domains and ranges.

Definition 12 (Global Context) A global context is a sub-ontology GC_u=(C_u, P_u, A_u) of O such that $C_u \subseteq DepC$, $P_u \subseteq P$, and A_u is a set of domain and range constraints that are more specific than those described in A: $\forall p \in P_u$, $domain_u(p) \sqsubseteq domain_O(p)$ and $range_u(p) \sqsubseteq range_O(p)$.

Example (Global Context). In Figure 5.1, there exist many possible global contexts. We present one:
GC_1=(C = {*Process*, *Bioreactor*, *Pump*, *Volume*},
P = {*hasDevice*, *hasVolume*, *hasUnit*},
A = {*domain*(*hasDevice*) = *Process*, *range*(*hasDevice*) = *Bioreactor* $\sqcup$ *Pump*,
domain(*hasVolume*) = *Bioreactor* $\sqcup$ *Pump*, *range*(*hasVolume*) = *Volume*,
domain(*hasUnit*) = *Volume*, *range*(*hasUnit*) = *xsd* : *string*})

C. Order Relation between Global Contexts

We define here the order relation between the global contexts, by relying on the inclusion of the sets of properties and classes. Thanks to this order relation, the set of all global contexts of a target class *tc* can be represented as a lattice of contexts.

Definition 13 (Order Relation between Global Contexts) Let $GC_u = (C_u, P_u, A_u)$ and $GC_v = (C_v, P_v, A_v)$ be two global contexts. The context GC_u is more specific than GC_v, noted $GC_u \leq GC_v$, if $C_v \subseteq C_u$, $P_v \subseteq P_u$, and $\forall p \in P_v$, $domain_v(p) \sqsubseteq domain_u(p)$ and $range_v(p) \sqsubseteq range_u(p)$.

Example (Order Relation between Global Contexts). $GC_1 \leq GC_2$, with GC_2 =
GC_2=(C = {*Process*, *Bioreactor*}, P = {*hasDevice*},
A = {*domain*(*hasDevice*) = *Process*, *range*(*hasDevice*) = *Bioreactor*})

D. Contextual Description of Instances according to a Global Context

In our approach, two instances are considered as identical in a given global context, when all the properties described in this context are instantiated for both instances, and when theses descriptions are the same. Before defining the contextual identity relationship, we firstly define the notion of contextual description of a target class instance.

Definition 14 (Contextual Description according to a Global Context) Given a set of RDF triples $\mathcal{F}$, a global context $GC_u = (C_u, P_u, A_u)$ and an instance i of a target class tc, a contextual description G_i of i in GC_u is the maximal set of triples that describe i in $\mathcal{F}$ such that:

- G_i forms a connected graph that contains at least one triple where i is a subject or an object
- $\forall\ t = (s, p, o) \in G_i$, $p \in P_u$, $directType(s) \sqsubseteq domain_u(p)$ and $directType(o) \sqsubseteq range_u(p)$
- $\forall\ j$ a class instance of G_i, and $\forall p \in P_u$ such as $directType(j) \sqsubseteq domain_u(p)$, then $\exists\ t_a = (j, p, k) \in G_i$, with $directType(k) \sqsubseteq range_u(p)$

Example (Contextual Description according to a Global Context). Figure 5.2 presents an extract of the ontology O, the global context GC_1, and the contextual descriptions G_{pr1} and G_{pr2} of $pr1$ and $pr2$ respectively in GC_1.

5.1.4 Contextual Identity

From two contextual descriptions of two class instances, we want to define in which conditions (i.e. contexts) these two instances are considered identical. In this work, we consider that properties are local complete: if a property p is instantiated for a given class instance i, we consider that all its property values are declared for this instance in the knowledge graph.

Since a local completeness is assumed, two instances can be considered as identical when the contextual graphs, formed by the contextual descriptions, are isomorphic up to a renaming of the instance's IRI. Note that since some classes can be removed from the global context, this constraint can in fact be considered class by class.

Definition 15 (Identity in a Global Context) Given a global context GC_u, a pair of instances i_1 and i_2 are identical in GC_u, noted $identiConTo_{<GC_u>}(i_1, i_2)$, only if the two graphs G_{i_1} and G_{i_2}, that represent the contextual descriptions of i_1 and i_2 respectively, are isomorphic up to a rewriting of the IRI of the class instances, and considering equality for literals.

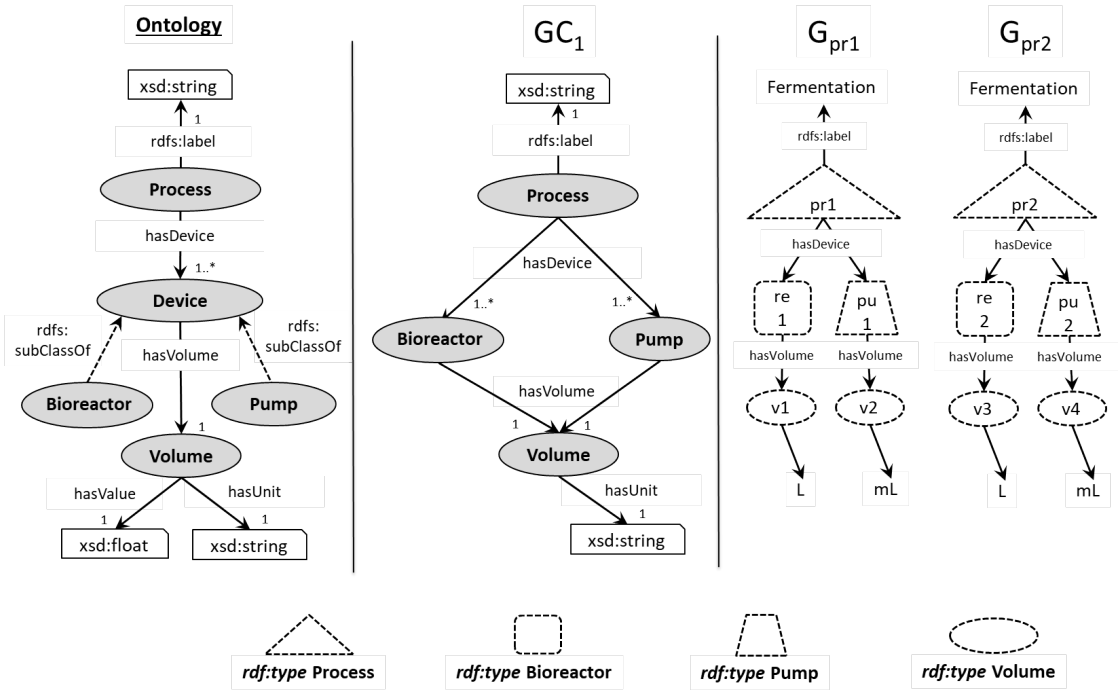


Figure 5.2: An extract of the ontology $\mathcal{O}$, the global context GC_1, and the contextual descriptions G_{pr1} and G_{pr2} of $pr1$ and $pr2$ respectively in GC_1.

Example (Identity in a Global Context). Given the following global context GC_3:
GC_3=(C = {$Process$, $Bioreactor$, $Pump$, $Volume$},
P = {$hasDevice$, $hasVolume$, $hasValuee$, $hasUnit$},
A = {$domain(hasDevice) = Process$, $range(hasDevice) = Bioreactor \sqcup Pump$,
$domain(hasVolume) = Bioreactor$, $range(hasVolume) = Volume$,
$domain(hasValue) = Volume$, $range(hasValue) = xsd : float$,
$domain(hasUnit) = Volume$, $range(hasUnit) = xsd : string$})

The identity link expressing that $pr1$ and $pr2$ are identical in the global context GC_3 is noted $identiConTo_{<GC_3>}(pr1, pr2)$. This identity relation takes into account the volume of the *Bioreactor* described by its value and its measure unit, but only considers the presence of the *Pump* in the processes without considering its volume. In addition, $pr1$ and $pr2$ are also identical in the context GC_1 (example from Definition 12), where we consider all the devices composing the drugs and for every device we consider its volume, but reduce the description of a volume to its measure unit. This identity relation is noted $identiConTo_{<GC_1>}(pr1, pr2)$.

These two contexts are not the only contexts where $pr1$ and $pr2$ are identical, as they are also identical in GC_2 (example from Definition 13). However, GC_1 and GC_3 are the most specific contexts in which these two instances are identical. Since more general identity links such as $identiConTo_{<GC_2>}(pr1, pr2)$ can be inferred using the order relation between global contexts, the contextual

identity relations will only be specified for the most specific global context(s):

Given GC_u and GC_v two global contexts, with $GC_u \leqslant GC_v$, then $identiConTo_{<GC_u>}(i_1, i_2) \Rightarrow identiConTo_{<GC_v>}(i_1, i_2)$.

5.2 Detection of Contextual Identity Links

Now that the contextual identity relation is defined, the goal of our approach is to determine for each pair of instances, the contexts in which they are identical. We propose an algorithm named DECIDE (DEtection of Contextual IDEntity), that takes as input a target class tc, and determines for each pair of instances (i_1, i_2) $\in I^{tc} \times I^{tc}$, the set of the most specific global contexts in which the identity relation is true. This algorithm is composed of three main steps: (i) selecting the set of descriptive classes $DepC$ (Definition 11), (ii) constructing similarity graph(s), and finally (iii) calculating the most specific global context(s). This section presents the algorithm DECIDE, and the necessary notions.

Our approach for detecting contextual identity links relies on the notion of local context that composes the global contexts.

Definition 16 (Local Context) A local context of a class c is a global context that is limited to the properties in which c is the domain or range.

We distinguish between the outgoing local contexts $LC_k^{out}(c)$ that captures the properties in which c is the domain, and the incoming local contexts $LC_k^{in}(c)$ that captures the properties in which c is the range:

– $LC_k^{out}(c) = (C_k^{out}, P_k^{out}, A_k^{out})$, a local context where $\forall p \in P_k^{out}$, $domain(p) = c$.
– $LC_k^{in}(c) = (C_k^{in}, P_k^{in}, A_k^{in})$, a local context where $\forall p \in P_k^{in}$, $range(p) = c$.

5.2.1 Experts Knowledge

In order to filter out some irrelevant contexts, this algorithm takes in consideration certain expert knowledge when it is available. This knowledge, given as a set of constraints, concerns the presence or the co-occurrence of certain classes, properties and/or axioms. More precisely, an expert can specify three types of constraints:

Unwanted Properties (UP). Refer to properties that experts want to discard in the identity contexts (i.e. global contexts). Such constraints can be used

when property values correspond to unstructured text, known to be particularly heterogeneous, or when the property subjects or objects are evolutive or insignificant to compare two instances for a given task. In such cases, an expert can declare that a property p is unwanted for a given domain c_i (or a particular range c_j) by adding a constraint $up = (c_i, p, *)$ (respectively $up = (*, p, c_j)$) in UP. When a property is unwanted in all domains and ranges, the constraint $(*, p, *)$ can be used. In such cases, $p \notin P$ in all contexts.

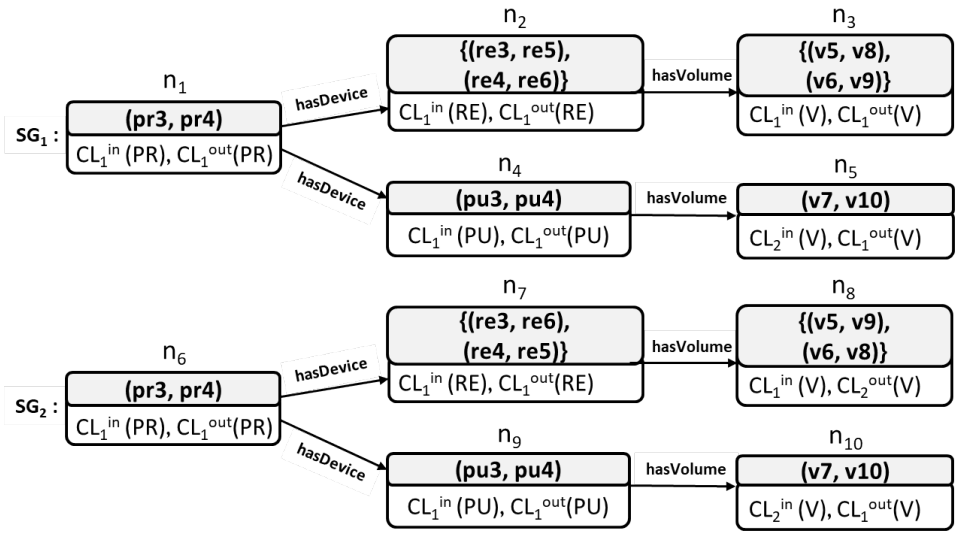
Necessary Properties (NP). A necessary property is a constraint noted $np = (c_i, p, *)$ or $(*, p, c_j)$. When such constraints are added to NP, only global contexts where the property $p \in P$, with $c_i \in domain(p)$ (respectively $c_j \in range(p)$) are considered.

Co-occurring Properties (CP). A co-occurrence constraint $cp = \{(c_i, p_1, *), ..., (c_i, p_n, *)\}$ can be declared to guarantee that a certain class c_i will be either declared as the domain (or range) of *all* the properties indicated in the constraint, or will be declared for *none* of them. For instance, to declare that the volume's value has no meaning without its measure unit (and vice versa), an expert can add the constraint $cp_1 = \{(Volume, hasValue, *), (Volume, hasUnit, *)\}$. Meaning that no context will contain the axiom $(domain(hasValue) = Volume)$ without also containing the axiom $(domain(hasUnit) = Volume)$, and vice-versa.

5.2.2 Contextual Identity in RDF

A global context is represented as a named graph [Carroll et al., 2005], with each named graph containing the considered axioms of the ontology and the identity statements valid in this context. A contextual identity assertion between two instances i_1 and i_2 in a named graph, indicates that this context represents the most specific global context in which these two instances are identical (Definition 15). Since equality is used for literals identity, the here presented contextual identity links are symmetric, transitive, and reflexive. The order relation between the global contexts is represented in the original graph using the Named Graphs Vocabulary[3], with the relation `rdfg:subGraphOf`. An example of the output of `DECIDE` on the Figure 5.1 knowledge graph is available at `https://github.com/raadjoe/DECIDE_v2/tree/master/Example`.

[3] `http://www.w3.org/2004/03/trix/rdfg-1/`



CL_1^{in} (PR) = ∅
CL_1^{out} (PR) = (A={domain(hasDevice) = Process, range(hasDevice) = Bioreactor ⊔ Pump, domain(rdfs:label) = Process, range(rdfs:label) = xsd:string})

CL_1^{in} (PU) = (A={domain(hasDevice) = Process, range(hasDevice) = Pump})
CL_1^{out} (PU) = (A={domain(hasVolume) = Pump, range(hasVolume) = Volume})

CL_1^{in} (RE) = (A={domain(hasDevice) = Process, range(hasDevice) = Bioreactor})
CL_1^{out} (RE) = (A={domain(hasVolume) = Bioreactor, range(hasVolume) = Volume})

CL_1^{in} (V) = (A={domain(hasVolume) = Bioreactor, range(hasVolume) = Volume})
CL_1^{out} (V) = (A={domain(hasValue) = Volume, range(hasValue) = xsd:float, domain(hasUnit) = Volume, range(hasUnit) = xsd:string})

CL_2^{in} (V) = (A={domain(hasVolume) = Pump, range(hasVolume) = Volume})
CL_2^{out} (V) = (A={domain(hasUnit) = Volume, range(hasUnit) = xsd:string})

Figure 5.3: The two possible similarity graphs for the pair (pr_3, pr_4). For simplicity reasons, C, and P are not represented in this Figure for all the local contexts.

5.2.3 DECIDE - Algorithm for Detecting Contextual Identity

The goal of the algorithm DECIDE is to determine for each pair of instances (i_1, i_2) $\in I^{tc} \times I^{tc}$ of a target class tc given by the user, the set of the most specific global contexts in which the identity relation is valid. DECIDE requires to have the knowledge graph $\mathcal{B}$ and the target class tc as input. In addition, the set of constraints UP, NP, CP can also be given as input, when available. Algorithm 2 details the approach of detecting contextual identity links, composed of the three following main steps:

i. Collect the set of Descriptive Classes. The set $DepC$ (see Definition 11) is collected for indicating the abstraction level (in the sense of the subsumption relationship) of the classes that should be considered while construct-

Algorithm 2: DECIDE: DEtection of Contextual IDEntity

Input:
– $\mathcal{B}$: the RDF knowledge graph
– tc: the target class
– $K(NP, UP, CP)$: the expert constraints
Output: $MSContexts$: set of most specific global contexts for each pair of instances

```
1  DepC ← getDepC(B) ;
2  I^tc ← list of instances of directType(tc) ;
3  foreach ( (i1, i2) ∈ I^tc × I^tc with i1 ≠ i2) do
4      GCset ← ∅;
5      SGset ← constructSimilarityGraphs(i1, i2, DepC, K, B) ;
6      foreach (SG ∈ SGset) do
7          n0 ← SG.getNode(i1, i2) ;
8          N ← ∅; a ← ∅; GC ← ∅; LCset ← ∅;
9          GC ← generateGC(n0, a, GC, LCset, N, SG, K) ;
10         if (∄ GC1 ∈ GCset, such that GC1 ≤ GC) then
11             GCset.add(GC) ;
12         if (∃ GC2 ∈ GCset, such that GC ≤ GC2) then
13             GCset.remove(GC2) ;
14         foreach (LC ∈ LCset) do
15             GC ← ∅; GC.add(LC) ;
16             GC ← generateGC(n0, a, GC, LCset, N, SG, K);
17             if (∄ GC1 ∈ GCset, such that GC1 ≤ GC) then
18                 GCset.add(GC) ;
19             if (∃ GC2 ∈ GCset, such that GC ≤ GC2) then
20                 GCset.remove(GC2) ;
21     MSContexts.add(GCset, (i1, i2));
22 return MSContexts;
```

ing the similarity graphs, and consequently generating the global contexts.

For instance, the set $DepC$ of the knowledge graph presented in Figure 5.1 will contain the following classes: $\{Process, Bioreactor, Pump, Volume\}$. The class *Device* is not considered in the global contexts, since it is not directly instantiated.

ii. Construct the Similarity Graph(s). For each pair of instances of the target class tc, one or more similarity graphs are constructed. A similarity graph represents a set of possible mappings of the class instances for each property appearing in their RDF descriptions. A node n_i of the similarity graph represents a set of mapped pair of instances of a class c in $I^c \times I^c$. In addition, each node of the similarity graph contains the most specific outgoing local context $LC_{out}(c)$ and the most specific incoming local context $LC_{in}(c)$, in which they are identical (according to Definition 15). These local con-

texts verify the set of constraints K given by the experts. The construction of each similarity graph is directed by the source node representing the pair of instances of the target class. The direction of the arcs indicates the domains and ranges of the considered properties in the axioms of the corresponding local contexts.

For instance, the similarity graphs corresponding to the pair of instances (pr_3, pr_4) of the target class *Process* are presented in Figure 5.3. In this example, the property *hasDevice*, having multiple values for the same class (*Bioreactor*), has led to the construction of two similarity graphs. SG_1 considers the mapping of the instance re_3 with re_5, and the instance re_4 with re_6, while the similarity graph SG_2 considers the mapping of re_3 with re_6, and re_4 with re_5. The nodes corresponding to the volumes of the Bioreactors are associated with different outgoing local contexts, depending on the considered mapping.

iii. Generate the Most Specific Global Context(s). Relying on the constructed similarity graphs, a global context GC is generated using the set of the local contexts, insuring the presence of no more than one local context per class in the same global context. The most specific global contexts are generated using the function *generateGC*, which traverses the similarity graph SG using a depth-first search algorithm. This function, described in Algorithm 3, aims to add for each node its most specific outgoing local context $LC_{out}(c)$, already calculated in SG, to the current global context GC (i.e. the most specific global context). Let n be the current node during the algorithm's traversal of the similarity graph, looking at its outgoing local context we distinguish between three cases:

1. If GC does not contain a local context $LC_{ex}(c)$ for the class c, or if GC contains $LC_{ex}(c)$ with $LC_{ex}(c)$ equal to the local context $LC_n(c)$ of n, then $LC_n(c)$ is added to GC. The function *generateGC* is then recursively recalled for each node n_{dst} in SG, such as there is an edge between n and n_{dst}.
2. If GC contains a local context $LC_{ex}(c)$ for the class c, and $LC_n(c)$ is more specific than $LC_{ex}(c)$, then the function *generateGC* is recursively recalled for each destination node n_{dst} in SG, such as there is an edge between n and n_{dst} labelled p, and exists an axiom a in GC with $a = \{domain(p) = c$ and $directType(n_{dst}) \sqsubseteq range(p)\}$ or $a = \{range(p) = c$ and $directType(n_{dst}) \sqsubseteq domain(p)\}$.
3. If GC contains a local context $LC_{ex}(c)$ for the class c, and $LC_n(c)$ is not more specific than $LC_{ex}(c)$, then the function *generateGC* is not recalled for this graph node. Moreover, the domain representing the type of the node source and the range representing the class c of the property p that led to this graph element will be removed from the current global context. Finally, GC is updated, verifying that the axioms of

the graph still forms a connected component, and verifying that the expert constraints are all still respected.

In both cases (2) and (3), $LC_n(c)$ and the most specific local context that generalizes $LC_n(c)$ and $LC_{ex}(n)$ will be added to a list $LCset$, in order to guarantee the presence of these local contexts in other global contexts. Therefore, resulting in several most specific global contexts for the same pair.

5.2.4 Contextual Identity Links Examples

This section presents some examples explaining the output of DECIDE in several cases (e.g. case where the domains and the ranges of a property are the same). These examples will help clarify some aspects of the algorithm, and discuss the benefits and limits of the here proposed identity relation. For this, we rely on the ontology extracts and instances of the 'Processes' example in Figure 5.1 and the 'Drugs' example in Figure 5.4.

Target Class Process (Figure 5.1)

(pr1, pr2). When applied on the pair $(pr1, pr2)$, DECIDE would result in a single similarity graph, since there is only one possible mapping of the class instances. This similarity graph results in two global contexts GC_1 (see Example of Definition 12) and GC_3 (see Example of Definition 15), representing the most specific contexts in which these two processes are identical.

(pr3, pr4). When applied on the pair $(pr3, pr4)$, DECIDE would result in two similarity graphs, both presented in Figure 5.3. Since the global context resulting from SG_1 is more specific than the one generated from SG_2, the output of DECIDE is one global context, in which all the ontology axioms are considered.

(pr1, pr3) and (pr1, pr4). When applied on the pair $(pr1, pr3)$ and the pair $(pr1, pr4)$, DECIDE would result in a single similarity graph, and eventually one most specific global context for each pair. Since a local completeness is assumed, the class *Bioreactor* is not considered for both pairs in the following resulting identity context:

GC_4=(C = {$Process, Pump, Volume$},
P = {$hasDevice, hasVolume, hasValue, hasUnit$},
A = {$domain(hasDevice) = Process$, $range(hasDevice) = Pump$,
$domain(hasVolume) = Pump$, $range(hasVolume) = Volume$,
$domain(hasValue) = Volume$, $range(hasValue) = xsd : float$,
$domain(hasUnit) = Volume$, $range(hasUnit) = xsd : string$})

Algorithm 3: Generate GC: Global Contexts Generation

Input:
– n: an similarity graph node
– a_s: axiom indicating the type of the node source with the property source
– GC: the current global context
– $LCset$: set of unused local contexts
– N: list of visited nodes
– SG: the similarity graph
– $K(NP, UP, CP)$: the expert constraints
Output: GC: the current most specific global context

```
1  if (n ∉ N) then
2      N.add(n) ;
3      LC_n(c) ← getOutgoingLocalContext(n) ;
4      LC_ex(c) ← GC.getExistingLocalContext(c) ;
5      if (LC_ex(c) == null or LC_ex(c) == LC_n(c)) then
6          GC.add(LC_n(c)) ; // if it does not exist
7          E^n ← SG.getEdges(n) ;
8          foreach (e_out ∈ E^n such that e_out = p(n, n_dst)) do
9              a ← {domain(p) = c, range(p) = type(n_dst)} ;
10             GC ← generateGC(n_dst, a, GC, LCset, N, SG, K) ;
11         foreach (e_in ∈ E^n such that e_in = p(n_dst, n)) do
12             a ← {domain(p) = type(n_dst), range(p) = c} ;
13             GC ← generateGC(n_dst, a, GC, LCset, N, SG, K) ;
14     else
15         if (LC_n(c) ⩽ LC_ex(c)) then
16             E^n ← SG.getEdges(n) ;
17             foreach (e_out ∈ E^n such that e_out = p(n, n_dst)) do
18                 a ← {domain(p) = c, range(p) = type(n_dst)} ;
19                 if (a ∈ GC) then
20                     GC ← generateGC(n_dst, a, GC, LCset, N, SG, K);
21             foreach (e_in ∈ E^n such that e_in = p(n_dst, n)) do
22                 a ← {domain(p) = type(n_dst), range(p) = c} ;
23                 if (a ∈ GC) then
24                     GC ← generateGC(n_dst, a, GC, LCset, N, SG, K);
25         else
26             GC.remove(a_s); // remove the source axiom from GC
27             GC ← updateGC(K, SG); // verify if GC is connected and the
                   experts constraints are satisfied
28         LCset.add(LC_n(c)) ; //if it does not already exist
29         LCset.add(intersect(LC_n(c), LC_ex(c))) ; //if it does not already exist
30 return GC;
```

(pr2, pr3) and (pr2, pr4). Similarly to the previous case, when applied on the pair ($pr2$, $pr3$) and the pair ($pr2$, $pr4$), DECIDE would result in a single similarity graph, and eventually one most specific global context for each pair. Since the volume of $pu2$ is different than the one of $pu3$ and $pu4$, the value of the class $Volume$ is not considered, resulting in the following resulting identity context:

GC_5=($C = \{Process, Pump, Volume\}$,
$P = \{hasDevice, hasVolume, hasUnit\}$,
$A = \{domain(hasDevice) = Process, range(hasDevice) = Pump,$
$domain(hasVolume) = Pump, range(hasVolume) = Volume,$
$domain(hasUnit) = Volume, range(hasUnit) = xsd : string\}$)

Target Class Drug (Figure 5.4)

In order to better investigate the output and the limitations of DECIDE, we present in Figure 5.4, a case where two properties have similar domains and ranges, and the case where a property has the same class as domain and range. This example shows the contexts in which two drugs with different names, but with the same chemical structure are considered identical. Figure 5.5 presents the similarity graph of each pair of instances of the target class *Drug*.

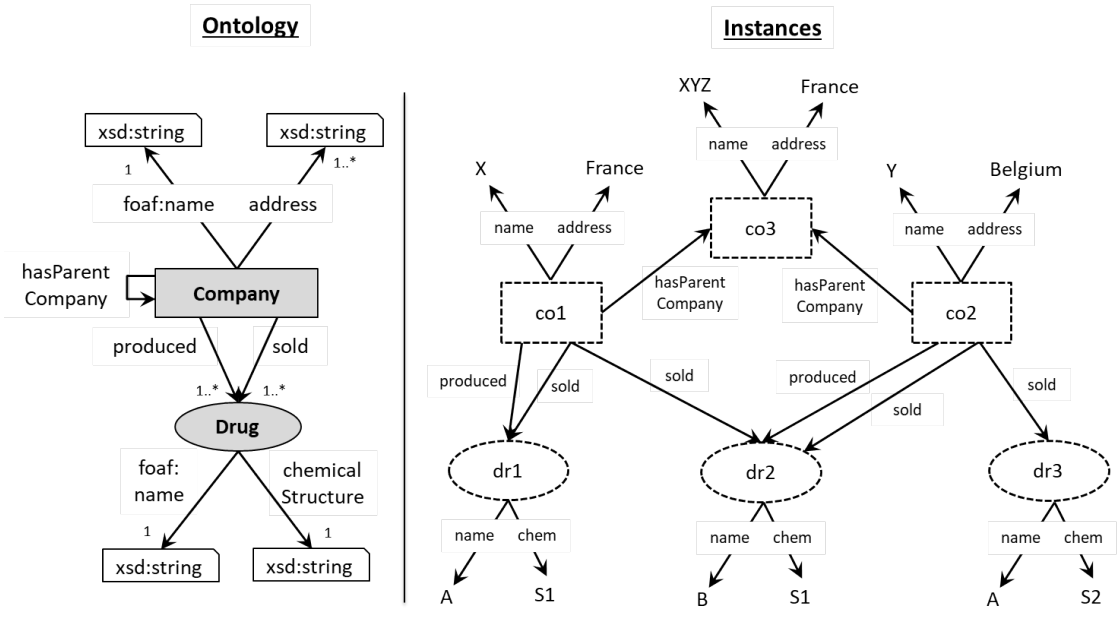


Figure 5.4: An extract of ontology O, with three instances of the target class *Drug*.

(dr1, dr2). When applied on the pair ($dr1$, $dr2$), DECIDE would result in a single similarity graph, resulting in the following most specific global context:

GC_6=(C = {$Drug, Company$},
P = {$produced$, $chemicalStructure$},
A = {$domain(produced) = Company$, $range(produced) = Drug$,
$domain(chemicalStructure) = Drug$, $range(chemicalStructure) = xsd : string$})

The interpretation of the contextual identity $identiConTo_{<GC_6>}(dr1, dr2)$ indicates that these two instances have the same chemical structure, and are both produced by companies that produce drugs with the same chemical structure. Meaning that in a scientific context where the name of the drug is irrelevant, and only the chemical structure matters, these two drugs are considered identical and these IRIs can be used interchangeably. The property *sold* is not considered in this identity context, due to the local completeness we assume in the identity definition ($dr1$ is sold by one company $co1$, while $dr2$ is sold by two companies $co1$ and $co2$). In addition, the property *hasParentCompany* is not considered in this global context, due to the instance $co3$ not having the property *produced*. An additional (most specific) global context where the former property is considered without the latter cannot exist, since the contextual descriptions G_{dr1} and G_{dr2} (Definition 14) do not form a connected graph in that case.

(dr1, dr3). When applied on the pair ($dr1, dr3$), `DECIDE` would result in a single similarity graph, resulting in the following most specific global context:

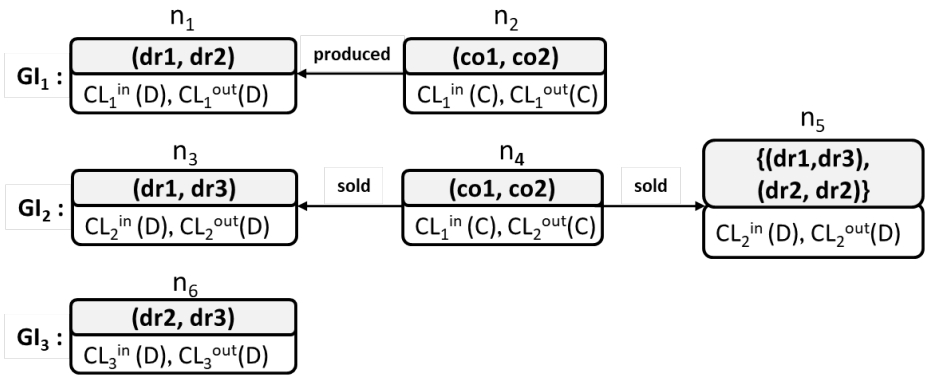
GC_7=(C = {$Drug, Company$},
P = {$sold$, $name$},
A = {$domain(sold) = Company$, $range(sold) = Drug$,
$domain(name) = Drug$, $range(name) = xsd : string$})

The interpretation of the contextual identity $identiConTo_{<GC_7>}(dr1, dr3)$ indicates that these two instances have the same name, and are both sold by companies that have sold drugs with the same name.

(dr2, dr3). Other than the fact that both $dr2$ and $dr3$ are of type *Drug*, there is no context in which these two instances can be used interchangeably.

Benefits & Limitations

An advantage of the here presented contextual identity relation is that contexts in which the identity of two instances of a target class holds are explicit. Meaning that the contexts in which these two instances can be used interchangeably are known and specified to the modeller. These contexts are not just a set of properties, as a given property can be included in a context for a subset of classes, and not be considered for other classes. For instance, the property *foaf:name* which is used for designating both the names of a *Drug* and a *Company*, is considered in GC_7 for the former and not for the latter. In addition, the specificity level of the resulting identity contexts is directly related to the required



CL_1^{out} (D) = (A={domain(chemStructure) = Drug, range(chemStructure) = xsd:string})
CL_2^{out} (D) = (A={domain(name) = Drug, range(name) = xsd:string})
CL_3^{out} (D) = ∅
CL_1^{in} (D) = (A={domain(produced) = Company, range(produced) = Drug})
CL_2^{in} (D) = (A={domain(sold) = Company, range(sold) = Drug})
CL_3^{in} (D) = ∅

CL_1^{in} (C) = ∅
CL_1^{out} (C) = (A={domain(produced) = Company, range(produced) = Drug, domain(hasParentCompany) = Company, range(hasParentCompany) = Company})
CL_2^{out} (C) = (A={domain(sold) = Company, range(sold) = Drug, domain(hasParentCompany) = Company, range(hasParentCompany) = Company})

Figure 5.5: The similarity graphs for the pairs $(dr1, dr2)$, $(dr1, dr3)$ and $(dr2, dr3)$. For simplicity reasons, C and P are not represented in this Figure for all the local contexts.

modelling choices and requirements deployed by the modeller. For instance, if the modeller is more interested in the geographic location of the companies and have modelled the data accordingly, the identity contexts would have been able to provide more specific contextual identity links. For example, declaring that $(dr1, dr2)$ are both produced by European countries, hence inferring that every EMA[4] rule considered for $dr1$ should also be considered for $dr2$.

A limitation of our proposed contextual identity relation, as the isomorphism of the instances' contextual descriptions (Definition 15), that it does not necessarily represent the most common graph in which two instances are identical. But, it represents the most specific vocabulary in which these two instances are considered identical. For instance, in the case of the pair $(dr1, dr2)$, the property *hasParentCompany* cannot be considered in the identity context despite the fact that both companies have indeed the same parent company (i.e. same IRI). Meaning that in a context where the parent company of the $dr1$ producer is causing some controversies over the production of this type of drugs (i.e. with chemical structure of S1), we lack the information that the $dr2$ producer shares the same parent company, and is identical to $dr1$ in this particular context.

[4]European Medicines Agency

5.3 Conclusion

In this chapter, we have introduced a new contextual identity relation, and proposed an approach (DECIDE) for automatically detecting these contextual identity links in an RDF knowledge graph. The approach is based on the notion of global contexts representing sub-ontologies, in which two instances are identical. The algorithm detects for each pair of instances of a target class given by the user, the most specific contexts in which this pair of instances are identical. More general contexts can be inferred from the most specific ones, thanks to the order relation that hierarchizes all the global contexts. Furthermore, this approach can take into account some experts' constraints, which can be in the form of a list of necessary properties for the identity link, list of unwanted properties, and list of properties that must occur together.

In comparison with [Beek et al., 2016], the main predecessor of this work, the contextual identity relation we propose in this chapter is more precise and expressive. Firstly, instead of solely considering the local properties describing the concerned instances (i.e. path of length 1), we consider in our contexts all the properties in the knowledge graph. This is done by propagating in the graph and considering also properties describing instances related to the concerned instances, and so on. In addition, our approach does not solely rely on the notion of properties, but also on the ontology classes and axioms. This allows us to consider a property for certain classes, and not consider it for other classes, in the same identity context. Finally, we propose an algorithm to detect, and explicitly represent in RDF, these identity contexts. This allows users to directly test and use these proposed identity links, using the code available at https://github.com/raadjoe/DECIDE_v2.

To evaluate the applicability and relevance of the here proposed contextual identity relation, we present in the next chapter an application of DECIDE on the complex case of scientific knowledge graphs. In addition, we present how the detected contextual identity links can be exploited to predict, with a certain degree of confidence, certain missing values in these knowledge graphs.

CHAPTER 6
CONTEXTUAL IDENTITY FOR LIFE SCIENCES KNOWLEDGE GRAPHS

This chapter is based on the following publications:

- Joe Raad, Nathalie Pernelle, Fatiha Saïs, Juliette Dibie, Liliana Ibanescu, and Stéphane Dervaux. "Comment représenter et découvrir des liens d'identités contextuels dans une base de connaissances : applications à des données expérimentales en science du vivant". In *Revue d'Intelligence Artificielle*, 32(3):345–372, 2018.
- Liliana Ibanescu, Juliette Dibie, Stéphane Dervaux, Elisabeth Guichard, Joe Raad. "PO^2 - A Process and Observation Ontology in Food Science. Application to Dairy Gels". In *Research Conference on Metadata and Semantics Research*, pages 155–165, 2016.

In the previous chapter, we presented a new approach for defining the identity relation. Instead of checking indiscernibility with respect to all properties, as currently adapted in the `owl:sameAs` construct, we explicitly parametrize the identity relation over certain parts of the ontology. This allows the creation of semantic links between entities that can not be declared as identical in the strict sense of identity, since they do not share all their properties, and can not be used interchangeably in all contexts. Such cases are quite common in scientific data, where experiments can rarely be declared the same, as they are mostly conducted by different scientists, in various circumstances, using similar products. This incapacity of semantically linking slightly different experiments has been a serious barrier for knowledge-based systems to fully exploit scientific data, as they are either weakly connected with little semantics (e.g. using `skos:closeMatch`), or are incorrectly declared the same (using `owl:sameAs`). In addition, the classic problems of the heterogeneity of the formats in which scientific data are published (e.g. scientific publications, Excel files, lab reports), and the terminological variations encountered across the multiple scientific datasets (e.g. synonyms, aliases, multilingualism) still remain serious barriers in fully exploiting the large amount of data produced everyday. As a way for limiting these syntactic and semantic problems, life sciences publishers became one of the most frequent adopters of Semantic Web technologies and Linked Data principles for publishing their data and encoding their knowledge. This adoption is starkly obvious in the Linked Open Data Cloud diagram, in which the life sciences knowledge graphs make up a significant portion of the cloud, with 339 out of the 1,184 knowledge graphs available in April 2018, describing life sciences data [Polleres et al., 2018].

With such significant resources already been invested in publishing life sciences data in RDF, there is an obvious and increasing interest to make use of this wealth of data for generating new insights, and discovering novel implicit associations. Working closely with experts of the French National Institute of Agricultural Research (INRA) in the context of the LIONES interdisciplinary project[1], we aim at providing them with such possibilities by making their data 'five star'. This five stars rating system, outlined by Tim Berners-Lee in 2010, provide a set of goals and incremental steps for creating high quality and freely accessible data sources:

⋆ Publish data on the Web in any format, with an open licence (e.g. PDF file)

⋆⋆ Use structured data formats (e.g. Excel file)

⋆ ⋆ ⋆ Use non-proprietary formats (e.g. CSV file instead of Excel)

⋆ ⋆ ⋆⋆ Use open standards from W3C to represent data (e.g. RDF and OWL)

⋆ ⋆ ⋆ ⋆ ⋆ Link your data to other data sets on the Web for providing context

The first four stars are relatively easy to reach, and enable some data reuse. However, users still have to handle all the semantic issues related to its integration. In order to have data that is easily discoverable, interoperable, and exploitable in knowledge-based systems, it is necessary to reach the fifth star. This final step is achieved by favouring the reuse of existing vocabularies, and expressing links with well-known semantic predicates (e.g. `rdfs:subClassOf` for subsumption relations, and `owl:sameAs` for identity relations). However, since strict identity links such as `owl:sameAs` are rarely deployable in scientific datasets, we apply our approach for detecting and expressing identity links that are semantically interpretable.

In this chapter, we introduce a new 'five star' knowledge graph for life sciences, based on scientific experiments conducted and collected from two INRA research groups. This knowledge graph firstly provide domain experts with various semantic connections between the different participants of each scientific experiment (e.g. this sensor is used to collect a measure of this product, as part of an observation conducted in a certain experiment). In addition, and by applying our contextual identity link detection approach, this knowledge graph can provide experts with different levels of identity connections between the experiments and their participants. More specifically, this chapter makes the following contributions:

1. It presents a new conceptual model that allows to model complex scientific data from different life sciences applications. This OWL ontology strikes

[1]Project funded by the Center for Data Science of the University of Paris-Saclay

a balance between the expressiveness of the underlying description logic, the reasoning efficiency, and the practicality of use by domain experts. Aiming for semantic interoperability, this ontology is designed mostly by reusing parts of existing well-established ontologies.

2. It presents a new knowledge graph for life sciences describing two different domains: the mechanisms leading to the release of flavour compounds during dairy gel consumption and their impact on global sensory perception, and the process of stabilisation of micro-organisms. This knowledge graph is constructed with a methodology that requires mutual efforts with domain experts, enriching the core conceptual model with domain specific knowledge.
3. It presents an experimental evaluation of contextual identity link detection applied on a scientific knowledge graph.
4. It presents a first use case for exploiting the detected contextual identity links for discovering certain types of rules. After the experts validation, these rules can be used to predict, with a certain degree of confidence, unobserved measures in a scientific experiment and consequently complete the knowledge graph with implicit assertions.

The rest of this chapter is structured as follows. Section 6.1 presents the five-star knowledge graph for life sciences, and describes the construction process. Section 6.2 presents the first use case of detecting contextual identity links for life sciences. In Section 6.3, we exploit the detected contextual identity links for detecting rules that can help complete the constructed knowledge graph. Section 6.4 summarizes the experiments' results, and Section 6.5 concludes.

6.1 Five Star Knowledge Graph for Life Sciences

In this section, we describe a new knowledge graph constructed in collaboration with domain experts from two INRA research groups: the BioMiP[2] team (Bioproducts, Food, Micro-organisms and Processes) and the FFOPP[3] team (Flaveur, Food Oral Processing et Perception) of the GMPA and CSGA research units, respectively. In what follows, we present the application domain and the workflow of the knowledge graph construction from Excel files.

[2] https://www6.versailles-grignon.inra.fr/gmpa_eng/Research-teams/BioMiP

[3] https://www2.dijon.inra.fr/csga/site_engl/equipe_1.php

6.1.1 Application Domain

The aim of our ongoing collaboration with the domain experts is to model semantic links between the different objects participating in, and generated by the experimental transformation processes. Once these semantic links are modelled and made explicit, this knowledge can be interrogated, analysed, and exploited in various knowledge-based tasks that can help improve the quality of the products, and limit the environmental impact caused by these processes. In this project, we deal with experimental processes from two domains:

Stabilisation of Micro-organisms. Micro-organisms are biological agents which present a large scale of applications in food domains (e.g. ferments) or in medical domains (e.g. probiotiques). With the need of concentrated micro-organisms stabilized and in ready-to-use form continuously increasing, the control of their production process has become an important issue. This production process relies on a complex system, involving several unit operations: fermentation, cooling, concentration, formulation, freezing or lyophilisation and the storage of the stabilized micro-organism. Many data have been generated from experiments on micro-organisms at different scales (from the microbial cell components to the target functionality at the population level), and at different stages of the production process by the researchers of the BioMiP team. This data are mainly collected for two specific purposes: (i) describing and archiving the conducted experimental processes, and (ii) studying the micro-organisms quality evolution, and the environmental impacts caused by these processes [Pénicaud et al., 2014]. This data is collected as part of the *CellExtraDry* project.

Release of Flavour Compounds during Dairy Gel Consumption. These experiments aim at exploring the mechanisms at the origin of and influencing food mental representation in human. More specifically the data collected by the FFOPP team study three different mechanisms: (i) the production process of French hard cheeses, where different parameters were measured (e.g. the product's pH) during each step of the production (e.g. cooling, moulding); (ii) the sensory perception during in-mouth food breakdown, with a focus on the product's texture (e.g. firmness, granularity) and taste (e.g. intensity, saltness); and finally (iii) the study of the cheese's rheological properties (e.g. the Young's modulus which measures the stiffness of the cheese) [Guichard et al., 2017]. This data is collected as part of the *Caredas* project.

6.1.2 Conceptual Model

The ontology conceptualization process follows an iterative approach, as the data model was continuously influenced by several factors, mainly the experts' different backgrounds and types of data. For instance, whilst in the cheese production process a certain observation solely results in one measure (e.g. an observation measures the pH of the studied cheese), in the micro-organism's stabilization process an observation can result in a series of measurements that are not interpretable when separated. After enumerating the important terms that should be considered in the model, and analysing a number of related ontologies, we have reached a consensus about a structure, in which the ontology concepts are grouped into the five following parts. Figure 6.1 presents the relationship between these five parts.

Processes. This part of the ontology concerns the main experimental process, the itineraries, and the different steps composing each itinerary. For instance, the process of cheese production can be conducted according to several itineraries (i.e. recipes), with each itinerary representing a specific execution of a set of interrelated steps.

Participants. This part represents the objects that participate and are deployed in a certain process (e.g. the cheese, its ingredients, the materials deployed for handling it, and the set of instructions that are followed in each step).

Observations. This part represents the observations conducted in the experimental process and the sensors deployed for performing these observations. An observation can be conducted on different scales (e.g. cellular, molecular), and can observe a product, material, step, or the whole itinerary (e.g. in the case of measuring the environmental impact).

Attributes. This part of the ontology describes the participants input characteristics (e.g. this step uses 20 grams of salt), and the observation measures (e.g. the measured pH of this cheese is 5.5).

Temporal Relations. This part focuses on the temporal aspect of the experiments, describing the dates of the experiments and the time relation between the different steps.

Aiming for semantic interoperability when designing each part of the ontology, we have reused and extended several well-established ontologies and concepts. Modelling decisions for the first three parts of the ontology were influenced by the structure of the Sensor, Observation, Sample, and Actuator (SOSA) ontology [Janowicz et al., 2018]. This ontology, developed by a joint working group of the Open Geospatial Consortium (OGC) and the W3C on

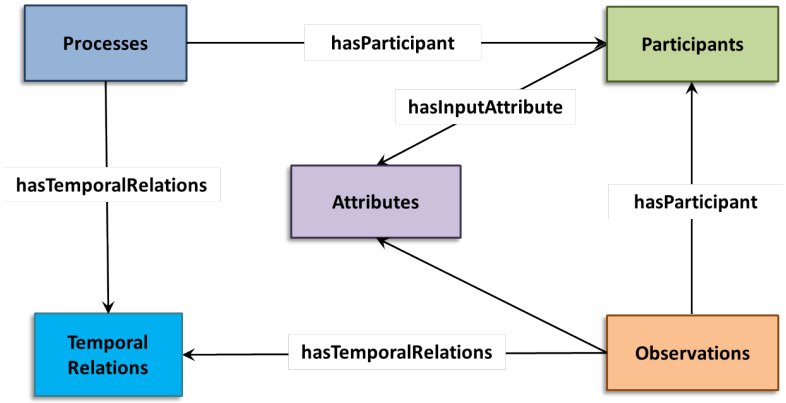


Figure 6.1: The five main ontology parts and their relations.

Spatial Data on the Web, provides a general-purpose specification for modelling the interaction between entities involved in the acts of observation, actuation, and sampling. It represents a lightweight replacement for the Semantic Sensor Network (SSN) ontology, which is harder to deploy due to its strong ontological commitments resulting from its alignment with the Dolce Ultra-Light ontology (DUL). With SOSA not recommending any particular way for modelling results, we have used external vocabularies specifically designed for modelling quantity values, and the 'Attributes' part of the ontology. For this we have used the Quantities, Units, Dimensions and Data Types (QUDT) ontologies [Hodgson et al., 2014] designed by NASA, with the goal of standardizing data structures and facilitate data integration and its interoperability[4]. Finally, for representing the ontology's temporal concepts, we have used the Time ontology in OWL (OWL-Time) [Cox and Little, 2017]. This ontology provides a vocabulary for expressing facts about topological relations among instants and intervals, together with information about durations, and temporal positions. Finally, and for the goal of increasing semantic interoperability, particularly in the life sciences domain, the model was fully integrated with the Basic Formal Ontology (BFO) [Arp et al., 2015]. BFO is a small, and genuine upper level ontology. It does not contain physical, chemical, biological or other terms which would properly fall within the coverage domains of the special sciences, and complexify its integration process. An important factor for adopting BFO in our model is its popularity[5] amongst life sciences domain. This would facilitate the interoperability of our model, and increase its visibility with respect to this domain's users.

Figure 6.2 presents an overview of the core concepts of the resulting model PO^2 (Process and Observation Ontology). The core ontology created in a top-down approach with BFO, is expressed in OWL, and is composed of 67 classes,

[4]Following NASA's metric confusion that caused the loss of a $125 million Mars orbiter.
[5]List of users: `http://basic-formal-ontology.org/users.html`

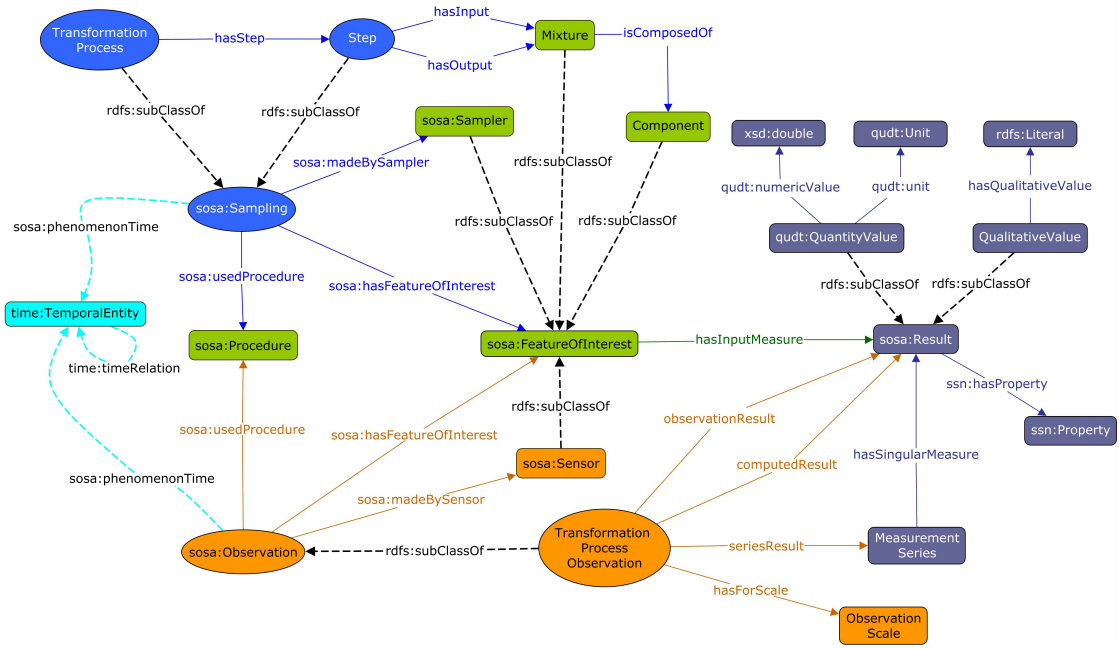


Figure 6.2: Core Concepts of the *PO*^{2} Ontology

61 object properties, and 12 data properties. Most of the core ontology classes belong to different namespaces, where concepts preceded by `sosa:`, `ssn:`, `qudt:`, and `time:`, respectively belong to the SOSA[6], SSN[7], QUDT[8], and OWL-TIME[9] namespaces. *PO*^{2} is published as part of the AgroPortal ontology library [Jonquet et al., 2018], and is available at `http://agroportal.lirmm.fr/ontologies/PO2`. This portal, based on the BioPortal technology, provides several state-of-the-art features [d'Aquin and Noy, 2012] that are dedicated for increasing the visibility and facilitating access to agronomic and life sciences data (e.g. ontology search, versioning, and visualization; semantic annotation; storage and exploitation of ontology alignments).

6.1.3 Knowledge Graph Construction

For creating a 'five star' knowledge graph, a conversion needs to take place from the data provided by the experts into RDF. Several mutual efforts have taken place for organizing the experts data to enable, with high precision, the automatic conversion into RDF. These efforts have mainly focused on restructuring large parts of the experts textual data into more concise tabular formats, using a

[6] `http://www.w3.org/ns/sosa/`
[7] `http://www.w3.org/ns/ssn/`
[8] `http://qudt.org/schema/qudt/`
[9] `http://www.w3.org/2006/time#`

common vocabulary. Respecting the experts wishes to continue collecting and archiving their data in Excel spreadsheets, we have created a set of guidelines for helping domain experts to provide us with machine-processable data, whilst still using Microsoft Excel as an archiving tool. These guidelines structure the expert data into several categories of Excel spreadsheets (e.g. files describing the process and its steps, files for describing the materials and methods, observation files). Figure 6.3 presents an excerpt of an Excel spreadsheet describing a certain observation conducted in the 'Cultivability' step, as part of the 'Fermentation' step. This spreadsheet describe the date and scale of the observation, and refers to other Excel files for describing the material and method used for this observation. This observation results in several raw measures (described in the ontology by the *observationResult* property), that are used for obtaining the computing measures (described by the *computedResult* property).

In order to manage and uniformize the vocabulary adopted by the experts, we have used and extended parts of the AgroVoc[10] multilingual thesaurus [Caracciolo et al., 2013]. This thesaurus is managed by the Food and Agriculture Organization of the United Nations (FAO), and serves as a controlled vocabulary for the indexing of publications in agricultural science and technology. Agrovoc is modelled in SKOS-XL[11], and contains over 35K concepts, described in over 20 languages (including French, in which the expert data are described).

Now that the Excel files are structured, and the vocabulary is uniformized, the last step consists of 'semantisizing' the experts' data. For this, we have developed a JAVA tool that processes the different categories of Excel files, and convert the experts data into RDF. In order to migrate from a semi-formal thesaurus-like structure to a formal ontology, we have transformed[12] the SKOS concepts, adapted in Agrovoc, to OWL classes. However, such mapping could be problematic, since a `skos:concept` might sometimes represent an instance, and the `skos:broader` relation can refer to an `rdf:type` relation instead of an `rdfs:subClassOf`. After manual verifications, such cases do not occur in the parts adopted from the Agrovoc thesaurus. An example of a SKOS concept we use is the leaf node of Agrovoc *glucose*[13]. In our model, *glucose* does indeed represent an OWL class, as a `rdfs:subClassOf po2:Component`, and instantiated for representing specific measures of glucose in a certain experiment.

The knowledge graph for life sciences resulting from the conversion of 2,845 Excel files contains 2,738,203 triples, divided into 21 named graphs. Each named graph represents a certain project in which several transformation processes were conducted. On average, a project describes 21 transformation processes

[10] `http://agrovoc.uniroma2.it/agrovoc/agrovoc/en/`

[11] `https://www.w3.org/TR/skos-reference/skos-xl.html`

[12] `https://www.w3.org/2006/07/SWD/SKOS/skos-and-owl/master.html#Transform`

[13] `http://aims.fao.org/aos/agrovoc/c_3287`

	A	B	C	D
1	**Informations générales**			
2	*Date de la mesure*	2016-02-23		
3	*Heure*			
4	*Durée de la manipulation*			
5	*Etape*	Fermentation		
6	*Sous étape*	Cultivabilité		
7	*Répétition*	1		
8	*Echelle*	Population		
9				
10	***Descriptif du produit***			
11	Code échantillon	2016-02-22-CellExtraDry-002		
12	Nom du fichier descriptif du cycle de vie	2016-02-22-CellExtraDry-002-FicheDescriptifCV		
13	Nature de l'échantillon	Humide		
14				
15	**Matériel utilisé**			
16	*Identifiant*	Matériel 16		
17	Fichier	2016-CellExtraDry-Matériel&Méthode		
18				
19	**Méthode utilisée**			
20	*Identifiant*	Méthode 2		
21	Fichier	2016-CellExtraDry-Matériel&Méthode		
22				
23	**Données calculées**			
24	*Caractéristiques*	*Moyenne*	*Écart-type*	*Unité*
25	*Concentration*	3.17E+08		UFC/mL
26	*Logarithme de la concentration*	3.17E+08		log(UFC/mL)
27	*Concentration par batch*	6.33E+12		UFC/batch
28	*Logarithme de la concentration par batch*	12.80		log(UFC/batch)
29				
30	**Données brutes**			
31	***Dilution***	***Dénombrement***	***Concentration***	***Concentration***
32	***mL***	***UFC***	***UFC/mL***	***log(UFC/mL)***
33		Nombre de colonies viables	UFC x Dilution	
34	1E-05	190	3.17E+08	8.50

Figure 6.3: Excerpt of an Excel spreadsheet describing an observation conducted in the 'Cultivability' step, as part of the 'Fermentation' step.

(total of 453 transformation process), with each process containing around 4 steps (total of 1830 steps), and each step containing two mixtures (one input, one output). In these projects, a total of 4315 observations were conducted at 6 different scales, measuring 623 different properties (e.g. temperature, pH). This knowledge graph can be queried and downloaded at `http://sonorus.agroparistech.fr:7200`.

6.2 Detection of Contextual Identity in Scientific Experiments

Now that the knowledge graph composed of hundreds of different experimental processes is created, the next goal is to semantically link these experiments. Since `owl:sameAs` can not be deployed for asserting such connections as the experimental conditions tend to vary, even slightly, between each experiment, and since alternative identity predicates have limited semantics (as discussed in section 2.4.2), we want to link these experiments using our proposed notion of contextual identity. For this, we have applied the `DECIDE` algorithm on the

resulting knowledge graph. At the time of conducting these experiments, only 11 out of the currently available 21 projects (i.e. named graphs) were created, with all these projects related to the release of flavour compounds during dairy gel consumption.

As presented in the ontology's five main parts (Figure 6.1), and implemented in the knowledge graph's conceptual model (Figure 6.2), a distinction is made between the actual experiments that include the processes (e.g. cheese productions) with their participants (e.g. products and devices), and between the observations conducted at the end of each step (e.g. observing the pH of the cheese). These observations contain a large number of missing information, since not every measure is consistently observed in each experiment's step. Therefore, we have discarded these observations by adding the properties that relate the experiments to the observations to the *Unwanted Properties (UP)* set of constraints (described in section 5.2.1).

6.2.1 DECIDE Results

Table 6.1 presents the results of `DECIDE` applied on 11 projects of this knowledge graph, and when the *Mixture* class and the *Step* class are considered separately as target classes. A mixture is a component which is composed of at least one other component (e.g. the processed cheese which is composed of 20g of salt). There are 1,187 instances of type *Mixture* in these projects, and 581 of type *Step*, forming respectively 703,891 and 168,490 pairs of instances to consider. The algorithm takes around 22 hours[14] for detecting the most specific global contexts, in which each of these pair of instances are identical.

Out of the 950 classes in the ontology, only the 784 most general instantiated classes, representing the descriptive classes, are used for determining the contexts (see section 5.1.3). On average, only one similarity graph per pair of instances is necessary for detecting contextual identity links. This is due to the few multivalued properties, that have values of the same *directType*, that can lead to the construction of several similarity graphs (as in the case of the property *hasDevice* linking the pr_3 and pr_4 processes to two Bioreactors each, as presented in Figure 5.1). A similarity graph is composed on average of 5.26 nodes for the *Mixture* class, and composed of 8.25 nodes for the *Step* class. These similarity graphs allowed to generate 1,279,376 identity links valid in 2,232 different global contexts for the pair of instances of the class *Mixture* and 348,017 identity links valid in 718 contexts for the instances of the class *Step*. These results show that two instances of these target classes, may be identical in more than one more specific global context (1.81 for $Mixture$ and 2.06 for $Step$). Finally,

[14] Executed on an 8GB RAM Windows 10 machine, with an Intel Core 4×2.6 GHz process.

Table 6.1: Results of *DECIDE* on the two target classes *Mixture* and *Step*

	Mixture	**Step**
# *Individuals of target class*	1,187	581
# *Possible Pairs*	703,891	168,490
# *Descriptive Classes (Total Classes)*	784 (950)	784 (950)
# *Similarity Graphs per Node*	1.004	1.085
# *Nodes per Similarity Graph*	5.26	8.25
# *Different Global Contexts*	2,232	718
# *Identity Links*	1,279,376	348,017
# *Identity Links per pair*	1.81	2.06

these detected global contexts, represented as named graphs, largely vary in their specificity, with a number of axioms varying between 2 (a general context containing one property, with a single domain and range) and 88 axioms.

6.2.2 Use of Experts Constraints

We have also studied how the addition of expert constraints could impact the results of the approach. For this, we have used a sample of the data, representing a single project, and containing 153 pairs of the *Mixture* target class. Without the expert constraints, `DECIDE` detects 502 identity links valid in 37 different global contexts (3.28 links per pair). A first expert constraint imposes that the value of an attribute (instance of the class *Attribute*) can not exist without its unit of measurement. By adding this co-occurrence constraint, `DECIDE` then discovers 377 identity links valid in 24 different global contexts (2.46 links per couple), leading to the removal of 125 irrelevant contextual identity links. The experts have also informed us that if the presence of water in the mixtures is considered, it is also necessary to consider the quantity of water in order for the context to be relevant. By adding this co-occurrence constraint cp_2 = $\{(Mixture, isComposedOf, Water), (Water, hasAttribute, Weight)\}$, the number of global contexts decreases from 37 to 35.

This evaluation indicates that the addition of constraints can significantly reduce the number of contexts and therefore the number of irrelevant contextual identity links. Of course, not all constraints have the same impact on the results. For instance, if the expert indicates that the property *isComposedOf*, connecting the Mixtures to its components is an irrelevant property, this would result in a total of 4 different global contexts, and 198 contextual identity links (1.29 links

per couple), as the removal of such property heavily reduces the size of the graphs describing the instances to be compared.

6.3 Contextual Identity Links for Rule Detection

The purpose of this experiment is to evaluate whether contextual identity links can be used to discover rules. More precisely, and since we have not considered the observations in the identity contexts, we seek to determine the probability of two experiments, being identical in a certain context, to have the same observation values. Eventually, it might then be possible to predict, with a certain degree of confidence, unobserved measures in an experiment.

According to Leibniz's "Indiscernibility of Identicals" principle [Forrest, 2008], a genuine identity between two objects (e.g. experiments), indicates that every property (e.g. an observed measure) asserted to one is asserted to the other: $x = y \wedge p_i(x, z) \rightarrow p_i(y, z)$ with $p_i \in P$. In this prediction task, we aim to detect for each context GC_i, the set Ψ of properties $\{p_1, ..., p_n\}$, where $identiConTo_{<GC_i>}(x, y) \wedge p_i(x, z_1) \rightarrow p_i(y, z_2)$ with $z_1 \simeq z_2$ and $\Psi \cap P^{GC_i} = \emptyset$. Such rules can be written as:

$$r = identiConTo_{<GC_i>}(x, y) \rightarrow same(m)$$

with m representing a certain observatory measure $\in \Psi$ (e.g. pH measure). Since the detected contextual identity links are only stated for the most specific contexts of each pair, we have exploited the global contexts' order relation (Definition 13) to obtain the complete set of contextual identity links for each global context. In order to evaluate the quality of a rule r, we calculate the following measures:

Error rate. For each pair of instances identical in GC_i, where a measure is observed for both instances, we calculate an error rate. The error rate er for a measure m between two instances x and y is calculated as follows:

$$er_m(x, y) = \frac{|m(x) - m(y)| \times 100}{|m(max) - m(min)|}$$

where $m(max)$ and $m(min)$ represent respectively the maximal and minimal value taken for the measure m in the dataset. The error rate of a rule for a global context GC_i is the average of the error rates for each pair of instances identical in this context.

Support. Representing the number of pair of instances identical in GC_i, and having the measure m.

Table 6.2: Evaluation of 20 rules by the experts

Impossible	Unlikely	Don't Know	Why Not	Very Plausible
3	5	4	5	3

Based on the output of the DECIDE algorithm for the class *Mixture*, we have generated 38,844 rules. The number of rules varies between one and 313 rules per context. On average, the support of a rule varies between 1 (e.g. only a single pair of instances having the measure *Bitter* in a certain context) and 15,075. The rules' error rate varies between 0 and 100%, with 1,005 rules having an error rate < 1%. On average, the error rate of a rule is around 35%.

In addition, we have tested if the rule's error rate varies depending on the specificity of the context. The experiments show that on average, the error rate of a rule decreases by 12 p.p[15] when a global context is replaced by a more specific global context. For instance given the following rules:

$$r_1 = identiConTo_{<GC_i>}(x, y) \rightarrow same(m_1)$$

$$r_2 = identiConTo_{<GC_j>}(x, y) \rightarrow same(m_1)$$

with $GC_j \leqslant GC_i$.

On average, r_2 has an error rate lower by 12 p.p than r_1. Indicating that the more a rule's context is specific, the more precise a rule is. Also indicating that contextual identity links can be exploited for predicting missing measures, with different confidence levels.

We have asked domain experts to evaluate the best 20 generated rules, chosen based on the error rate and support. More specifically, we have chosen the rules that can be easily understood by experts (i.e. with the fewest axioms) such that the error rate is less than 15% and has the highest support. The plausibility of the 20 rules given to the experts was evaluated using a scale of 5 appreciations: "impossible", "unlikely", "don't know", "why not", and "very plausible". Table 6.2, which presents the evaluation of the experts, shows that among these 20 rules, 3 are very plausible. These 3 rules are presented in the table 6.3, representing the rules in which the experts are aware of the impact of the properties considered in the contexts on the value of the observed measure. For example, the expert found that it is very plausible that two mixtures with the same citric acid weight, would have the same observed pH value (first rule). These "very plausible" detected rules, represent implicit experts knowledge, which we can use to complete unobserved measures and consequently the knowledge graph. In addition to these known rules, we were able to provide experts with 14 rules that could be the subject of further studies. These are the

[15]percentage point

Table 6.3: Error rate and support of the most plausible rules

Rule	*Error Rate*	*Support*
$identiConTo_{<GC_1>}(x, y) \rightarrow$ same(pH)	6.19 %	57
$identiConTo_{<GC_3>}(x, y) \rightarrow$ same(Hardness)	1.86 %	66
$identiConTo_{<GC_2>}(x, y) \rightarrow$ same(Friability)	4.52 %	647

rules that have been evaluated as "plausible", "don't know", and "unlikely". For example, the expert considered that it is possible that when two mixtures have the same amount of water, they will also share the same observed viscosity measure (rule considered as plausible). On the other hand, three of the provided rules seem impossible to the experts, based on their knowledge that there is no dependence between the properties considered in the identity context and the measures observed.

We have also exploited the contextual identity links and the generated rules to answer competency questions provided by experts. For instance, experts are interested whether there is a dependency between having the same *amount of lipid* in the mixtures and the observed *rheology notes*, corresponding to three types of measures (MD, Wf, and σf). For answering this question we have selected, using a SPARQL query, all the global contexts (i.e. named graphs) containing at least the following axioms:

domain(isComposedOf) = Mixture , range(isComposedOf) = Lipid
domain(hasWeight) = Lipid , range(hasWeight) = Weight ,
domain(hasValue) = Weight , range(hasValue) = xsd:float ,
domain(hasUnit) = Weight , range(hasUnit) = xsd:string

Since there is no global context which contains solely these axioms, we have selected the least specific ones resulting from this SPARQL query. This way, we can reduce the effect of the additional axioms, also included in this context, might have on the *rheology notes*. From the remaining five least specific global contexts GC_{res} that contain these axioms, we have provided experts with the average of all rules of the following type, for each measure (MD, Wf, and σf):

$$r = identiConTo_{<GC_i>}(x, y) \rightarrow same(m)$$

with $GC_i \in GC_{res}$, and m representing either MD, Wf, or σf .

The average error rate for the measure 'σf' is 5.2%, while the average error rate for 'MD' and 'Wf' is 13.8% and 11.5% respectively. This experiment suggests that there exist indeed a high dependency between having the same *amount of lipid* in the mixtures and the observed *rheology notes*, especially for the σf measure.

6.4 Results Summary

Our collaboration with the domain experts, and the here presented experiments conducted on this knowledge graph describing scientific experiments have shown that:

- The use of genuine identity links such as the *owl:sameAs* link is rarely required in scientific datasets, since the experiments' environment tend to change, even slightly from one experiment to another, resulting in a propagation of incorrect observational measures.
- Asking domain experts to specify the contexts in which two instances are considered identical is not an intuitive task, since the identity contexts are task dependent and differ between each expert. Instead, specifying some constraints on these contexts in a form of necessary, unwanted, and co-occurring properties is a more effective way to benefit from the experts knowledge.
- Contextual identity links, detected for each pair of instances of a target class, allow to store the similarities of these instances and facilitate their querying.
- Contextual identity links can be used for generating rules that can help predict some of the missing observation measures. Since generated rules in more specific contexts have better error rates than rules detected in less specific ones, the specificity of a context can serve as a confidence indicator of the rule.
- The relevance of a certain context can vary depending on the conducted observations. For instance, the identity of the mixtures' composition is required in tasks that study the mixtures' acidity, while the identity of the steps in which the mixtures appear, is required in tasks studying the experiments' environmental impact.

6.5 Conclusion

In this chapter, we introduced a new knowledge graph describing two specific domains: the mechanisms leading to the release of flavour compounds during dairy gel consumption, and the stabilisation of micro-organisms. This graph is based on scientific experiments conducted and collected from the BioMiP and FFOPP teams of the French National Institute of Agricultural Research (INRA). This continuously growing knowledge graph provide experts with homogenized data, both in terms of its published format and in terms of the used terminologies, allowing the expert data and knowledge to be easily interrogated and

consumed. In addition, by favouring the reuse of concepts in the graphs' core model, and the deployed vocabulary, this data is also semantically interoperable and can be consumed by a large number of knowledge-based applications.

This knowledge graph provides experts with various explicit and implicit semantic connections between the experiments' participants. In order to provide experts with various semantic connections between the different conducted experiments, we have applied our approach for detecting contextual identity links. By applying `DECIDE` separately on the experiments' main classes *Mixture* and *Step*, we have detected more than 1.5M contextual identity link between the different experiments. These links were later deployed for discovering certain types of rules that have exploited the global contexts' order relation. With rules in more specific contexts having better error rates than rules detected in less specific ones, the specificity of a context can serve as a confidence indicator.

CHAPTER 7
CONCLUSION & PERSPECTIVES

This chapter discusses the results of the research presented in this thesis, as well as its limitations, lessons learned during the process of conducting it, and some lines for future work.

7.1 Summary of Results

This thesis have investigated one specific research question: *how to limit the excessive and incorrect use of identity links in knowledge graphs.* In order to address this identity problem, we have proposed different, yet complementary solutions. In the following, we highlight the main results of this thesis.

In Chapter 2, we have investigated existing approaches that have contributed to this research question by studying the use of identity in the Web of Data, and proposing possible solutions. This survey has focused on four categories of approaches: (i) studies that have analysed the use of identity links in the Web of Data; (ii) solutions that help users or applications to identify IRIs referring to the same real world entity, and distinguish similar labels referring to different real world entities; (iii) approaches that aimed at detecting erroneous identity links and/or validate correct ones; and finally (iv) approaches that proposed alternative identity relations as a way for limiting the incorrect use of `owl:sameAs`. This survey shows the following:

Existing identity analyses are not representative enough. All identity analyses were conducted on a relatively small number of identity links, compared to the size of the Web of Data. This drawback shows the need of having identity management services that can help harvest, filter, and store large collections of identity links, and consequently enable discovering important aspects of the identity use in the Web of Data.

Existing identity management services have many limitations. In their current status and architecture, existing identity management services are not able to provide reliable solutions in terms of semantic interpretability, terms coverage, and up-to-date support. The current situation shows that easily finding, understanding, and reusing identical terms is still a difficult task for users and applications. Hence the risk of misusing, and erroneously linking terms in the Web of Data is still present.

Existing identity link invalidation approaches are not feasible in the Web. Approaches that can be efficiently applied on the whole Web of Data has yet to emerge. Existing approaches are either not developed to be applied

to a large number of links, or require assumptions on the data that are not valid in the context of the Web.

Alternative identity links lack semantics. Existing alternatives consist of either simple predicates that do not explicitly state the contexts in which two terms are identical, or approaches that expresses the identity relation by relying solely on the local properties. The current situation shows that the lack of well-defined alternatives risk maintaining this excessive and incorrect use of `owl:sameAs`.

In Chapter 3, we have showed that the presence of an identity observatory service that collects and hosts a large set of identity statements can help uncover different aspects of identity. The here presented `sameas.cc` dataset and Web service provides easy access and download to the largest collection of `owl:sameAs` statements collected to date, and the resulting identity sets. In addition, we have presented an efficient approach for extracting and storing the identity statements, and calculating their transitive closure. These resources have enabled us to conduct several analyses over the identity use in the Web of Data, including the number of explicit and implicit `owl:sameAs` statements, its kernel, and analyses on the aggregated level of datasets. The analyses we presented in this chapter is an order of magnitude larger than previous conducted identity analyses. In addition of enabling large-scale identity analyses, the here presented resources can help users and applications in finding and reusing identical terms, and consequently enabling many identity-based services, such as question answering and ontology alignment services.

In Chapter 4, we have showed that ranking each identity link in the Web of Data is feasible in practice. We have presented an approach that relies on the community structure of the `owl:sameAs` network, and their symmetrical characteristic, for assigning an error degree fo each `owl:sameAs` link. This approach does not require any assumptions on the data, and have been applied on the whole `sameas.cc` dataset, containing over 558M `owl:sameAs` statements. With an accuracy of 86%, the manual evaluation of around 1000 `owl:sameAs` shows that the here introduced error degree can indeed be used for distinguishing correct `owl:sameAs` from erroneous ones. In addition, the evaluation shows that a symmetrical identity link has more chances of correctness than a non-symmetrical one, hence suggesting that a mutual agreement on linksets can have a measurable impact on the quality of identity assertions.

In Chapter 5, we have showed that the classical identity relation standardized in OWL is problematic, and there is a need for new context-dependent identity relations. We have introduced a new identity relation that expresses identity between two class instances, that holds in a context defined with regard to a domain ontology. We have proposed an approach for automatically detecting, and representing the most specific contexts in which two instances

are identical. This approach, can consider certain expert constraints that should be respected by all detected contexts, and given in the form of necessary, unwanted, and co-occurring properties.

In Chapter 6, we have showed that the proposed contextual identity relation is applicable and beneficial in scientific knowledge graphs, where the classical notion of identity can not be applied. We have constructed a knowledge graph for life sciences composed of several distinct projects, from two different domains: the mechanisms leading to the release of flavour compounds during dairy gel consumption and their impact on global sensory perception, and the process of stabilisation of micro-organisms. We have showed that despite the rather large number of highly connected classes of the here constructed graph, thousands of contextual identity links can be detected for semantically linking the experiments' participants. The experiments show that the use of expert constraints can have a massive impact in reducing the runtime and the number of irrelevant identity contexts. In addition, we have exploited these contextual identity links to generate thousands of rules, which were calculated using the global contexts' order relation. The experiments show that the contexts' specificity can serve as a rule's confidence indicator, with rules in more specific contexts having better error rate in average than rules detected in less specific contexts. After the experts validation, these rules can be used to predict, with a certain degree of confidence, unobserved measures in a scientific experiment and consequently complete the knowledge graph with implicit assertions.

7.2 Discussion and Future Work

In this final section, we outline various avenues for future work, motivated by certain limitations in the contributions presented in this thesis.

A. Identity Management Service

Identity management services represent an important aspect in solving the presented identity problem, as they can facilitate the re-use of IRIs, and enable large scale identity analyses. As an essential way for maintaining and improving our identity management, several directions can be implemented and investigated.

Links' Provenance Inclusion. Despite relying on a collection of freely accessible datasets from the LOD Laundromat, tracking the provenance of each identity statement is still a difficult task for the user, as it requires searching in the LOD Laundromat (Wardrobe), for identifying the dataset(s) responsible for each identity assertion. In the next update of this service, we

plan to provide the provenance of each explicit identity statement. This will help users to discard unwanted or untrusted sources when using the `sameas.cc` dataset, and enable analyses at the level of the links' datasets, not only according to the IRIs' namespaces as presented in this thesis.

Identity Observation over Time. Since the 2015's LOD Laundromat crawl in which our dataset is based on, a large number of identity statements might have been deprecated or added by now. And due to the identity's transitive trait, even few changes in the explicit identity network can massively reshape the resulted identity sets, and change the here presented analyses. As a way to observe changes on how identity is used in the Web of Data, we will update `sameas.cc` as soon as a new crawl of the LOD Laundromat is performed.

B. Detection of Erroneous Identity Links

Detecting existing erroneous identity links represents a necessary aspect for controlling the quality of the Web, and dealing with the identity problem at hand. Having an efficient approach that can be applied on the whole Web of Data is an important research direction, that was investigated in this thesis. However, in order to improve several aspects of the here presented approach, several directions can be implemented and investigated. In the following, we outline these possible directions, starting from short-term works to longer-term ones.

Additional Evaluation. An important limitation of the here presented experiments is the number of manually evaluated links in which we base our results on, compared to the number of links in the Web of Data. In the short term, we will look into the use of crowdsourcing for evaluating a larger number of `owl:sameAs` links. In fact, the experiments conducted by [Acosta et al., 2013] shows that using a majority voting strategy, paid microtask workers can evaluate interlinks with an accuracy as high as 94%. This will allow us to have more representative precision and recall evaluation, and more importantly allow us to understand the conditions in which our approach can be applied.

Inclusion of Duplicate Identity Links. We have tested our approach on the *LOD-a-lot* dataset which discards millions of duplicate statements from the LOD Laundromat 2015 crawl. Since our approach is based on the topology of the network, and the number of `owl:sameAs` assertions between its terms, we can also consider including duplicate `owl:sameAs` assertions in our data graph. This indicates that an `owl:sameAs` statement between two terms can have a weight much higher than two, when the

same statement is declared by different datasets. For this, we will investigate how these duplicate identity links can be included in the error degree, and study whether the redundancy of `owl:sameAs` links have a similar impact on its quality, as demonstrated for symmetry.

Equality Set's Size Impact. Our experiments suggest that the precision of our approach is highly dependent on the number of terms in an equality set (precision dropping from 88% in the largest equality set to 40% in random ones, for a threshold of 0.99). As a way of reducing the number of false positives in our approach, we will investigate the impact of including the equality sets' size in the link's error degree. More specifically, we will study which aspect of the equality set's size (number of terms, number of links, or number of communities) has the most impact on the precision of our approach, and how it can be included in the error degree.

Combining Community Detection Techniques. An important limitation of the here presented experiments, is its dependency on a single community detection technique. With the *Louvain* algorithm relying on modularity optimization for detecting densely connected nodes, we can consider other state-of-the-art methods such as the statistical inference-based method by [Rosvall and Bergstrom, 2008] and the multiresolution method by [Ronhovde and Nussinov, 2009] which have also proven their efficiency in terms of accuracy and scalability according to [Lancichinetti and Fortunato, 2009b]'s analysis. As a first step, we can conduct the same experiments for each of these other techniques, and compare their resulting community structure using precision, recall, and accuracy. As a longer-term direction, we will investigate combining the results from these different techniques. For combining the several techniques, different strategies could be considered. Firstly, despite its use of modularity as an objective function for detecting the community structure, *Louvain* does not guarantee achieving a maximum modularity. Hence, as a first direction we can consider applying the different community detection methods separately on each equality set, and choose the community structure with the highest modularity measure. Another direction considers applying these techniques on each equality set, but also calculating the links' error degrees separately for each technique. Then several directions can be considered for combining the resulting error degrees such as voting, or defining aggregation functions. Finally, we can consider choosing a different community detection technique for each equality set. In this strategy, we can investigate for each type of network structure, the community detection technique that can be applied more efficiently. This will allow us to combine the different techniques, whilst consuming minimal resources.

Combining with state-of-the-art Approaches. A significant limitation of the here presented approach is its inability in detecting erroneous `owl:sameAs` links belonging to equality sets of cardinality 2. In fact, such

links can only have two possible error degrees: 0.5 for non-symmetrical statements, and 0 for symmetrical ones. This limitation impacts around 55M `owl:sameAs` statements that belong to equality sets with a cardinality of 2 (around 10% of all `owl:sameAs` statements). Hence, as a longer term direction, we will investigate combining our approach with other types of approaches. A first strategy can consider using other techniques for detecting erroneous links in smaller equality sets. For instance, when the terms' textual description is available, we can consider comparing the similarity of the terms' textual descriptions. This type of approach has proven its efficiency by [Cuzzola et al., 2015], reporting high precision when the terms' textual description is available. When it is not the case, we can consider applying consistency checking techniques such as [Papaleo et al., 2014, Hogan et al., 2012]. In addition, since our approach can be applied on the whole data with no requirements, and suggests higher recall than precision, another strategy can be defined for improving the precision of our approach. This strategy can consider applying our approach first on the whole dataset, and then deploy other types of approaches on links with high error degree.

C. Contextual Identity Relation

Having different weaker types of identity can massively limit the excessive and incorrect use of `owl:sameAs`. Representing the contexts in which identity holds is a necessary aspect for limiting the `owl:sameAs` use, as it formally informs users about the contexts in which these two instances can be used interchangeably. This direction of defining and detecting the identity contexts has been investigated in this thesis, and can be extended in several ways.

Identity of Literals. Since literals appear in one out of three Semantic Web statements in the Web of Data [Ilievski et al., 2015], a future direction can consider a more adapted definition for measuring identity of literals. Instead of the lexical expression equality currently adopted in our approach, we can investigate whether identity between different Semantic Web datatypes should be authorized, and whether a more lenient approach for the identity of literals can be considered. For instance, the two lexical expressions 0.1 and 0.10000000009 map to the same value according to datatype `xsd:float` (32 bit), but map to different values according to datatypes `xsd:double` (64 bit) and `xsd:decimal` (128 bit), where the digits of precision is different [Beek, 2018]. With the adoption of a more relaxed identity of literals, a study on its impact in inference is required.

Adaptation Strategies. The requirement of having the same conceptual model represents a significant limitation of our proposed identity relation. This

requirement limits the use of this identity relation in the context of the Web, and restricts its use to specific knowledge graphs. In addition, since computing the identity relation for each pair of instances could result in the propagation in the whole knowledge graph, the applicability of the here proposed algorithm is limited to smaller knowledge graphs. A future direction can investigate several strategies, for adapting our identity relation to certain ontology mappings and relaxing the algorithm's constraints (e.g. limiting the graph search to a lower depth). These more relaxed measures would allow our approach to complement the detection of erroneous links, and replace the incorrect `owl:sameAs` in the Web with a more adapted contextual identity relation.

Contexts of Difference. In addition of detecting and representing contexts in which two instances are identical, we can also explore defining contexts in which two instances are explicitly different. Such contexts can be useful for experts, as it informs them in which applications two class instances can not be used interchangeably. This notion of difference can not be deducted from the identity contexts, as they do not distinguish between the absence of a property, for one or both instances, and the difference of the property values.

Knowledge Discovery. By combining the detected contextual identity links, with the contexts where instances are explicitly different, we can exploit our approach in other tasks. In particular, we aim at discovering causal rules, in which the contextual identity links and the contexts of difference can allow us to compare experiments, and use the instances temporal aspects, for identifying the causes of variations in the observation measures.

As a longer term direction, we will investigate the possibility of implementing certain changes of practice, in terms of how identity is asserted in the LOD Cloud. This practice encourages Linked Data publishers to validate the 'correctness' of an `owl:sameAs` statement, with respect to its corresponding identity set, and prior to its assertion. Such notion of correctness can be defined and parametrized according to several hypothesises, such as logical consistency [CudreMauroux et al., 2009, Hogan et al., 2012, Papaleo et al., 2014], UNA validation [de Melo, 2013, Valdestilhas et al., 2017], terms' descriptions similarity [Paulheim, 2014, Cuzzola et al., 2015], and/or the identity statement's impact on the network structure [Guéret et al., 2012, Sarasua et al., 2017, Raad et al., 2018b]. By providing users and applications the possibility of validating an identity statement's correctness according to different hypothesises, such practice can limit the "sameAs problem" from the source. For implementing such tool, several necessary directions can be investigated, such as large scale inconsistency detection, and ontology mappings. We note that this direction does not intend to force users to go through an authority in

order to link their data, but intends to serve as a way of labelling incorrect identity assertions or re-qualifying these links into a more parametrized identity relation. The goal is by preventing the publication of incorrect `owl:sameAs`, and detecting the incorrect existing ones, we envision to construct a parallel and a higher quality subset(s) of the LOD Cloud.

APPENDIX A
RÉSUMÉ EN FRANÇAIS

A.1 Introduction

Le Linked Open Data est une initiative du W3C[1], qui définit un ensemble de bonnes pratiques pour publier et lier des données structurées sur le web. En utilisant des technologies du web sémantique, des applications peuvent partager, extraire, interroger ou raisonner sur les données publiées. Le web des données référencé par le terme LOD (Linked Open Data) a récemment pris une nouvelle dimension avec la publication de grandes quantités de données (le LOD est passé de 500 millions de triplets RDF[2] en 2007 à plus de 140 milliards de triplets en 2018). Ces données, publiées sous forme de graphes de connaissances RDF, sont encyclopédiques comme celles de DBpedia, Yago et Wikidata ou bien concernent différents domaines d'application comme les sciences du vivant, la culture ou encore l'économie.

En l'absence d'une autorité de nommage centrale sur le web des données, il est fréquent que ces différents graphes de connaissances utilisent des noms différents pour référer à la même entité du monde réel. Par exemple, pour référer à l'ancien président des États-Unis Barack Obama, il existe plus que 440 noms (IRI[3]) dans le web des données, utilisés par différentes sources (e.g. 'dbr:44th_US_president', 'yago:Barack_Obama', 'wd:Q76'). Quand plusieurs noms sont utilisés pour désigner la même entité, des assertions `owl:sameAs` sont nécessaires pour accéder à l'ensemble des informations qui décrivent l'entité. De telles déclarations d'identité ont une sémantique logique très stricte, indiquant que chaque propriété associée à un nom sera également déduite pour l'autre, et vice versa. Bien que ces inférences puissent être extrêmement utiles pour enrichir les systèmes fondés sur les connaissances tels que les moteurs de recherche et les systèmes de recommandation, l'utilisation incorrecte de l'identité peut conduire à des effets négatifs importants dans un espace de connaissances global comme le web des données (i.e. inconsistance, inférences d'assertions erronées). Or, différentes études existantes ont montré que le constructeur `owl:sameAs` est souvent utilisé incorrectement sur le web des données, et ceci pour plusieurs raisons. Premièrement, la plupart de ces assertions `owl:sameAs` sont générées par des méthodes automatiques de liage utilisant des stratégies dont la précision n'est pas garantie. Par exemple, un algorithme liant des livres en fonction de la similarité de leurs titres et de leurs auteurs n'est pas toujours précis, car deux éditions différentes du même livre peuvent également partager ces deux traits sans être identiques. De plus, l'identité

[1] World Wide Web Consortium
[2] Resource Description Framework
[3] Internationalized Resource Identifier

n'est pas valide dans tous les contextes, puisque deux choses peuvent être considérées comme identiques pour certains utilisateurs dans certains contextes, alors qu'elles seront considérées comme différentes pour d'autres personnes ou dans d'autres contextes. Par exemple, deux médicaments partageant la même structure chimique, mais fabriqués par différentes sociétés, peuvent être considérés comme identiques dans un contexte scientifique, mais comme différents dans un contexte commercial.

Comme il n'existe pas d'alternative au constructeur `owl:sameAs` dont la sémantique soit bien définie, celles-ci sont rarement utilisées en pratique, et chaque application du LOD est obligée de prendre une décision en fonction de chaque assertion `owl:sameAs` qu'elle rencontre. Ce problème d'utilisation incorrecte de l'identité n'est pas spécifique au web sémantique et est présent dans tous les systèmes de représentation des connaissances [Grant and Subrahmanian, 1995, Nguyen, 2007]. Cependant, le problème est particulièrement important sur le web des données en raison de sa taille, de l'hétérogénéité de son contenu, et de l'absence d'une autorité de nommage centrale. Actuellement, ce problème de l'utilisation de l'identité dans le web sémantique est largement reconnu et a été qualifié de "crise d'identité" [Bouquet et al., 2007] ou de "problème du sameAs" [Halpin et al., 2010]. Aussi, une approche appropriée pour gérer ces liens d'identité est nécessaire pour que le web des données soit un succès en tant qu'espace de connaissances intégré.

La gestion de l'identité dans les graphes de connaissances est l'objectif principal de cette thèse. Plus précisément, cette thèse s'intéresse à l'un des aspects particulier de ce problème d'identité qu'est l'utilisation incorrecte des liens d'identité dans les graphes de connaissance. Cette thèse n'adresse pas certains des sujets de recherche connexes, tels que le liage de données et l'alignement d'ontologies pour la détection de liens `owl:sameAs` [Ferrara et al., 2013, Nentwig et al., 2017]. En outre, cette thèse ne traite pas de la distinction entre la localisation d'un document électronique avec une URL et la désignation d'une ressource RDF avec un IRI, connu sous le nom de problème de "Sens et Référence" [Halpin, 2010]. Afin de limiter ce problème de liens d'identité erronés ou inappropriés dans les graphes de connaissances, cette thèse propose différentes solutions complémentaires qui permettent d'observer et d'utiliser les liens d'identités existants, de détecter les liens erronés, et de représenter les liens d'identité qui ne sont valides que dans un contexte sémantique donné. Dans la suite, nous présentons brièvement ces différentes contributions.

A.2 Etat de l'art

Dans le chapitre 2, nous avons examiné les approches existantes qui ont contribué au problème de l'utilisation incorrect des liens d'identité dans les graphes de connaissances. Cette étude s'est concentrée sur quatre catégories d'approches: (i) les études ayant analysé l'utilisation de liens d'identité dans le Web des données; (ii) les services permettant aux utilisateurs ou aux applications de rechercher les adresses IRI faisant référence à la même entité du monde réel et de distinguer des noms similaires faisant référence à différentes entités du monde réel; (iii) les approches visant à détecter des liens d'identité erronés ou à valider ceux qui sont corrects; et enfin (iv) les approches proposant des relations d'identité alternatives comme moyen de limiter l'utilisation incorrecte de owl:sameAs. Cette étude montre ce qui suit :

Les approches d'analyse des liens d'identité existantes ne sont pas assez représentatives. Toutes les analyses d'utilisation des liens d'identité ont été effectuées sur un nombre relativement petit de liens d'identité, par rapport à la taille du Web de données. Cette limitation montre la nécessité de disposer de services de gestion d'identité pouvant contribuer à la collecte, au filtrage et au stockage de vastes collections de liens d'identité, ce qui permettrait de découvrir des caractéristiques importantes de l'utilisation de l'identité dans le Web des données.

Les services de gestion d'identité existants comportent de nombreuses limitations. Les services de gestion d'identité existants ne sont pas en mesure de fournir des solutions fiables en termes d'interprétabilité sémantique, de couverture des termes et de prise en charge à jour. La situation actuelle montre qu'il est toujours difficile pour les utilisateurs et les applications de trouver, de comprendre et de réutiliser des termes identiques. Par conséquent, le risque de mauvaise utilisation des termes dans le Web de données est toujours présent.

Les approches existantes d'invalidation de lien d'identité ne sont pas applicables sur le Web des données. Des approches pouvant être efficacement appliquées sur l'ensemble du Web de données n'ont pas encore émergé. Les approches existantes ne sont pas développées pour être appliquées à un grand nombre de liens, ou nécessitent des hypothèses sur les données (i.e. données homogènes, existences de descriptions textuelles ou d'axiomes décrivant la sémantique des propriétés) qui ne sont pas valides dans le contexte du LOD .

Les liens d'identité alternatifs manquent de sémantique. Les alternatives existantes sont soit des prédicats subjectifs qui n'énoncent pas explicitement les contextes dans lesquels deux termes sont identiques, soit des approches qui représentent le contexte dans lequel la relation d'identité est

valide mais en s'appuyant uniquement sur les propriétés décrivant l'entité localement (chemins de longueur 1 dans le graphe de données). La situation actuelle montre que l'absence d'alternatives bien définies risque de renforcer l'utilisation excessive et incorrecte du `owl:sameAs`.

A.3 Service de gestion et d'analyse d'identité

Dans le chapitre 3, publié dans [Beek et al., 2018], nous avons montré que la présence d'un service de gestion d'identité qui collecte et héberge un grand nombre de déclarations d'identité peut aider à découvrir différents aspects de l'usage d'identité dans le LOD. Le jeu de données et le service Web `sameas.cc` présentés dans ce chapitre, permettent un accès facile à la plus grande collection de liens `owl:sameAs` collectées à ce jour et aux classes d'équivalences résultantes. De plus, nous avons présenté une approche efficace pour extraire et stocker les liens d'identité et calculer leur clôture transitive. Ces ressources nous ont permis d'effectuer plusieurs analyses sur l'utilisation de l'identité dans le LOD, notamment le nombre de déclarations explicites et implicites de `owl:sameAs`, et des analyses au niveau agrégé des jeux de données. Les analyses que nous avons présentées dans ce chapitre sont d'un ordre de magnitude supérieur à celles présentés dans le chapitre 2. En plus de faciliter l'analyse des liens d'identité à grande échelle, les ressources présentées ici peuvent aider les utilisateurs et les applications à trouver et à réutiliser des termes identiques, et par conséquent aider de nombreux services fondés sur l'identité, tels que les services de recherche d'information ou les approches d'alignement d'ontologies. Ces liens `owl:sameAs`, avec leurs classes d'équivalences résultantes après clôture transitive, sont accessibles à travers notre service de gestion d'identité : `http://sameas.cc`

A.4 Méthode de détection des liens d'identité erronés

Dans le chapitre 4, publié dans [Raad et al., 2018a, Raad et al., 2018b], nous avons proposé une approche permettant d'attribuer un degré d'erreur pour chaque lien `owl:sameAs`. Notre méthode se base sur la densité de la ou des communautés auxquelles les entités impliquées par le lien appartiennent et sur l'existence d'un lien symétrique. Cette méthode se base sur l'algorithme de détection de communauté de Louvain. L'un des avantages de cette approche est qu'elle ne repose que sur le graphe formé par les liens d'identité, et qu'elle ne nécessite aucune hypothèse sur les données. De plus, nous avons montré qu'une telle approche peut être appliquée à l'ensemble du jeu de données `sameas.cc`, ensemble contenant plus de 558 millions liens `owl:sameAs`. Nous

avons manuellement évalué 1000 `owl:sameAs` et cette évaluation montre que le degré d'erreur que nous avons défini peut être effectivement utilisé pour distinguer les liens d'identité corrects des liens erronés (précision de 86%). De plus, l'évaluation montre qu'un lien d'identité symétrique a plus de chances d'être correct qu'un lien non-symétrique, suggérant ainsi que la prise en compte de l'existence d'un accord mutuel sur un lien peut améliorer l'évaluation de sa qualité. L'outil de détection des liens d'identité erronés est disponible sur le lien suivant : `https://github.com/raadjoe/LOD-Community-Detection`.

A.5 Relation d'identité contextuelle

Dans le chapitre 5, basé sur [Raad et al., 2017a, Raad et al., 2017b], nous avons montré que la relation d'identité classique définie dans OWL est problématique, et qu'il est nécessaire de créer de nouvelles relations d'identité dépendantes du contexte. Nous avons défini une nouvelle relation d'identité qui exprime une identité entre deux instances d'une classe, qui est valide dans un contexte défini par rapport à une ontologie de domaine (sous-ensemble de classes, de propriétés et d'axiomes). Nous avons proposé une approche permettant de calculer les contextes sémantiques les plus spécifiques dans lesquels deux instances sont identiques. Cette approche peut prendre en compte certaines contraintes expertes qui doivent être respectées dans tous les contextes détectés et saisis sous la forme de propriétés nécessaires, indésirables et co-occurrentes. L'outil de détection automatique des liens d'identité contextuelle est disponible sur le lien suivant : `https://github.com/raadjoe/DECIDE_v2`.

A.6 Graphes de connaissance pour les sciences de la vie

Dans le chapitre 6, basé sur [Ibanescu et al., 2016, Raad et al., 2018c], nous avons montré que notre approche de détection de relations d'identité contextuelle est applicable et pertinente dans les graphes de connaissances scientifiques, où la notion classique d'identité peut rarement être appliquée. Nous avons construit un graphe de connaissances pour les sciences de la vie pour des données issues de deux projets INRA, représentant des processus de transformation dans deux domaines d'applications : stabilisation des micro-organismes et l'interaction des macromolécules de l'aliment et de la salive dans des systèmes complexes. Afin de sémantiquement lier les centaines d'expérimentations réalisées, nous avons construit un graphe de connaissances basé sur l'ontologie PO^2 (Process and Observation Ontology), sur lequel nous avons appliqué notre approche de détection automatique des liens d'identité contextuelle. Nous avons montré que, malgré le nombre assez élevé de classes fortement connectées du

graphe, des milliers de liens d'identité contextuels peuvent être détectés qui permettent de lier sémantiquement les éléments participants à ces expériences. L'évaluation montre que l'utilisation de contraintes expertes peut avoir un impact considérable sur la réduction du temps d'exécution et sur le filtrage des contextes d'identité non pertinents. De plus, nous avons exploité ces liens d'identité contextuelle pour générer des milliers de règles qui permettent de déduire des observations à partir des données expérimentales. Ces règles sont calculées en s'appuyant sur la relation d'ordre entre contextes. L'évaluation montre que la spécificité des contextes peut servir d'indicateur de confiance d'une règle. En effet, les règles déduites dans des contextes plus spécifiques ont en moyenne, un taux d'erreur plus faible que les règles détectées dans des contextes moins spécifiques. Une fois validées par les experts, ces règles peuvent être utilisées pour compléter, avec un certain degré de confiance, les mesures manquantes d'une expérience scientifique et par conséquent compléter le graphe de connaissances avec de nouvelles assertions. L'ontologie PO^2 et le graphe de connaissances introduits dans ce chapitre sont respectivement accessibles au liens suivants : `http://agroportal.lirmm.fr/ontologies/PO2` et `http://sonorus.agroparistech.fr:7200/`.

BIBLIOGRAPHY

[Acosta et al., 2013] Acosta, M., Zaveri, A., Simperl, E., Kontokostas, D., Auer, S., and Lehmann, J. (2013). Crowdsourcing linked data quality assessment. In *International Semantic Web Conference*, pages 260–276. Springer.

[Arenas et al., 2008] Arenas, A., Fernandez, A., and Gomez, S. (2008). Analysis of the structure of complex networks at different resolution levels. *New Journal of Physics*, 10(5):053039.

[Arp et al., 2015] Arp, R., Smith, B., and Spear, A. D. (2015). *Building ontologies with basic formal ontology*. Mit Press.

[Bakker, 1987] Bakker, R. R. (1987). Knowledge graphs: representation and structuring of scientific knowledge (doctoral dissertation). University Twente.

[Beek, 2018] Beek, W. (2018). The 'k' in 'semantic web' stands for 'knowledge' (doctoral dissertation). Retrieved from the Digital Academic REpository of VU University Amsterdam.

[Beek et al., 2018] Beek, W., Raad, J., Wielemaker, J., and van Harmelen, F. (2018). sameas. cc: The closure of 500m owl: sameas statements. In *Extended Semantic Web Conference*, pages 65–80. Springer.

[Beek et al., 2014] Beek, W., Rietveld, L., Bazoobandi, H. R., Wielemaker, J., and Schlobach, S. (2014). Lod laundromat: a uniform way of publishing other people's dirty data. In *International Semantic Web Conference*, pages 213–228. Springer.

[Beek et al., 2016] Beek, W., Schlobach, S., and van Harmelen, F. (2016). A contextualised semantics for owl: sameas. In *International Semantic Web Conference*, pages 405–419. Springer.

[Blondel et al., 2008] Blondel, V. D., Guillaume, J.-L., Lambiotte, R., and Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008.

[Boccaletti et al., 2007] Boccaletti, S., Ivanchenko, M., Latora, V., Pluchino, A., and Rapisarda, A. (2007). Detecting complex network modularity by dynamical clustering. *Physical Review E*, 75(4):045102.

[Bouquet et al., 2003] Bouquet, P., Giunchiglia, F., Van Harmelen, F., Serafini, L., and Stuckenschmidt, H. (2003). C-owl: Contextualizing ontologies. In *International Semantic Web Conference*, pages 164–179. Springer.

[Bouquet et al., 2007] Bouquet, P., Stoermer, H., and Giacomuzzi, D. (2007). Okkam: Enabling a web of entities. *I3*, 5:7.

[Caracciolo et al., 2013] Caracciolo, C., Stellato, A., Morshed, A., Johannsen, G., Rajbhandari, S., Jaques, Y., and Keizer, J. (2013). The agrovoc linked dataset. *Semantic Web*, 4(3):341–348.

[Carroll et al., 2005] Carroll, J. J., Bizer, C., Hayes, P., and Stickler, P. (2005). Named graphs, provenance and trust. In *International conference WWW*, pages 613–622. ACM.

[Correndo et al., 2012] Correndo, G., Penta, A., Gibbins, N., and Shadbolt, N. (2012). Statistical analysis of the owl: sameas network for aligning concepts in the linking open data cloud. In *International Conference on Database and Expert Systems Applications*, pages 215–230. Springer.

[Cox and Little, 2017] Cox, S. and Little, C. (2017). Time ontology in owl. *W3C Recommendation*.

[CudreMauroux et al., 2009] CudreMauroux, P., Haghani, P., Jost, M., Aberer, K., and De Meer, H. (2009). idmesh: graph-based disambiguation of linked data. In *International conference WWW*, pages 591–600. ACM.

[Cuzzola et al., 2015] Cuzzola, J., Bagheri, E., and Jovanovic, J. (2015). Filtering inaccurate entity co-references on the linked open data. In *International DEXA Conference*, pages 128–143. Springer.

[d'Aquin and Noy, 2012] d'Aquin, M. and Noy, N. F. (2012). Where to publish and find ontologies? a survey of ontology libraries. *Web Semantics: Science, Services and Agents on the World Wide Web*, 11:96–111.

[de Melo, 2013] de Melo, G. (2013). Not quite the same: Identity constraints for the web of linked data. In *The Twenty-Seventh AAAI Conference on Artificial Intelligence*. AAAI Press.

[de Rooij et al., 2016] de Rooij, S., Beek, W., Bloem, P., van Harmelen, F., and Schlobach, S. (2016). Are names meaningful? quantifying social meaning on the semantic web. In *International Semantic Web Conference*, pages 184–199. Springer.

[Ding et al., 2010a] Ding, L., Shinavier, J., Finin, T., McGuinness, D. L., et al. (2010a). owl: sameas and linked data: An empirical study. In *Proceedings of the Second Web Science Conference*.

[Ding et al., 2010b] Ding, L., Shinavier, J., Shangguan, Z., and McGuinness, D. L. (2010b). Sameas networks and beyond: analyzing deployment status and implications of owl: sameas in linked data. In *International Semantic Web Conference*, pages 145–160. Springer.

[Donetti and Munoz, 2004] Donetti, L. and Munoz, M. A. (2004). Detecting network communities: a new systematic and efficient algorithm. *Journal of Statistical Mechanics: Theory and Experiment*, 2004(10):P10012.

[Euzenat et al., 2007] Euzenat, J., Shvaiko, P., et al. (2007). *Ontology matching*, volume 18. Springer.

[Färber et al., 2016] Färber, M., Bartscherer, F., Menne, C., and Rettinger, A. (2016). Linked data quality of dbpedia, freebase, opencyc, wikidata, and yago. *Semantic Web*, (Preprint):1–53.

[Fernández et al., 2017] Fernández, J. D., Beek, W., Martínez-Prieto, M. A., and Arias, M. (2017). Lod-a-lot. In *International Semantic Web Conference*, pages 75–83. Springer.

[Fernández et al., 2013] Fernández, J. D., Martínez-Prieto, M. A., Gutiérrez, C., Polleres, A., and Arias, M. (2013). Binary rdf representation for publication and exchange (hdt). *Web Semantics: Science, Services and Agents on the World Wide Web*, 19:22–41.

[Ferrara et al., 2013] Ferrara, A., Nikolov, A., and Scharffe, F. (2013). Data linking for the semantic web. *Semantic Web: Ontology and Knowledge Base Enabled Tools, Services, and Applications*, 169:326.

[Forrest, 2008] Forrest, P. (2008). The identity of indiscernibles. *The Stanford Encyclopedia of Philosophy*.

[Fortunato, 2010] Fortunato, S. (2010). Community detection in graphs. *Physics reports*, 486(3-5):75–174.

[Freeman and White, 1993] Freeman, L. C. and White, D. R. (1993). Using galois lattices to represent network data. *Sociological methodology*, pages 127–146.

[Geach, 1967] Geach, P. (1967). Identity. *Review of Metaphysics*, 21:3–12.

[Giménez-García et al., 2017] Giménez-García, J. M., Zimmermann, A., and Maret, P. (2017). Ndfluents: an ontology for annotated statements with inference preservation. In *International ESWC Conference*, pages 638–654. Springer.

[Girvan and Newman, 2002] Girvan, M. and Newman, M. E. (2002). Community structure in social and biological networks. *Proceedings of the national academy of sciences*, 99(12):7821–7826.

[Glaser et al., 2009] Glaser, H., Jaffri, A., and Millard, I. (2009). Managing co-reference on the semantic web. In *Proceedings of the WWW Workshop on Linked Data on the Web, LDOW*.

[Grant and Subrahmanian, 1995] Grant, J. and Subrahmanian, V. S. (1995). Reasoning in inconsistent knowledge bases. *IEEE Trans. Knowl. Data Eng.*, 7(1):177–189.

[Guéret et al., 2012] Guéret, C., Groth, P., Stadler, C., and Lehmann, J. (2012). Assessing linked data mappings using network measures. In *Extended Semantic Web Conference*, pages 87–102. Springer.

[Guichard et al., 2017] Guichard, E., Repoux, M., Qannari, E., Labouré, H., and Feron, G. (2017). Model cheese aroma perception is explained not only by in vivo aroma release but also by salivary composition and oral processing parameters. *Food & function*, 8(2):615–628.

[Halpin, 2010] Halpin, H. (2010). Sense and reference on the web (doctoral dissertation). University of Edinburgh.

[Halpin et al., 2010] Halpin, H., Hayes, P. J., McCusker, J. P., McGuinness, D. L., and Thompson, H. S. (2010). When owl:sameAs isn't the same: An analysis of identity in Linked Data. In *International Semantic Web Conference*, pages 305–320. Springer.

[Halpin et al., 2015] Halpin, H., Hayes, P. J., and Thompson, H. S. (2015). When owl: sameas isn't the same redux: towards a theory of identity, context, and inference on the semantic web. In *International and Interdisciplinary Conference on Modeling and Using Context*, pages 47–60. Springer.

[Halpin and Presutti, 2009] Halpin, H. and Presutti, V. (2009). An ontology of resources: Solving the identity crisis. In *European Semantic Web Conference*, pages 521–534. Springer.

[Hodgson et al., 2014] Hodgson, R., Keller, P. J., Hodges, J., and Spivak, J. (2014). Qudt-quantities, units, dimensions and data types ontologies. *USA Available http://qudt.org March*.

[Hogan et al., 2010] Hogan, A., Harth, A., Passant, A., Decker, S., and Polleres, A. (2010). Weaving the pedantic web. *LDOW*, 628.

[Hogan et al., 2011] Hogan, A., Harth, A., Umbrich, J., Kinsella, S., Polleres, A., and Decker, S. (2011). Searching and browsing linked data with swse: The semantic web search engine. *Web semantics: science, services and agents on the world wide web*, 9(4):365–401.

[Hogan et al., 2012] Hogan, A., Zimmermann, A., Umbrich, J., Polleres, A., and Decker, S. (2012). Scalable and distributed methods for entity matching, consolidation and disambiguation over linked data corpora. *Web Semantics: Science, Services and Agents on the World Wide Web*, 10:76–110.

[Huang et al., 2017] Huang, Z., Yang, J., van Harmelen, F., and Hu, Q. (2017). Constructing disease-centric knowledge graphs: a case study for depression (short version). In *Conference on Artificial Intelligence in Medicine in Europe*, pages 48–52. Springer.

[Ibanescu et al., 2016] Ibanescu, L., Dibie, J., Dervaux, S., Guichard, E., and Raad, J. (2016). Po2 - a process and observation ontology in food science. application to dairy gels. In *Research Conference on Metadata and Semantics Research*, pages 155–165. Springer.

[Ilievski et al., 2015] Ilievski, F., Beek, W., Van Erp, M., Rietveld, L., and Schlobach, S. (2015). Lotus: Linked open text unleashed. In *Proceedings of Consuming Linked Open Data (COLD)*.

[Jaffri et al., 2008] Jaffri, A., Glaser, H., and Millard, I. (2008). URI disambiguation in the context of linked data. In *WWW Workshop on Linked Data on the Web, LDOW*.

[Janowicz et al., 2018] Janowicz, K., Haller, A., Cox, S. J., Le Phuoc, D., and Lefrancois, M. (2018). Sosa: A lightweight ontology for sensors, observations, samples, and actuators. *Journal of Web Semantics*.

[Jonquet et al., 2018] Jonquet, C., Toulet, A., Arnaud, E., Aubin, S., Yeumo, E. D., Emonet, V., Graybeal, J., Laporte, M.-A., Musen, M. A., Pesce, V., et al. (2018). Agroportal: A vocabulary and ontology repository for agronomy. *Computers and Electronics in Agriculture*, 144:126–143.

[Joshi et al., 2012] Joshi, A. K., Jain, P., Hitzler, P., Yeh, P. Z., Verma, K., Sheth, A. P., and Damova, M. (2012). Alignment-based querying of linked open data. In *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, pages 807–824. Springer.

[Kripke, 1972] Kripke, S. A. (1972). Naming and necessity. In *Semantics of natural language*, pages 253–355. Springer.

[Krisnadhi et al., 2015] Krisnadhi, A. A., Hitzler, P., and Janowicz, K. (2015). On the capabilities and limitations of owl regarding typecasting and ontology design pattern views. In *International Experiences and Directions Workshop on OWL*, pages 105–116. Springer.

[Lancichinetti and Fortunato, 2009a] Lancichinetti, A. and Fortunato, S. (2009a). Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities. *Physical Review E*, 80(1):016118.

[Lancichinetti and Fortunato, 2009b] Lancichinetti, A. and Fortunato, S. (2009b). Community detection algorithms: a comparative analysis. *Physical review E*, 80(5):056117.

[Lancichinetti et al., 2008] Lancichinetti, A., Fortunato, S., and Radicchi, F. (2008). Benchmark graphs for testing community detection algorithms. *Physical review E*, 78(4):046110.

[Lewis, 1986] Lewis, D. (1986). On the plurality of worlds. *Oxford*, 14:43.

[Loyola, 2007] Loyola, W. (2007). Comparison of approaches toward formalising context: implementation characteristics and capacities. *Electronic Journal of Knowledge Management*, 5(2):203–214.

[MacQueen et al., 1967] MacQueen, J. et al. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA.

[Mealling and Daniel, 1999] Mealling, M. and Daniel, R. (1999). Uri resolution services necessary for urn resolution (rfc 2483).

[Miles and Bechhofer, 2009] Miles, A. and Bechhofer, S. (2009). Skos simple knowledge organization system reference. w3c recommendation 18 august 2009.

[Nentwig et al., 2017] Nentwig, M., Hartung, M., Ngonga Ngomo, A.-C., and Rahm, E. (2017). A survey of current link discovery frameworks. *Semantic Web*, 8(3):419–436.

[Newman and Girvan, 2004] Newman, M. E. and Girvan, M. (2004). Finding and evaluating community structure in networks. *Physical review E*, 69(2):026113.

[Newman and Leicht, 2007] Newman, M. E. and Leicht, E. A. (2007). Mixture models and exploratory analysis in networks. *Proceedings of the National Academy of Sciences*, 104(23):9564–9569.

[Nguyen, 2007] Nguyen, N. T. (2007). *Advanced Methods for Inconsistent Knowledge Management (Advanced Information and Knowledge Processing)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

[Nguyen et al., 2014] Nguyen, V., Bodenreider, O., and Sheth, A. (2014). Don't like RDF reification?: making statements about statements using singleton property. In *International conference WWW*, pages 759–770. ACM.

[Nurdiati and Hoede, 2008] Nurdiati, S. and Hoede, C. (2008). 25 years development of knowledge graph theory: the results and the challenge. *Memorandum*, 1876.

[Palla et al., 2005] Palla, G., Derényi, I., Farkas, I., and Vicsek, T. (2005). Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435(7043):814.

[Papaleo et al., 2014] Papaleo, L., Pernelle, N., Saïs, F., and Dumont, C. (2014). Logical detection of invalid sameas statements in rdf data. In *International Conference EKAW*, pages 373–384. Springer.

[Paulheim, 2014] Paulheim, H. (2014). Identifying wrong links between datasets by multi-dimensional outlier detection. In *WoDOOM*, pages 27–38.

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

[Pénicaud et al., 2014] Pénicaud, C., Landaud, S., Jamme, F., Talbot, P., Bouix, M., Ghorbal, S., and Fonseca, F. (2014). Physiological and biochemical responses of yarrowia lipolytica to dehydration induced by air-drying and freezing. *PloS one*, 9(10):e111138.

[Plantié and Crampes, 2013] Plantié, M. and Crampes, M. (2013). Survey on social community detection. In *Social media retrieval*, pages 65–85. Springer.

[Polleres et al., 2018] Polleres, A., Kamdar, M. R., Fernandez Garcia, J. D., Tudorache, T., and Musen, M. A. (2018). A more decentralized vision for linked data. In *Working Papers on Information Systems, Information Business and Operations*. Department fur Informationsverarbeitung und Prozessmanagement, WU Vienna University of Economics and Business, Vienna.

[Porter et al., 2009] Porter, M. A., Onnela, J.-P., and Mucha, P. J. (2009). Communities in networks. *Notices of the AMS*, 56(9):1082–1097.

[Raad et al., 2018a] Raad, J., Beek, W., Pernelle, N., Saïs, F., and van Harmelen, F. (2018a). Détection de liens d'identité erronés en utilisant la détection de communautés dans les graphes d'identité. 23(3-4):95–118.

[Raad et al., 2018b] Raad, J., Beek, W., van Harmelen, F., Pernelle, N., and Saïs, F. (2018b). Detecting erroneous identity links on the web using network metrics. In *International Semantic Web Conference*, pages 391–407. Springer.

[Raad et al., 2017a] Raad, J., Pernelle, N., and Saïs, F. (2017a). Détection de liens d'identité contextuels dans une base de connaissances. In *IC 2017-28es Journées francophones d'Ingénierie des Connaissances*, pages 56–67.

[Raad et al., 2017b] Raad, J., Pernelle, N., and Saïs, F. (2017b). Detection of contextual identity links in a knowledge base. In *Proceedings of the Knowledge Capture Conference*, page 8. ACM.

[Raad et al., 2018c] Raad, J., Pernelle, N., Saïs, F., Dibie, J., Ibanescu, L., and Dervaux, S. (2018c). Comment représenter et découvrir des liens d'identités contextuels dans une base de connaissances : applications à des données expérimentales en science du vivant. 32(3):345–372.

[Reichardt and Bornholdt, 2004] Reichardt, J. and Bornholdt, S. (2004). Detecting fuzzy community structures in complex networks with a potts model. *Physical Review Letters*, 93(21):218701.

[Ronhovde and Nussinov, 2009] Ronhovde, P. and Nussinov, Z. (2009). Multiresolution community detection for megascale networks by information-based replica correlations. *Physical Review E*, 80(1):016109.

[Rosvall and Bergstrom, 2008] Rosvall, M. and Bergstrom, C. T. (2008). Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences*, 105(4):1118–1123.

[Sarasua et al., 2017] Sarasua, C., Staab, S., and Thimm, M. (2017). Methods for intrinsic evaluation of links in the web of data. In *European Semantic Web Conference*, pages 68–84. Springer.

[Schlegel et al., 2014] Schlegel, K., Stegmaier, F., Bayerl, S., Granitzer, M., and Kosch, H. (2014). Balloon fusion: Sparql rewriting based on unified coreference information. In *Data Engineering Workshops (ICDEW), 2014 IEEE 30th International Conference on*, pages 254–259. IEEE.

[Schmachtenberg et al., 2014] Schmachtenberg, M., Bizer, C., and Paulheim, H. (2014). Adoption of the linked data best practices in different topical domains. In *International Semantic Web Conference*, pages 245–260. Springer.

[Singhal, 2012] Singhal, A. (2012). Introducing the knowledge graph: things, not strings. https://googleblog.blogspot.com/2012/05/introducing-knowledge-graph-things-not.html.

[Valdestilhas et al., 2017] Valdestilhas, A., Soru, T., and Ngomo, A.-C. N. (2017). Cedal: time-efficient detection of erroneous links in large-scale link repositories. In *International Conference on Web Intelligence*, pages 106–113. ACM.

[Verborgh et al., 2017] Verborgh, R., Kuhn, T., and Sambra, A., editors (2017). *Proceedings of the Workshop on Decentralizing the Semantic Web*, number 1934 in CEUR Workshop Proceedings, Aachen.

[Verborgh et al., 2016] Verborgh, R., Vander Sande, M., Hartig, O., Van Herwegen, J., De Vocht, L., De Meester, B., Haesendonck, G., and Colpaert, P. (2016). Triple pattern fragments: a low-cost knowledge graph interface for the web. *Web Semantics: Science, Services and Agents on the World Wide Web*, 37:184–206.

[Wang et al., 2014] Wang, X., Tiropanis, T., and Davis, H. C. (2014). Optimising linked data queries in the presence of co-reference. In *European Semantic Web Conference*, pages 442–456. Springer.

[Wilkinson et al., 2016] Wilkinson, M. D., Dumontier, M., Aalbersberg, I. J., Appleton, G., Axton, M., Baak, A., Blomberg, N., Boiten, J.-W., da Silva Santos, L. B., Bourne, P. E., et al. (2016). The fair guiding principles for scientific data management and stewardship. *Scientific data*, 3.

[Xie et al., 2013] Xie, J., Kelley, S., and Szymanski, B. K. (2013). Overlapping community detection in networks: The state-of-the-art and comparative study. *Acm computing surveys (csur)*, 45(4):43.

[Yang et al., 2016] Yang, Z., Algesheimer, R., and Tessone, C. J. (2016). A comparative analysis of community detection algorithms on artificial networks. *Scientific Reports*, 6:30750.

[Zhou and Lipowsky, 2004] Zhou, H. and Lipowsky, R. (2004). Network brownian motion: A new method to measure vertex-vertex proximity and to identify communities and subcommunities. In *International conference on computational science*, pages 1062–1069. Springer.

université PARIS-SACLAY
ÉCOLE DOCTORALE
Agriculture, alimentation, biologie, environnement, santé (ABIES)

Titre : Gestion d'identité dans des graphes de connaissances

Mots clés : Web sémantique; Web de données; Graphes de Connaissances; Identité

Résumé : En l'absence d'une autorité de nommage centrale sur le Web de données, il est fréquent que différents graphes de connaissances utilisent des noms (IRI) différents pour référer à la même entité. Chaque fois que plusieurs noms sont utilisés pour désigner la même entité, les faits *owl:sameAs* sont nécessaires pour déclarer des liens d'identité et améliorer l'exploitation des données disponibles. De telles déclarations d'identité ont une sémantique logique stricte, indiquant que chaque propriété affirmée à un nom sera également déduite à l'autre et vice versa. Bien que ces inférences puissent être extrêmement utiles pour améliorer les systèmes fondés sur les connaissances tels que les moteurs de recherche et les systèmes de recommandation, l'utilisation incorrecte de l'identité peut avoir des effets négatifs importants dans un espace de connaissances global comme le Web de données. En effet, plusieurs études ont montré que *owl:sameAs* est parfois incorrectement utilisé sur le Web des données. En s'appuyant sur une collection de 558 millions liens d'identité, cette thèse montre comment des mesures de réseau telles que la structure de communauté du réseau *owl:sameAs* peuvent être utilisées afin de détecter des liens d'identité éventuellement erronées. En outre, afin de limiter l'utilisation excessive et incorrecte du *owl:sameAs*, nous définissons une nouvelle relation pour représenter l'identité de deux instances d'une classe dans un contexte spécifique. Cette relation d'identité s'accompagne d'une approche permettant de détecter automatiquement ces liens, avec la possibilité d'utiliser certaines contraintes expertes pour filtrer des contextes non pertinents. La détection et l'exploitation de ces liens d'identité contextuels sont effectuées sur un graphe de connaissances pour les sciences de la vie, construits en collaboration avec des experts de l'INRA.

Title: Identity Management in Knowledge Graphs

Keywords: Semantic Web; Linked Data; Knowledge Graphs; Identity

Abstract: In the absence of a central naming authority in the Web of Data, it is common for different knowledge graphs to refer to the same thing by different names (IRIs). Whenever multiple names are used to denote the same thing, *owl:sameAs* statements are needed in order to link the data and foster reuse. Such identity statements have strict logical semantics, indicating that every property asserted to one name, will also be inferred to the other, and vice versa. While such inferences can be extremely useful in enabling and enhancing knowledge-based systems such as search engines and recommendation systems, incorrect use of identity can have wide-ranging effects in a global knowledge space like the Web of Data. With several studies showing that *owl:sameAs* is indeed misused for different reasons, a proper approach towards the handling of identity links is required in order to make the Web of Data succeed as an integrated knowledge space. By relying on a collection of 558 million identity statements, this thesis shows how network metrics such as the community structure of the *owl:sameAs* graph can be used in order to detect possibly erroneous identity assertions. In addition, as a way to limit the excessive and incorrect use of *owl:sameAs*, we define a new relation for asserting the identity of two class instances in a specific context. This identity relation is accompanied by an approach for automatically detecting these links, with the ability of using certain expert constraints for filtering irrelevant contexts. As a first experiment, the detection and exploitation of the detected contextual identity links are conducted on a knowledge graph for life sciences, constructed in the context of this thesis in a collaboration with experts from the French National Institute of Agricultural Research (INRA).

Université Paris-Saclay
Espace Technologique / Immeuble Discovery
Route de l'Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France